

INTAREA
Internet-Draft
Intended status: Informational
Expires: 28 April 2022

T. Eckert
Futurewei Technologies USA
N. Shenoy
Rochester Institute of Technology
25 October 2021

Functional Addressing (FA) for internets with Independent Network
Address Spaces (IINAS)
draft-eckert-intarea-functional-addr-internets-01

Abstract

Recent work has raised interest in exploring network layer addressing that is more flexible than fixed-length addressing as used in IPv4 (32 bit) and IPv6 (128 bit).

The reasons for the interest include both support for multiple and potentially novel address semantics, but also optimizations of addressing for existing semantics such as unicast tailored not for the global Internet but to better support private networks / limited domains.

This memo explores in the view of the author yet little explored reasons for more flexible addresses namely the problems and opportunities for Internetworking with Independent Network Address Spaces (IINAS).

To better enable such internetworks, this memo proposes a framework for a Functional Addressing model. This model also intends to support several other addressing goals including programmability and multiple semantics.

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 <https://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 28 April 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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
1.1. Overview	3
1.2. Disclaimer	3
2. Challenges	4
2.1. High level observations	4
2.2. Internetworking limited domain networks with IP addressing	5
2.3. Shorter addresses	9
2.4. Additional semantics	9
2.5. Programmability	10
3. FA-IINAS: Functional Addressing (FA) for Internetworking with Independent Network Address Spaces (IINAS)	10
3.1. Addressing for unicast	11
3.2. Forwarding	12
3.2.1. Dispose Function	12
3.2.2. Steering Function	12
3.2.3. Multiple semantics	12
3.2.4. Internetworking Function	14
3.3. Control Plane	16
3.3.1. Unicast routing	16
3.3.2. Naming	17
3.3.3. Routing	18
3.3.4. Routing policies	19
3.4. Hardware considerations	20
3.4.1. Forwarding plane simplicity	20
3.4.2. Optimizing for smaller networks	21
3.4.3. Maximum address sizes	21
3.5. Example packet header encoding	21
4. Inspirations	22
4.1. E.164	23

4.2. MPLS	25
4.3. Segment Routing SR-MPLS / SRv6	25
4.4. Research	26
5. Summary and conclusions	26
6. Changelog	27
7. Informative References	27
Authors' Addresses	30

1. Introduction

1.1. Overview

Recent work has examined the value of more flexible than fixed-length addressing used in IPv4 (32 bit) and IPv6 (128 bit), see for example [I-D.jia-intarea-scenarios-problems-addressing], and [I-D.jia-flex-ip-address-structure].

The reasons for this interest include both support for multiple and potentially novel address semantics, see for example [I-D.king-irtf-semantic-routing-survey] and [I-D.king-irtf-challenges-in-routing], but also optimizations of addressing for existing semantics, such as unicast, that are tailored not for the global Internet but to better support private networks and limited domains ([RFC8799]).

This memo describes one, in the view of the author yet little explored reason, for more flexible addresses namely the problems and opportunities for Internetworking with Independent Network Address Spaces (IINAS).

To better enable such internetworks, this memo proposes a framework for a Functional Addressing model. This model also intends to support several other addressing model goals including programmability and multiple semantics.

This memo calls the addressing model functional, because addresses are constructed as a structure of
`func1{parameter(s),func2{parameter(s),...i.funcN{parameter(s)}}}`.

1.2. Disclaimer

Any proposals made by this document are explicitly for the purpose of presenting example options of realizing concepts introduced in the memo. There is no intent for any proposals in this document to directly become anything more than just experimental implementations for proof of concept purposes. Equally so or even more so, readers are welcome to pick up any subset of ideas from this memo that they are interested in and reuse it in other designs.

2. Challenges

This section discusses challenges that gave rise to the proposal in this document. It explores in more detail the core challenge not well explored elsewhere and already detailed elsewhere.

2.1. High level observations

There are three core challenges we can observe that limit the ability to build more varied internetworking solutions for non-solely Internet use-cases with especially IPv6:

- * Fixed size address space: IPv4/IPv6 address space is fixed length, not allowing to adopt address length to shorter or longer demands. While it is possible to add more addressing via extension headers, there is no option to not send, or shorten the IPv4/IPv6 base header addresses, when they are not required. While the reasons for fixed size addressing in IPv4/IPv6 can be understood for the feasible high-speed, low-cost forwarders of the 1900th, when IPv6 was conceived, these reasons are today (in the opinion of the author) as obsolete as ATM cells where by the end of the 1990th when both hardware forwarding and mathematical models allowed to provide all ATM type QoS with variable sized packets.
- * The Internet as the primary, if not only use-case driving the design: The address space semantics provided especially by IPv6 is very much focused on the one use-case that drove the development of IPv6: The Internet. While it was and will continue to be the core and sufficient reason for maintaining IPv6, it is not sufficient in the opinion of the author for the much broader use of IPv6. As of today, a likely overwhelming number of hosts using TCP/IP(v6) protocol stacks are not "on the Internet" and the majority likely is not even "connected to the Internet", but instead, they are part of limited domains. This even includes many routers in large service providers that are used to service Internet traffic. Routers in these networks are only in networks that may be called an "underlay" limited domain networks using MPLS, SR-MPLS or SRv6 and Internet traffic is tunneled across them. When the network design is secure, those routers are neither "on" the internet nor "connect to" the Internet.
- * Transparent end-to-end addressing at the core of the IP/IPv6 protocol design, but an ever more diverse reality breaking that design for good reasons: The current core principle of IPv4 and IPv6 is that forwarders have to be passing network layer (IPv4/IPv6) addresses transparently and are not allowed to touch/modifying them. This is the core behavior to support primarily the Internet use case. Yet, the IPv4 Internet today would not

work without NAT, and arguably, the same may also happen to the IPv6 Internet, especially when networks attaching to inexpensive Internet offerings want to avoid complex src/dst forwarding for IPv6 multihoming, and/or avoid renumbering upon change of provider addresses. Even more so, interconnecting IPv4 and IPv6 networks has resulted in no fewer than 24 IPv4/IPv6 NAT solutions (see https://en.wikipedia.org/wiki/IPv6_transition_mechanism), giving rise to the question if and how on-path processing of addressing can be proactively become part of future addressing designs to support more flexible internetworking - translating the best of past NAT experience into better future designs. This is a core option of what FA-IINAS can do.

2.2. Internetworking limited domain networks with IP addressing

One of the core challenges of the existing IP(v4) and IPv6 addressing model are the addressing they provide for private networks with or without connectivity to the Internet, which are also called limited domain networks [RFC8799].

One reference example is that of networking inside a particular product/solution/installation, and then compositing this product with other products, probably even multiple times, hierarchical, as show in picture Figure 1. These type of designs are traditional in industrial networks. Similar issues and solutions can be found in networks with multiple layers of NAT such as Home Networks that are dorm rooms connected via NAT to a dorm network, connected via another NAT to a campus network, connected via yet another NAT to maybe finally, the Internet. Similarly designs can happen with more complex topologies in federated private networks.

In pre-IP industrial networks, individual products were hiding their interior elements by some (combination) of elements that controlled the interior behavior completely and provided only an abstracted view of the machinery to the outside.

With the introduction of IP networking into these type of solutions, the ability for gateways to become IP routers and providing connectivity into the machinery throughout the larger internetwork opened up many important improvements, but of course also challenges, especially for security.

Benefits of network layer internetwork connectivity includes options such as control loops that can more easily be built across multiple components/levels of the hierarchy and controllers that can be pulled out of machinery and positioned elsewhere in the network, enabling virtualization and resource multiplexing. Multiple independently running control systems can be implemented in parallel, including

solutions like device vendor preventive maintenance telemetry, operator managed firmware update or third-party orchestrated security audits or intrusion detection/prevention, just to name a few.

With IP connectivity, all this can be built without the need of understanding how to get through various layers of fixed-functionality higher-than-network layer gateways that can not be extended by third parties. Instead, new designs are based on end-to-end IP connectivity - plus appropriate set of security measures at gateway routers, of course an appropriate set of security/filtering measures, for example MUD, [RFC8520].

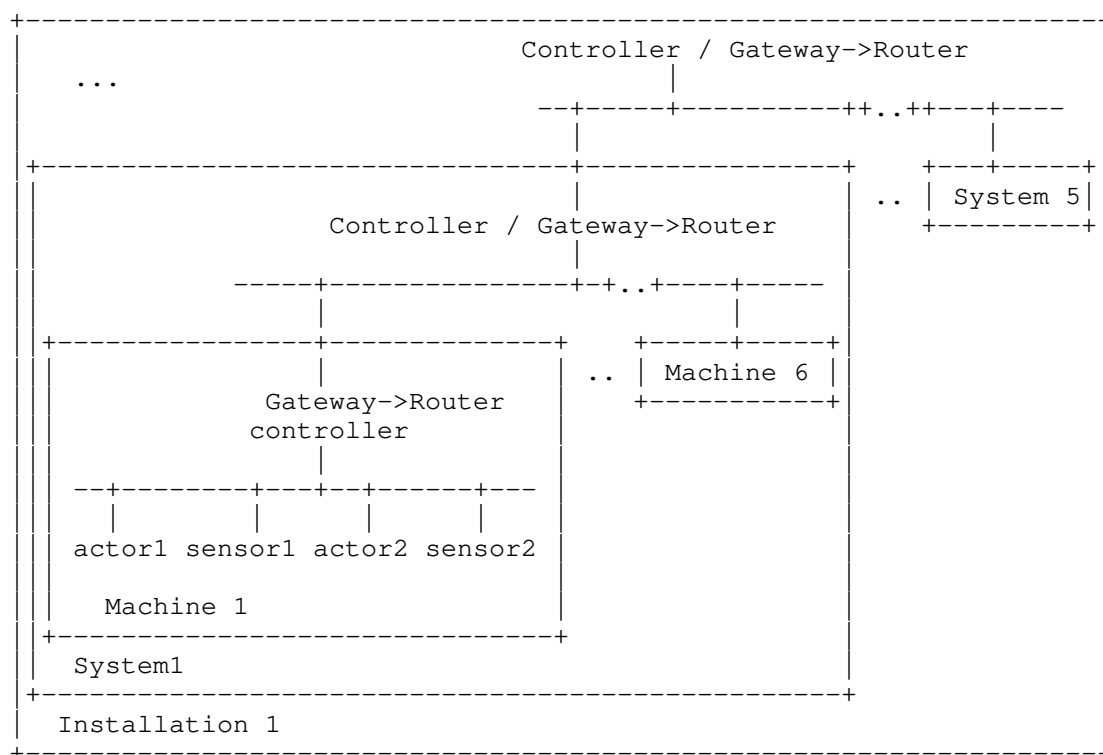


Figure 1: Example hierarchical composed internetwork

In the opinion of the author, the most easily adopted addressing architecture in these type of solutions today is also the one widely used: IPv4 with [RFC1918] addresses. These addresses are actually owned permanently for each deployment case - as long as the scope of addressing is well defined.

In result, a common scheme of addressing in machinery such as the one shown in Figure 1 is to reuse the same 10.0.0.0/8 or 192.168.0.0/16 addresses for every instance of a product/machinery manufactured. In the example, actor1 could use 10.0.0.1, sensor1 10.0.0.2 and so on. But equally, if Machine 3 was the same or similar, its internal components would share the same machinery. And when hundreds of these products are produced, they would all have the same addresses.

To allow deployment and composing those type of machineries, the router/switch connecting to the outside/next-level in a hierarchy will need simple NATing function for example statically mapping the 10.0.0.x on the inside to 10.0.1.x on the outside for Machine 1, where the same router/switch for Machine 3 would be configured to NAT from 10.0.0.x to 10.0.3.x. And likewise at the next layer of hierarchy, 10.0.y.x could be mapped to 10.z.y.x with a different y for every instance.

In support of solutions like this, many if not most industrial ethernet switches deployable as machinery gateways do therefore support this type of static NAT mappings. Likewise, common practices in industries rely on this addressing with composition via NAT approaches, including machineries as large as production lines or in transportation networks train cars and all their included machineries/equipment.

The desire to avoid NAT in IPv6 and availability of sufficient addressing space lead to replacing the concept of [RFC1918] in IPv4 with the concept of Unique Local Addresses (ULA) in IPv6, standardized in [RFC4193]. Instead of the few scoped prefixes of [RFC1918], ULA provide for 2^{40} different prefixes, and the design guidelines are theoretically simple: pick a random prefix and then you can interconnect your networks later on with a very low probability of address prefix collision/reuse.

Unfortunately, low probabilities of address collision is not a good design principle for most of these type of environments because there is really no good operational solution what do if such collision occurs, and rare errors are also very hard to build resilient solutions for. Also the probabilities begin to become much higher when not looking at a connection of just two or few of such ULA networks, but when there can be thousands of such networks, such as in the transportation networks use case.

In result, ULA is not very persuasive for many such deployments, especially when the alternative with IPv4 is address prefix mapping as required for NAT, when NAT an an almost free provisioning side effect of setting up the required connectivity via permit lists via network/transport filters. The need to automate such in-network filtering to secure such deployments can also be seen in the advent of MUD, [RFC8520].

If one considers that most of these subnet networks will have fewer than 253 hosts connected to it, then the IPv6 ULA solution does also not provide for any more bits for subnets than the 16 bits of z.y in the above example using IPv4 10.z.y.x with x being the host part: The lower 64 bits of the IPv6 address is hard to use for anything than the host parts with non-router hosts. The whole ULA prefix is 48 bits, leaving just 16 bit (128 - 64 -48). Add to that the non insignificant IPv6 packet header overhead plus fewer availability of NAT in IPv6 products because it is assumed to be less required, plus the insufficiency of "low likelihood of collisions" when attempting to utilize only ULA.

Vendors of equipment that have assigned Provider Independent IPv6 address space could of course allocate addressing from that space for equipment they manufacture or integrate, whether it is globally unique or "generic", e.g.: reused across every instance of a product and hence requiring NAT. Unfortunately, and unlike ethernet, where one actually does own addresses after buying an OUI, assigned IPv6 addressing is not permanent, and even though revocation of address allocation is not standard practice, standardized solutions for global IPv6 address space (like IPv4 global address space) really need to allow the ability for those addresses to be returnable instead of being handed off in products to customers.

Even though in hindsight, the hierarchical address allocation from the available 16 bits in 10.x.y.z for two layers of interconnections in the above example looks obvious and simple, in many cases the creation of multiple hierarchies is only an afterthought and the fixed address length and prior suboptimal assignment of addressing in a deployment will cause the need for a lot of re-addressing. This is a recurring problem in larger enterprise/commercial networks under unplanned growth or mergers & acquisitions, especially of course in IPv4. Likewise, once the 16 available bits in the above described NAT approach are used up, whether it is IPv4 or IPv6 with ULA, no further extensions of the design are possible.

2.3. Shorter addresses

As has been noted in prior memos, shorter addresses than IPv6 128 bit are highly desirable in private networks / limited domains whenever it is clear that the total required addressing space is much smaller and connectivity to e.g.: the Internet is not required. Evidence of such requirements can be found for example in header compression for IoT networks such as [RFC6282]. Such compression introduces yet another layer of complexity - the whole ecosystem of devices and diagnostic options has to support it to be equally acceptable as uncompressed packets.

2.4. Additional semantics

New semantics can only be introduced into existing IPv4/IPv6 when their required address size fits nicely into the 32 or 128 bit address space.

This section does not aim to be complete, see [I-D.king-irtf-semantic-routing-survey] for a broader survey. Instead it will provide additional levels of details for the benefits of fittingly sized addresses for few examples, that the author is familiar with.

When ignoring Anycast, IP Multicast is likely the most widely adopted additional semantic added to IPv4. With IPv6, IP Multicast became even more flexible and easy to deploy, because the additional bits of IPv6 addresses allowed to encode additional IP multicast parameters through additional fields in IPv6 addresses: Scope address field [RFC4291], SSM addresses [RFC4607], Unicast prefix multicast addresses [RFC3306] and embedded-RP [RFC3956]. Nevertheless, especially embedded-RP could have benefitted from even longer addresses because with the 128 bits available the solution had to take a hit in the complexity of deployment. It requires to engineer that RP address such that its non-0 host port is very short (4 bits).

In contrast, Bit Indexed Explicit Replication (BIER) which started in the IETF in 2014 and resulted in the architecture [RFC8279], did not choose the option to integrate into IP/IPv6 because it desired addresses sizes of at least per-network configurable from 64 to 4096 bit plus additional qualifiers of at least 16 bits (so-called SD, SI address qualifiers). This made it necessary for BIER to (re-)invent its own network layer packet header, [RFC8296] which duplicates pretty much all packet header fields of MPLS plus IP packets plus additional BIER header fields, so that it can be used in both MPLS and non-MPLS networks.

Similar arguments about the limited size of IPv6 address could likely be made for ICN/CCN networks because the semantic of their addresses is that of data items such as time slices of specific spatial and temporal resolutions of some media such as an audio/video recording - and those name spaces would ideally have addresses as long as URLs.

2.5. Programmability

Segment Routing via IPv6 (SRv6) introduced with [RFC8986] and [RFC8754] (SRH) and architecture in which source routing with an IPv6 extension header is combined with encoding of additional processing semantics into the destination and source routing hops IPv6 addresses. SRv6 calls this programmability.

SRv6 is a very flexible and theoretically extensible concept but challenged by the fixed address length design of IPv6. For most steering hop addresses, the bits reserved for this additional packet processing are not required, but when they are required there may even be too few bits available. Variable length addresses allowing for variable long programming field in the address would in the opinion of the author be highly beneficial.

One evidence for the programmability bits seen as wasteful in many cases is a variety of currently proposed drafts to provide more compressed source routing options for SRv6 (as of mid 2021).

3. FA-IINAS: Functional Addressing (FA) for Internetworking with Independent Network Address Spaces (IINAS)

This section outlines an addressing design that attempts to solve the above described challenges and calls it tentatively FA-IINAS. Functional Addressing refers to the design aspect that addresses in this design can be interpreted as functions with parameters.

Notwithstanding other granularities or options, this document assumes that addresses are textually represented in hexadecimal and that the minimum structure element of an address is 4 bit so that the different structural elements of an address can simply be shown as concatenation of hex digits. The "." character is inserted optionally to show where in an address one semantic part ends and another starts.

Like in IPv6 IoT networks, such as those using RPL ([RFC6550]) as their routing protocol, this memo starts by assuming all nodes are routers and that addresses are predominantly node addresses as opposed to IP/IPv6, which defines unicast addresses to be interface addresses. This is but an academic differentiation, because node addresses can also be represented as interface addresses of so-called "loopback" interfaces.

A network in this design is an independent address space, not shared with other networks. A network has theoretically unlimited long addresses whose prefixes are mapped onto the nodes of the network, which are expected to form a graph of transitively connected nodes. Practical limits to address length are subject to acceptable packetization.

3.1. Addressing for unicast

Each node is assigned one or more node prefixes from the networks address space and none of these node prefixes can be overlapping. In other words, no assigned nodeprefix can be a prefix of another assigned nodeprefix. This rule ensures that every node "owns" any address equal or longer to its assigned nodeprefix. Allocation of node prefixes is currently out of scope for this memo but could rely on any well-known methods including manual operator assigned, SDN controll managed, or as initially described in this document assigned by manufacturer/vendor.

Routing in a network is assumed to enable forwarding across the graph of the network to the node owning the nodeprefix of the address.

Given variable long addresses, the first observation of this addressing scheme is that it allows to combine short addresses with extensibility.

In a simple example the first 200 nodes are assigned addresses 01 ... c8, at which point in time the network operator gets worried about growth exceeding the 256 mark and starts to assign longer addresses: c90 ... f000, at which point in time ever increasing success might cause assignment of even longer prefixes.

Addresses longer than the assigned "nodeprefix" are used to instantiate a specific function on the node itself. A generic representation of an address could be
nodeprefix.function.{parameter}.

3.2. Forwarding

3.2.1. Dispose Function

When using a single digit function field, function = 0 could for example be "dispose" to decapsulate the packets payload and deliver it to the host stack. Parameter could for example be the next-protocol value, eliminating the need to have a separate packet header field for this parameter.

While not being the same crucial issue as for the node prefixes themselves, putting the next-protocol into the address makes it extensible too, so one would not run out of a 256 space as IPv4/IPv6 might do at some point.

3.2.2. Steering Function

Command = 1 could be a "steer" command and the parameter is another address. To act on the command, the node would strip the nodeprefix and command part of the address and forward it based on the address parameter. For example node 73 (e.g.: node with nodeprefix 73) receives a packet with destination address 73.1.55.1.33.0. It forwards the same packet with the stripped destination address 55.1.33.0 to node 55, which likewise forwards the packet with stripped destination address 33.0 to node 33, which ultimately receives it.

3.2.3. Multiple semantics

To introduce additional semantics into a network, such as for example multicasting, we need to generalize how to interpret the first part of the address, which so far was only interpreted to be a nodeprefix for unicast forwarding.

```
address = prefix{.nodefunction{.nodefunction-parameters}}
prefix = semantic{.semantic-parameters}
```

```
semantic / = unicast-forward
```

```
unicast-forward = <set of prefixes>
unicast-forward-parameters = node-prefix
```

```
semantic /= multicast-forward
multicast-forward = <set of prefixes>
multicast-forward-parameters = multicast-group
```

Figure 2

In other words, the prefix at the start of the address is composed of a semantic and its parameter, and the case discussed so far is simply the unicast-forward semantic followed by a node-prefix parameter.

Again, semantic can be an arbitrarily long or short prefix, but no semantic can be a prefix of another semantic.

In a practical example, this scheme is easily applied to existing IPv4 / IPv6 address spaces. For IPv4:

```
unicast-forward = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D
multicast-forward = E
```

Figure 3

In other words, because IP multicast uses addresses 224.0.0.0/4, its non-overlapping semantic prefix is E, and IPv4 unicast addresses use the non-overlapping prefixes 0...D. Assume further that a node in the network had assigned prefix 10.0.0.0/24, then this would translate in our scheme into:

0.A0000.XX

Figure 4

When a node processes this address, the 4-bit prefix 0 indicates that the following prefix has to be looked up in unicast forwarding. This prefix is A0000. Once the packet is delivered to the node, the remaining 8 bit XX can accordingly be interpreted by the node as a nodefunction with parameters.

Likewise, an address 239.1.2.3 would translate into E.F010203, so the first 4-bit E value would indicate that multicast forwarding needs to be applied to the rest of the address, and with IP Multicast forwarding not having further structure (ignoring willfully for simplicity of the example that it does, for example with SSM), all the remainder of the IPv4 address is the multicast-group

In summary, the logic does really only generalize what routers today already do when they do prefix lookups, except for the following core differences:

- * In IPv4/IPv6, the address semantic is hard-coded by IETF standards. In FA-IINAS they are definable by every network.
- * In IPv4/IPv6, there is no notion of nodefunction{.nodefunction-parameters}, only SRv6 has this concept.

In actual IPv4/IPv6 hardware forwarding lookups, one would not do one lookup for the semantic, followed by another lookup for the semantic-parameters for the case of unicast-forward, instead this would be flattened. The same type of flattening would of course be useable in FA-IINAS. Whether or how flattening or other optimizations are feasible for other semantics such as multicast is of course highly semantic and node implementation specific.

3.2.4. Internetworking Function

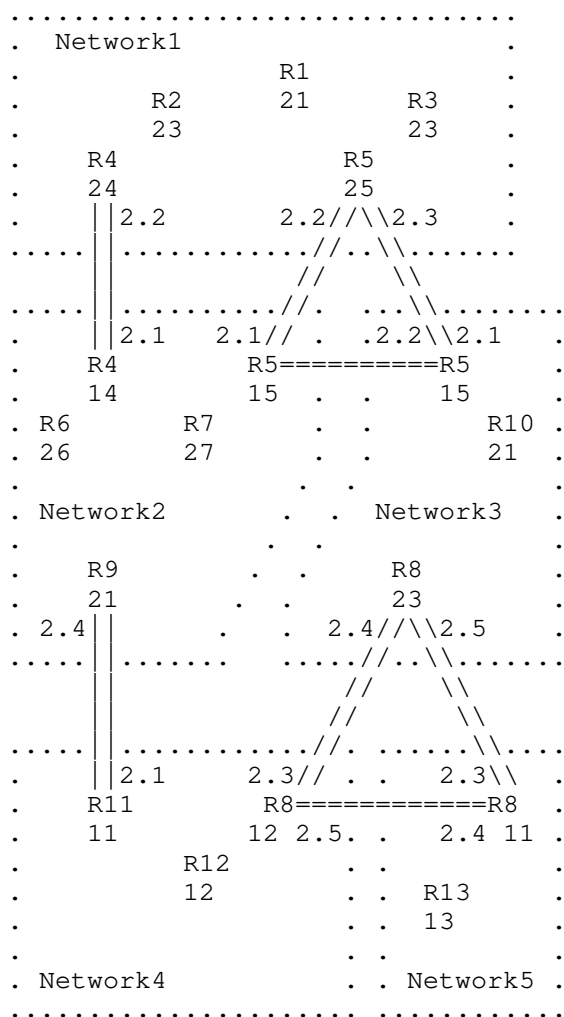


Figure 5: Internetworking example

Figure 5 shows an example internetworking topology of 5 networks, each with its own independent address space. Globally unique Rxx numbers are used to refer to routers.

An edge node is a router that has prefixes from two or more networks into which it connects. In the example, R4 connects into Network1 with prefix 24 and into Network2 with Prefix 14. Likewise, R8 connects into Network3 with prefix 23, into Network4 with prefix 12 and into Network5 with prefix 11. An edge node can be a router simply with different interfaces into different networks, or it can be decomposed into multiple devices, each in a separate network. In this section we describe behavior as if it was a single device.

For an edge node to pass a network into a separate network, the internetworking function on the node has to be called. In the example, this function is codepoint 2 on all edge nodes, and the first parameter is an identifier of local relevance for the network into which to pass the packet. In actual deployment, this function number can of course be locally significant to the Network and/or even each edge router, assuming appropriate control plane to assign the number to this function.

Assume R12 (12) in Network4 wants to send a packet to R1 (21) in Network1. To send it R12->R8->R5->R1, R12 would have to use a destination address of 12.2.3.15.2.1.21.0, or numerically without separators 0x12231521210.

12 will route the packet in Network4 towards R8 because of the destination address 12/8 prefix. .2 indicates to R8 that it should invoke the interworking function and pass the packet into Network 3. As part of the interworking function, R8 then strips all the address prefix it has processed so far from the destination address, leaving 15.2.1.21.0. R8 then forwards the packet with this destination address into Network 3, where it will be received by R5, which again invokes the interworking function due to .2, forwarding the packet into Network1, stripping 15.2.1.0 from the destination address and forwarding the packet with destination address 21.0 into Network1, where it will finally be received by R1 which passes the packet to its host stack because of dispose function 0.

To (optionally) allow for a return path, each edge node could equally but inversely process the source address: When R12 sends the packet, it would indicate a source address of 12.0. When R8 passes the packet via its interworking function into Network3, it would prepend its return path interworking function address, making the source address 23.2.4.12.0, where 23 is R8 address prefix in Network3 and 2.4 interworking function to return the packet into Network4. Likewise, when R5 processes the packet by its interworking function,

it would prepend its return path address element to the source address, before sending the packet into Network1, making the source address 25.2.3.23.2.4.12.0. This is then the address to which R1 could send return packets, and likewise, on its way towards R1, the address, for example when travelling via Network3 always has a returnable source address.

With this behavior of the interworking function, it is obvious, that address management of networks would want to keep a sufficiently large number of very short prefixes, such as those in this example or even shorter to address the interworking function in a sufficiently larger number of edge routers so that a complete internetwork path address will not become too long to exceed the maximum address lengths.

3.3. Control Plane

This section reviews a range of control plane considerations necessary to build a working solution out of the functional addressing. In short, what is required for functions to be flexibly configurable and extensible in the network, it requires a control plane that in its principles is very much based on what was learned in MPLS.

3.3.1. Unicast routing

FA-IINAS expects a control plane that supports routing for unicast-forward parameters (address prefixes) in the same way as it is done today for IPv4/IPv6. Except that it would be for address prefixes (multicast-forward-parameter) of different length and not limited to just 32/128 bits as in IPv4/IPv6.

In addition, FA-IINAS needs control-plane functions that allow defining the semantics and their prefixes, like the above example of 0...D for IPv4 style unicast-forwarding semantic and D for IPv4 style multicast-forwarding semantic.

One of the core challenges for this control plane function is that inconsistency between nodes can have significant different negative impacts than the today accepted "eventual consistency" in IPv4/IPv6 unicast routing that is achieved by the most widely deployed unicast forwarding control planes: distributed routing protocols (IGP/BGP).

The degree of concerns will highly depend on the actual new issues that could happen in the face of inconsistencies, and this can only be vetted with a given set of semantics.

In a most simple example, semantics may simple be configurable via a management plane, and such an approach can be pre-staged, pre-configured, validated network devices, such as in industrial or embedded environments.

In the case of a most flexible, agile type of network, control plane mechanisms would have to be extended to support strong consistency models, for example through node-to-node security associations coupled with a strong consistency network-wide-core-config mechanism. Such mechanisms could in the opinion of the author easily be built on the framework provided by [RFC8994] which provides these hop-by-hop security associations and inband control plane infrastructure, coupled with [RFC8990] as the protocol to negotiate the configuration with strong consistency.

3.3.2. Naming

3.3.2.1. Intra network naming

In FA-IINAS, nodes are acting as routers, and the addresses described are assigned to them persistently. This eliminates in many cases, especially when the network is primarily for m2m communications the need for DNS names, because effectively the address of a node is its persistent name.

In networks small enough, e.g.: maybe $\leq 20,000$ nodes, the very same argument can also apply to nodes that are hosts, e.g. without the need to support full routing/FA-IINAS operations, but still having a persistent address assigned that is routed in the networks routing protocol.

If indeed there is a need to use DNS or other naming schemes, then this is no different than applying naming with DNS to today's [RFC1918] addresses.

3.3.2.2. Simple inter network naming

The need to support (DNS) names is equally lower in interconnected FA-IINAS networks assuming the intra network naming arguments outlined before apply to the interconnected networks.

Because an address in a different FA-IINAS network is dependent on the path from/to its corresponding peer, it is of course not sufficient to simply have a global internetwork name to address mapping.

One of the likely oldest solutions is to align name resolution with packet forwarding so that the very same edge nodes between two networks that do translate addresses can accordingly also translate their name resolution. This was productized and fairly widely deployed as early as the late 1990th for IPv4 with rfc1918 addresses, see for example [CiscoNAT].

This type of solutions relies on well-known routing policies such as simple hierarchical routing though and are not generic for arbitrary topologies.

3.3.3. Routing

3.3.3.1. With internetwork topology knowledge

When FA-IINAS networks are connected in an arbitrary topology instead of a simple hierarchy, the fundamental problem is that of constructing the address of a target peer as a path through a set of appropriate network edge nodes in the address, followed by the nodes address within its network.

In many interconnected FA-IINAS networks, one can assume to have systems that can do this, such as in an industrial setting where a global view of the topology of networks exists and a PCE/SDN-controller will choose the path and can accordingly calculate also the addresses from the path.

3.3.3.2. With internetwork naming knowledge

A decentralized solution can be built by relying on a combination of naming and internetwork routing.

Every network (name space) is assigned a globally unique identifier. This identifier is only used in the control-plane, so it should be reasonably easy to have a set of construction mechanisms allowing everyone to easily create its own namespace, such as for example from some owned location (street address) and/or other owned names/identifier.

When a global naming system like DNS then exists, an FA-IINAS address is the combination of FA-IINAS network identifier and address within that network.

Across the interconnected FA-IINAS networks, the edge-routers would operate extended versions of a protocol like BGP through which any party can calculate desired paths. The extensions would include the FA-IINAS network identifiers and address prefix mapping rules of the edge-nodes, thereby allowing to also calculate addresses from FA-IINAS network identifiers and address.

When large number of small networks (such as users homes) connect to larger networks (such as an ISP), those ISP would be concerned of having to propagate millions of small FA-IINAS network mappings into BGP. This is not done today with IPv4/IPv6, and it would not scale any better with FA-IINAS. Instead, the fact that the home network would be reachable with one or more ISP could be done by also creating naming system mappings from the home networks identifier to the identifier and address prefix mappings of the ISP to which the home network is connected.

When a peer looks up a name and retrieves an FA-IINAS address but cannot find the FA-IINAS network identifier in its internetwork routing information, it can instead resolve it to the "next higher up" ISP FA-IINAS network-identifier/prefix - and recurse this until it has routing information.

Likewise, when a peer does not have any routing information (because it does not participate in internet routing information), it has to forward the appropriate resolution request hierarchically upward.

In summary, it would be architecturally "easy" to extend DNS and BGP with the necessary extensions to resolve names to FA-IINAS addresses and construct relative FA-IINAS addresses from this information.

3.3.4. Routing policies

Note that this "easy" part does not include the possible desire to be more or less flexible in path selection. Whereas today, packets, once they enter "the Internet" are not under steering control of the sender but under "hop by hop hot-potato steering" control of the ISP, with FA-IINAS this may be different - or the same. If a sender then constructed an FA-IINAS address implying an internetwork path that was not desirable for this traffic by the indicated transit networks, this would cause an error. Therefore, the above outlined procedures hinted at relying on the internetwork routing information whenever available and only resort to using naming system to fill in the additional (edge) information.

Today it is becoming more common to use alternative than "native Internet" paths by steering traffic across virtual/container routers in cloud DC, many of which have ample and underutilized international

connectivity. However, additional charges for compute and forwarding will apply. These type of high-overhead solutions could be replaced by FA-IINAS to steer traffic across such additional networks and without the need to instantiate VM/containers. It would require appropriate and lightweight identity and accounting forwarding plane packet header information so that those additional charges could be applied.

3.4. Hardware considerations

3.4.1. Forwarding plane simplicity

Forwarding of FA-IINAS packets based on destination address is the same type of prefix lookup on the destination address as it is today in IPv4/IPv6, except that the maximum lookup prefix can be shorter or longer, this is detailed in the next section.

The steering function should have a lookup complexity whose complexity is in the order of SR-MPLS or even simpler. It can constitute of a prefix lookup in the same forwarding table as non-steered forwarding, but the adjacency would then have to strip the looked up prefix from the destination address (comparable to MPLS label pop) and forward the packet again based on the remainder of the destination address - unless additional on-node service functions have to be invoked.

The interworking function is very much like the steering function, but it also prepends a return prefix to the source address field, making it the most expensive forwarding plane operation.

In general, the author assumes that packet processing that strips a prefix from the destination address and optionally adds a prefix to the source address is well feasible in next generation, highest-speed, lowest-cost forwarding engines.

Optimizations beyond this are possible but would break the independent address allocation across networks. For example, if it is possible for an edge node to have the same prefix length across the networks it connects to, and source address follows destination address in the packet encoding, then stripping the destination address could be achieved by shifting the destination address in a contiguous packet buffer, making head for the source address prefix to be prepended to the following source address field.

3.4.2. Optimizing for smaller networks

One of the benefits of FI-IINAS is that it allows to adopt the address space size based on the requirements of networks and therefore also allows to optimize hardware known to be built/sold only into limited size networks, such as many industrial and almost all embedded networks.

For example, low-cost, high-speed hardware forwarding might be possible to design less expensive with just 16 bit lookups instead of for up to 128 bit lookups, as may be required for IPv6. Equipment could be sold with that profile parameters "for networks with up to 2^{16} nodes".

Because of the way FA-IINAS is designed, a limit to 2^{16} nodes does not mean that FA-IINAS addresses are only 16 bits. Instead they can still be "arbitrary" long (where arbitrary is subject to a discussion point further below in this section). Just the length of the unicast-forward part of the address is limited to 16 bits.

3.4.3. Maximum address sizes

The permissible maximum size of source and destination address are primarily subject to the header size that inexpensive hardware forwarding can examine and modify. For future generations, this might likely be as much as 512 bytes, so to optimize hardware lookup it might be interesting to consider the option of carrying the addresses not consecutively, but carry them as

3.5. Example packet header encoding

The following encodings propose a couple of ideas that could be interesting in addressing.

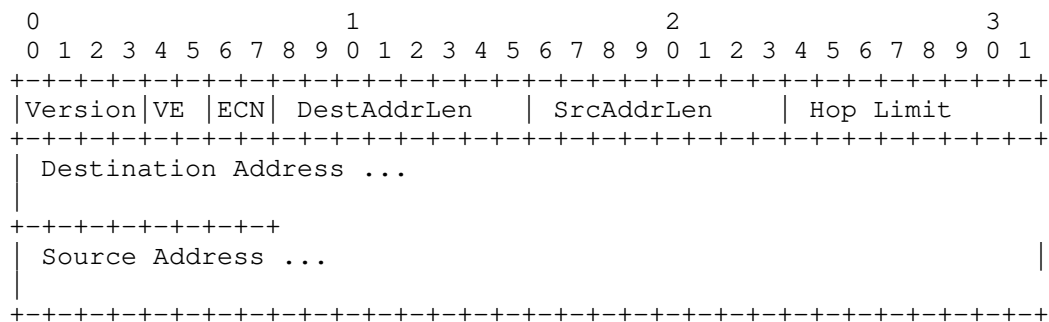


Figure 6: Example packet header encoding

Version: A version number for this packet header from the same registry as the IPv4/IPv6 version number field.

VE: Version Extension. 00. Reserved for future variations of the header, such as new extension header formats if desired, so as to not use up any more than one Version code point.

DestAddrLen: The length of the Destination Address field in bytes. Valid values are 1...255 bytes. One byte minimum length is mandatory because of the need to indicate some semantic for processing the packet.

SrcAddrLen: The length of the Source Address field in bytes. Valid values are 0...255 bytes. The Source Address field is therefore optional.

ECN: See [RFC3168] and the documents updating it.

Rsv: Reserved.

Hop Limit: As in IPv6

Beside the variable length of the Source and Destination address fields and hence their length indications, the difference to the IPv6 header are as follows:

Only the two ECN bits are maintained from the IPv4/IPv6 Traffic Class field. This is because in the majority of networks, the other 6 bits of Traffic Class, DSCP are not being used, and where QoS differentiation would be used, often additional or different QoS parameters may be required that are not supported by IPv4/IPv6. Such a new network header would thus be a great opportunity to improve on QoS header parameters through a better QoS extension header, where it is needed (outside scope of this document), and not proliferate not ubiquitously used elements in the base header. The same reason applies to removing the Flow Label field.

ECN on the other hand is very fundamental for the majority of all traffic in Internet and limited domain networks.

4. Inspirations

This section reviews prior addressing and networking technologies that did inspire this memo and compares it with them.

4.1. E.164

E.164 telephone numbers traditionally worked (and may still work) similar to this mechanisms handling of addresses by adding and removing prefixes and allowing to grow networks hierarchically.

In Germany for example a town/city might have had a subscriber numbering plan starting with 3 digit numbers and expanding over time into 5 digits. 0 was excluded as the first digit of any assigned number. Let our example subscriber have number 1234

When the phone systems of towns/cities where connected, dialing a different town/city would use a concatenation of the inter-city traffic discrimination code "0" followed by the dial code for the town/city, followed by the subscriber number. Let our example town dial code be 4111, the subscriber number dialed from a different city would be 04111 1234. Again, "0" was excluded as the first digit of a trunk prefix.

When finally the phone systems of countries where connected, dialing a different country would use a concatenation of the international traffic discrimination code "00" followed by the country dial code, which in our example is 49 for Germany followed by the dial code for the city, followed by the subscriber number - 0049 4111 1234 for our example subscriber. Note that this number would of course only work when calling from countries that also do use "00" as the international traffic discrimination code. When calling the number from the USA, one would have to dial 011 4111 1234, because the USA uses 011 as the internal traffic discrimination code.

Of course, understanding foreign countries traffic discrimination code rules to reverse engineer a foreign telephone number so as to translate it to the according rules of the calling-from country is one of the problems that is leading more and more subscribers to prefer the absolute E.164 telephone numbers like +49 4111 1234.

On the other hand, when the interplanetary telephone network will "soon" [I-D.draft-farrel-soon] arrive and there are not enough country codes available in Earth's existing numbering plan, one would have to find a way to attach prefixes in front of existing E.164 numbers, something that E.164 likely cannot afford, but which would be possible with UPVLA.

In our example the UPVLA address could be 0003 49 4111 1234 and a new solar system "absolute" address could be ++3 49 4111 1234.

Obviously, Mercury has to get 1, Venus 2 and Earth 3 and so on, so that it would be easier to remember how to dial other planets than it is now to remember how to dial other countries.

If one was to use the solution proposed in this memo to build the phone network addressing system with the example numbering plan, one could set up a multi-tiered internetwork as shown in Figure 7.

Soon:

```

.
. Solar System network .
.
. prefix "3" . |
. .... v strip 3 from dst, prepend 0 to dst
...| Planet Edge Node .... forward into global network
. .... ^ strip 0 from dst, prepend 3 to src
. prefix "0" . | forward into solar system network
.

```

Today:

```

.
. "global" network .
.
. prefix "49" . |
. +-----+ v strip 49 from dst, prepend 0 to dst
...| Country Edge Node |... forward into country network
. +-----+ ^ strip 0 from dst, prepend 49 to src
. prefix "0" . | forward into global network
.
. "country" network .
.
. prefix "4111" . |
. +-----+ v strip 4111 from dst, prepend 0 to dst
...| City Edge Node |... forward into city network
. +-----+ ^ strip 0 from dst, prepend 4111 to src
. prefix "0" . forward into country network
.
. city network .
.
. subscriber 1234 .
.....

```

Figure 7: Example internetwork for E.164 style address structure with FA-INAAS

Packets destined to an address starting with "0" would be routed to an edge node of the city network, which "owns" the "0" prefix, there that "0" prefix is stripped, but the cities own prefix of in the example "4111" is prepended to the source address, and then the packet is forwarded into the country network.

When a packet is received from the country network on a city edge node, the opposite happens, the cities own prefix, e.g.: 4111 is stripped from the destination address and 0 is prepended to the source address, then the packet is forwarded into the city network and routed to the destination. Which can generate return packets by just swapping source and destination addresses.

The same process will happen across 2 network tiers when a 00 prefix is used or even 3 network tiers, once we have (soon ;-) a Solar System Network.

Of course, each tier and each instance of each tier can choose its own addressing scheme and prefixes for the edge routers. It is left as an exercise to the reader for example to amend the example with its own home countries traffic discrimination codes.

4.2. MPLS

Adding/Removing or swapping prefixes is the core forwarding process step in Multiprotocol Label Switching [RFC3031]. Due to the time MPLS was designed, it had to have a very fixed size and functionality stack architecture, but as claimed in before, the author thinks that today an MPLS stack could easily be built just with the proposed addressing scheme address.

Compared to MPLS, the proposed scheme does not mandate that that every steering address needs to contain QoS (EXP) and TTL fields as are present in MPLS Label Stack entries, but of course they would be equally possible as parameters of appropriate functions.

Likewise the proposal does not think it is appropriate to require complicated scanning ahead into the address in search of Special Label Stack entries. Therefore, FA-IINAS would require that any per-hop accessible information that is not included in the hops function/parameters would have to be carried would have to be carried in a separated extensions header.

4.3. Segment Routing SR-MPLS / SRv6

FA-IINAS can support more compact packet steering than SR-MPLS when the prefixes are accordingly chosen to be shorter than the 32 bits for an LSE.

While it would be possible to emulate MPLS LSE by using prefixes of 20 bit and following them with 12 bit of functional parameters indicating EXP and TTL, the proposal in this memo does not assume that transit routers would be able to act on those EXP or TTL bits. While it would be easily possible to define such additional transit hop semantic through extensions to the control plane, the author believes that the per-path parameters of TTL in a base header and more flexible QoS in an extension header is the more likely most useful option for these two functions.

In comparison to SRv6, FA-IINAS allows of course more compact representation of steering hops and also more easily few or many per-hop bits for programmability, as desired.

What FA-IINAS does not provide for is to keep the sequence of steering addresses in the header up to the final receiver. This might be useful for diagnostics, but it is seemingly not so important that it did stop the adoption of SR-MPLS, where the steering hops are likewise removed from the packet header when steering happens.

Other functions than steering and per-steering hop programmability provided by SRv6 via SRH (such as its TLV for the receiver) are unaffected by this proposal and could equally be provided for by an SRH style extension header without the source routing part.

4.4. Research

[Haoyu] proposes a hierarchical addressing scheme and provides reviews in a lot more detail a set of other reasons for such addressing scheme. That paper does not allow for arbitrary composition of networks without a clear hierarchy or root thereof, as FA-IINAS does.

5. Summary and conclusions

This memo introduces a simple but hopefully very attractive addressing scheme that leverages variable length address sizes with the potential for simple address prefix processing (push/pop/swap) on steering hops, service-function hops and especially network edge nodes.

Push/pop/swap of network prefixes on network edge nodes allows to introduce a non-global internetworking address architecture that should make it a lot easier to build and manage many embedded, private or otherwise limited domain internetworks and optimize forwarding engines for a variety of different of these type of networks such as through known maximum network prefix lengths.

When network addresses as in FA-IINAS become effectively internetwork path addresses, they also allow for a much wider range of possible routing policies. In a time where the classical Internet with its "sender just gets one path", this can be a highly beneficial enhancement to explore (not that this was already proposed, maybe way ahead of its time and with vastly different mechanisms in solutions as early as [RFC1621], [RFC1622]).

In this version of the memo, these are only limited considerations about how to refine details of the proposal to find incremental, near-term deployment options, for example by using existing IPv6 headers and using an unassigned prefix to define FA-IINAS addressing semantic into it (limited of course to 128 bit then). These type of considerations can be subject for future revisions of this memo.

6. Changelog

00: Initial version

01: Refresh, new co-author

7. Informative References

[CiscoNAT] Akkiraju, P., Delgadillo, K., and Y. Rekhter, "Enabling Enterprise Multihoming with Cisco IOS Network Address Translation (NAT)", 2000, <http://staff.ustc.edu.cn/~james/cisco/nat/emios_wp.htm>.

[Haoyu] Song, H., Zhang, Z., Qu, Y., and J. Guichard, "Adaptive Addresses for Next Generation IP Protocol in Hierarchical Networks", IEEE 2020 IEEE 28th International Conference on Network Protocols (ICNP), n.d..

[I-D.draft-farrel-soon] Farrel, A., "A Definition of the Term "Soon" for Use in Discussions with Working Group Chairs and Area Directors", Work in Progress, Internet-Draft, draft-farrel-soon-07, 8 March 2021, <<https://www.ietf.org/archive/id/draft-farrel-soon-07.txt>>.

[I-D.jia-flex-ip-address-structure] Jia, Y., Chen, Z., and S. Jiang, "Flexible IP: An Adaptable IP Address Structure", Work in Progress, Internet-Draft, draft-jia-flex-ip-address-structure-00, 31 October 2020, <<https://www.ietf.org/archive/id/draft-jia-flex-ip-address-structure-00.txt>>.

- [I-D.jia-intarea-scenarios-problems-addressing]
Jia, Y., Trossen, D., Iannone, L., Shenoy, N., Mendes, P., 3rd, D. E. E., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-scenarios-problems-addressing-02, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-scenarios-problems-addressing-02.txt>>.
- [I-D.king-irtf-challenges-in-routing]
King, D. and A. Farrel, "Challenges for the Internet Routing Infrastructure Introduced by Changes in Address Semantics", Work in Progress, Internet-Draft, draft-king-irtf-challenges-in-routing-03, 14 June 2021, <<https://www.ietf.org/archive/id/draft-king-irtf-challenges-in-routing-03.txt>>.
- [I-D.king-irtf-semantic-routing-survey]
King, D. and A. Farrel, "A Survey of Semantic Internet Routing Techniques", Work in Progress, Internet-Draft, draft-king-irtf-semantic-routing-survey-02, 28 June 2021, <<https://www.ietf.org/archive/id/draft-king-irtf-semantic-routing-survey-02.txt>>.
- [RFC1621] Francis, P., "Pip Near-term Architecture", RFC 1621, DOI 10.17487/RFC1621, May 1994, <<https://www.rfc-editor.org/info/rfc1621>>.
- [RFC1622] Francis, P., "Pip Header Processing", RFC 1622, DOI 10.17487/RFC1622, May 1994, <<https://www.rfc-editor.org/info/rfc1622>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.

- [RFC3306] Haberman, B. and D. Thaler, "Unicast-Prefix-based IPv6 Multicast Addresses", RFC 3306, DOI 10.17487/RFC3306, August 2002, <<https://www.rfc-editor.org/info/rfc3306>>.
- [RFC3956] Savola, P. and B. Haberman, "Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address", RFC 3956, DOI 10.17487/RFC3956, November 2004, <<https://www.rfc-editor.org/info/rfc3956>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", RFC 4607, DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.
- [RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <<https://www.rfc-editor.org/info/rfc6550>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", RFC 8279, DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [RFC8296] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Tantsura, J., Aldrin, S., and I. Meilik, "Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks", RFC 8296, DOI 10.17487/RFC8296, January 2018, <<https://www.rfc-editor.org/info/rfc8296>>.

- [RFC8520] Lear, E., Droms, R., and D. Romascanu, "Manufacturer Usage Description Specification", RFC 8520, DOI 10.17487/RFC8520, March 2019, <<https://www.rfc-editor.org/info/rfc8520>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <<https://www.rfc-editor.org/info/rfc8990>>.
- [RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <<https://www.rfc-editor.org/info/rfc8994>>.

Authors' Addresses

Toerless Eckert
Futurewei Technologies USA
Santa Clara, CA 95050
United States of America

Email: tte@cs.fau.de

Nirmala Shenoy
Rochester Institute of Technology
New York, NY 14623
United States of America

Email: nxsvks@rit.edu

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: 27 April 2022

J. Henry
Cisco Systems
Y. Lee
Comcast
24 October 2021

Randomized and Changing MAC Address Use Cases
draft-henry-madinas-framework-03

Abstract

To limit the association between a device traffic and its user, client vendors have started implementing MAC address rotation. When such rotation happens, some in-network states may break, which may affect network efficiency and the user experience. At the same time, devices may continue sending other stable identifiers, defeating the MAC rotation purposes. This document lists various network environments and a set of network services that may be affected by such rotation. This document then examines settings where the user experience may be affected by in-network state disruption, and settings where other machine identifiers may expose the user privacy. Last, this document examines solutions to maintain user privacy while preserving user quality of experience and network operation efficiency.

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 <https://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 27 April 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
1.1. Requirements Language	3
2. MAC Address as an Identity: User vs. Device	3
3. The Actors: Network Functional Entities and Human Entities .	6
3.1. Network Functional Entities	6
3.2. Human-related Entities	7
3.3. The Trust and the Environments	8
3.4. The Purpose of Device Identification and Associated Problems	10
3.5. Scenario Mapping Table	12
3.6. Requirements Formulation	13
4. IANA Considerations	14
5. Security Considerations	14
6. Normative References	14
7. Informative References	14
Authors' Addresses	15

1. Introduction

It has become easier for attackers to track the activity of a personal device, particularly when traffic is sent over a wireless link. Once the association between a device and its user is made, identifying the device and its activity is sufficient to deduce information about what the user is doing, without the user consent.

To reduce the risks of correlation between a device activity and its owner, multiple vendors have started to implement Randomized and Changing MAC addresses (RCM). With this scheme, an end-device implements a different RCM over time when exchanging traffic over a wireless network. By randomizing the MAC address, the association between a given traffic flow and a single device is made more difficult, assuming no other visible unique identifiers are in use.

However, such address change may affect the user experience and the efficiency of legitimate network operations. For a long time, the unicity of the association between a device and a MAC address was assumed, despite the emergence of tools to flush out the MAC address

to bypass some network policies. When this association is broken, elements of network communication may also break. For example, sessions established between the end-device and network services may be lost and packets in translation may suddenly be without clear source or destination. As multiple clients implements fast-paced RCM rotations, network services may be over-solicited by a small number of stations that appear as many clients.

At the same time, some network services rely on the client station providing an identifier, which can be the MAC address or another value. If the client implements MAC rotation but continues sending the same static identifier, then the association between a stable identifier identifier and the station continues despite the RCM scheme. There may be environments where such continued association is desirable, but others where the user privacy has more value than any continuity of network service state.

There is a need to enumerate services that may be affected by RCM, and evaluate possible solutions to maintain both the quality of user experience and network efficiency while RCM happens and user privacy is reinforced. This document presents such assessment and recommendations.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC 2119 [RFC2119] RFC 8174 [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. MAC Address as an Identity: User vs. Device

Any device member of an IEEE 802 network [IEEE.802-1D.1993] includes several operating layers. Among them, the Media Access Control (MAC) layer defines rules to control how the device accesses the shared medium. In a network where a machine can communicate with one or more other machines, one such rule is that each machine needs to be identified, either as the target destination of a message, or as the source of a message (and thus the target destination of the answer). Initially intended as a 48-bit (6 octets) value, later versions of the Standard [IEEE.802.15.4P_2014] allowed this address to take an extended format of 64 bits (8 octets), thus enabling a larger number of MAC addresses to coexist as the 802 technologies became widely adopted.

Regardless of the address length, different networks have different needs, and several bits of the first octet are reserved for specific purposes. In particular, the first bit is used to identify the destination address either as an individual (bit set to 0) or a group address (bit set to 1). The second bit, called the Universally or Locally Administered (U/L) Address Bit, indicates whether the address has been assigned by a local or universal administrator. Universally administered addresses have this bit set to 0. If this bit is set to 1, the entire address (i.e., 48 bits) has been locally administered [IEEE.802-1Y.1990] Section 5.2.1.

The intent of this provision is important for the present document. The 802 Standard recognized that some devices may never travel and thus, always attaching to the same network, would not need a globally unique MAC address. To accommodate for this relaxed requirement, the second bit of the MAC address first octet the MAC address format was designed to express whether the address was intended to be globally unique, or if significance was only local. The address allocation method was not defined in the Standard in this later case, but the mechanism was defined in the same clause that defined that an address should be unique so as to avoid collision.

It is also important to note that the purpose of the Universal version of the address was to avoid collisions and confusion, as any machine could connect to any network, and each machine needs to determine if it is the intended destination of a message or its response. The same clause 5.2.1 reminds network designers and operators that all potential members of a network need to have a unique identifier (if they are going to coexist in the network). The advantage of a universal address is that a node with such an address can be attached to any Local Area Network (LAN) in the world with an assurance that its address is unique.

With the rapid development of wireless technologies and mobile devices, this scenario became very common. With a vast majority of 802 networks implementing radio technologies at the access, the MAC address of a wireless device can appear anywhere on the planet and collisions should still be avoided. However, the same evolution brought the distinction between two types of devices that the 802 Standard generally referred to as 'nodes in a network'. Their definition is found in the 802E Recommended Practice (clause 6.2). One type is a shared service device, which functions are used by a number of people large enough that the device itself, its functions or its traffic cannot be associated with a single or small group of people. Examples of such devices include switches in a dense network, 802.11 (WLAN) access points in a crowded airport, task-specific (e.g. barcode scanners) devices, etc. Another type is a personal device, which is a machine, a node, primarily used by a

single person or small group of people, and so that any identification of the device or its traffic can also be associated to the identification of the primary user or their traffic. Quite naturally, the unique identification of the device is trivial if the device expresses a universally unique MAC address. Then, the detection of elements directly or indirectly identifying the user of the device (Personally Identifiable Information, or PII) is sufficient to tie the universal MAC address to a user. Then, any detection of traffic that can be associated to the device becomes also associated with the known user of that device (Personally Correlated Information, or PCI).

This possible identification or association presents a serious privacy issue, especially with wireless technologies. For most of them, and in particular for 802.11, the source and destination MAC addresses are not encrypted even in networks that implement encryption (so that each machine can easily detect if it is the intended target of the message before attempting to decrypt its content, and also identify the transmitter, so as to use the right key when multiple unicast keys are in effect).

This unique identification of the user associated to a node was clearly not the intent of the 802 MAC address. A logical solution to remove this association is to use a locally administered address instead, and change the address in a fashion that prevents a temporal association between one MAC address and some PII to be maintained fruitfully. However, other network devices on the same LAN implementing a MAC layer also expect the unicity of the MAC address. When a device changes its MAC address, other devices on the same LAN may fail to recognize that the same machine is attempting to communicate with them. Additionally, multiple layers implemented at upper OSI layers have been designed with the assumption that each node on the LAN, using these services, would have a unique MAC address. This assumption sometimes adds to the PII confusion, for example in the case of Authentication, Association and Accounting (AAA) services authenticating the user of a machine and associating the authenticated user to the device MAC address. Other services solely focus on the machine (e.g. DHCP), but still expect each device to use a single MAC address. Changing the MAC address may disrupt these services.

3. The Actors: Network Functional Entities and Human Entities

The risk of service disruption is thus weighted against the privacy benefits. However, the plurality of actors involved in the exchanges tends to blur the boundaries of what privacy should be protected against. It might therefore be useful to list the actors to the network exchanges. Some actors are functional entities, some others are humans (or related) entities.

3.1. Network Functional Entities

Wireless access network infrastructure devices (e.g. WLAN access points or controllers): these devices participate in 802 LAN operations. As such, they need to uniquely identify machines as a source or destination so as to successfully continue exchanging frames. Part of the identification includes recording, and adapting to, devices communication capabilities (e.g. support for specific protocols). As a device changes its network attachment (roams) from one access point to another, the access points can exchange contextual information (e.g. device MAC, keying material) allowing the device session to continue seamlessly. These access points can also inform devices further in the wired network about the roam, to ensure that OSI model Layer 2 frames are redirected to the new device access point.

Other network devices operating at the MAC layer: many wireless network access devices (e.g., 802.11 access points) are conceived as Layer 2 devices, and as such they bridge a frame from one medium (e.g., 802.11 or Wi-Fi) to another (e.g., 802.3 or Ethernet). This means that a wireless device MAC address often exists on the wire beyond the wireless access device. Devices connected to this wire also implement 802 technologies, and as such operate on the expectation that each device is associated to a unique MAC address for the duration of continuous exchanges. For example, switches and bridges associate MAC addresses to individual ports (so as to know which port to send a frame intended for a particular MAC address). Similarly, authentication, authorization and accounting (AAA) services can validate the identity of a device and use the device MAC address as a first pointer to the device identity (before operating further verification). Similarly, some networking devices offer Layer-2 filtering policies that may rely on the connected MAC addresses. 802.1X-enabled devices may also selectively block the data portion of a port until a connecting device is authenticated. These services then use the MAC address as a first pointer to the device identity to allow or block data traffic. This list is not exhaustive. Multiple services are defined for 802.3 networks, and multiple services defined by the IEEE 802.1 working group are also applicable to 802.3 networks. Wireless access points may also

connect to other mediums than 802.3, which also implements mechanism under the umbrella of the general 802 Standard, and therefore expect the unique association of a MAC address to a device.

Network devices operating at upper layers: some network devices provide functions and services above the MAC layer. Some of them also operate a MAC layer function: for example, routers provide IP forwarding services, but rely on the device MAC address to create the appropriate frame structure. Other devices and services operate at upper layers, but also rely upon the 802 principle of unique MAC-to-device mapping. For example, DHCPv4 services commonly provide a single IP address per MAC address (they do not assign more than one IPv4 address per MAC address, and assign a new IPv4 address to each new requesting MAC address). ARP and reverse-ARP services commonly expect that, once an IP-to-MAC mapping has been established, this mapping is valid and unlikely to change for the cache lifetime. DHCPv6 services commonly do not assign the same IPv6 address to two different requesting MAC addresses. Hybrid services, such as EoIP, also assume stability of the device-to-MAC-and-IP mapping for the duration of a given session.

3.2. Human-related Entities

Over the air (OTA) observers: as the transmitting or receiving MAC address is usually not encrypted in wireless 802-technologies exchanges, and as any protocol-compatible device in range of the signal can read the frame header, OTA observers are able to read individual transmissions MAC addresses. Some wireless technologies also support techniques to establish distances or positions, allowing the observer, in some cases, to uniquely associate the MAC address to a physical device and its associated location. It can happen that an OTA observer has a legitimate reason to monitor a particular device, for example for IT support operations. However, it is difficult to control if another actor also monitors the same station with the goal of obtaining PII or PCI.

Wireless access network operators: some wireless access networks are only offered to users or devices matching specific requirements, such as device type (e.g., IoT-only networks, factory operational networks). Therefore, operators can attempt to identify the devices (or the users) connecting to the networks under their care. They can use the MAC address to represent an identified device.

Network access providers: wireless access networks are often considered beyond the first 2 layers of the OSI model. For example, several regulatory or legislative bodies can group all OSI layers into their functional effect of allowing network communication between machines. In this context, entities operating access

networks can see their liability associated to the activity of devices communicating through the networks that these entities operate. In other contexts, operators assign network resources based on contractual conditions (e.g., fee, bandwidth fair share). In these scenarios, these operators may attempt to identify the devices and the users of their networks. They can use the MAC address to represent an identified device.

Over the wire internal (OTWi) observers: because the device wireless MAC address continues to be present over the wire if the infrastructure connection device (e.g. access point) functions as a Layer 2 bridge, observers may be positioned over the wire and read transmission MAC addresses. Such capability supposes that the observer has access to the wired segment of the broadcast domain where the frames are exchanged. In most networks, such capability requires physical access to an infrastructure wired device in the broadcast domain (e.g. switch closet), and is therefore not accessible to all.

Over the wired external (OTWe) observers: beyond the broadcast domain, frames headers are removed by a routing device, and a new Layer 2 header is added before the frame is transmitted to the next segment. The personal device MAC address is not visible anymore, unless a mechanism copies the MAC address into a field that can be read while the packet travels onto the next segment (e.g. pre-[RFC4941] and pre-[RFC7217] IPv6 addresses built from the MAC address). Therefore, unless this last condition exists, OTWe observers are not able to see the device MAC address.

3.3. The Trust and the Environments

The surface of PII exposures that can drive MAC address randomization depends on the environment where the device operates, on the presence and nature of other devices in the environment, and on the type of network the device is communicating through. Therefore, a device can express an identity (such as a MAC address) that can be stable over time if trust with the environment is established, or that can be temporal if an identity is required for a service in an environment where trust has not been established. Trust is not a binary currency. Thus it is useful to distinguish what trust a personal device may establish with the different entities at play in a L2 domain:

1. Full trust: there are environments where a personal device establishes a trust relationship and can share a stable device identity with the access network devices (e.g., access point and WLC), the services beyond the access point in the L2 broadcast domain (e.g. DHCP, AAA). The personal device (or its user) has confidence that its identity is not shared beyond the L2 broadcast domain boundary.
2. Selective trust: in other environments, the device may not be willing to share a stable identity with some elements of the Layer 2 broadcast domain, but may be willing to share a stable identity with other elements. For example, a device may want to change the MAC address it uses to communicate with the access point while maintaining the same IP address across the MAC address rotation (thus expressing a stable identity to the DHCP server). That stable identity may or may not be the same for different services.
3. Zero trust: in other environments, the device may not be willing to share any stable identity with any entity reachable through the Layer 2 broadcast domain, and may express a temporal identity to each of them. That temporal identity may or not be the same for different services.

This trust relationship naturally depends on the relationship between the user of the personal device and the operator of the service. Thus, it is useful to observe the typical trust structure of common environments:

- A. Residential settings under the control of the user: this is typical of a home network with Wi-Fi in the LAN and Internet connection. In this environment, the MAC address activity may be detectable beyond the home walls. However, if traffic is encrypted (e.g. WPA3), some protection for OTA eavesdropping can be assumed. The wire segment within the broadcast domain is under the control of the user, and is therefore usually not at risk of hosting an eavesdropper. Full trust is typically established at this level. The device trusts the access point and all L2 domain entities beyond the access point. Traffic over the Internet does not expose the MAC address if it is not copied to another field before routing happens.
- B. Managed residential settings: examples of this type of environment include shared living facilities and other collective environments where an operator manages the network for the residents. The OTA exposure is similar to that of a home. A number of devices larger than in a standard home may be present, and the operator may be requested to provide IT support to the

residents. Therefore, the operator may need to identify a device activity in real time, but may also need to analyze logs so as to understand a past reported issue. For both activities, a device identification associated to the session is needed. Full trust is often established in this environment, at the scale of a series of a few sessions.

- C. Public guest networks: public hotspots, such as in shopping malls, hotels, stores, trains stations and airports are typical of this environment. The guest network operator may be legally mandated to identify devices or users or may have the option to leave all devices and users untracked. In this environment, trust is commonly not established with any element of the L2 broadcast domain (Zero trust model by default).
- D. Enterprises (with BYOD): users may be provided corporate devices or may bring their own devices. The devices are not directly under the control of a corporate IT team. Trust may be established as the device joins the network. Some enterprise models will mandate full trust, others, considering the BYOD nature of the device, will allow selective trust.
- E. Managed enterprises: in this environment, users are typically provided with corporate devices, and all connected devices are managed, for example through a Mobile Device Management (MDM) profile installed on the device. Full trust is created as the MDM profile is installed.

3.4. The Purpose of Device Identification and Associated Problems

Many network functional devices offering a service to a personal device use the device MAC address to maintain service continuity.

Wireless access points and controllers use the MAC address to validate the device connection context, including protocol capabilities, confirmation that authentication was completed, QoS or security profiles, encryption key material. Some advanced access points and controllers also include upper layer functions which purpose is covered below. A device changing its MAC address, without another recorded device identity, would cause the access point and the controller to lose these parameters. As such, the Layer 2 infrastructure does not know that the device (with its new MAC address) is authorized to communicate through the network. The encryption keying material is not identified anymore (causing the access point to fail decrypting the device traffic, and fail selecting the right key to send encrypted traffic to the device). In short, the entire context needs to be rebuilt, and a new session restarted. The time consumed by this procedure breaks any flow that

needs continuity or short delay between packets on the device (e.g. real-time audio, video, AR/VR etc.) The 802.11i Standard recognizes that a device may leave the network and come back after a short time window. As such, the standard suggests that the infrastructure should keep the context for a device up to one hour after the device was last seen. MAC address rotation in this context can cause resource exhaustion on the wireless infrastructure and the flush of contexts, including for devices that are simply in temporal sleep mode.

Other devices in the Layer 2 broadcast domain also use the MAC address to know whether and where to forward frames. MAC rotation can cause these devices to exhaust their resources, holding in memory traffic for a device which port location can no longer be found. As these infrastructure devices also implement a cache (to remember the port position of each known device), too frequent MAC rotation can cause resources exhaustion and the flush of older MAC addresses, including for devices that did not rotate their MAC. For the RCM device, these effects translate into session discontinuity and return traffic losses.

In wireless contexts, 802.1X authenticators rely on the device and user identity validation provided by a AAA server to open their port to data transmission. The MAC address is used to verify that the device is in the authorized list, and the associated key used to decrypt the device traffic. A change in MAC address causes the port to be closed to the device data traffic until the AAA server confirms the validity of the new MAC address. Therefore, MAC rotation can interrupt the device traffic, and cause a strain on the AAA server.

DHCP servers, without a unique identification of the device, lose track of which IP address is validly assigned. Unless the RCM device releases the IP address before the rotation occurs, DHCP servers are at risk of scope exhaustion, causing new devices (and RCM devices) to fail to obtain a new IP address. Even if the RCM device releases the IP address before the rotation occurs, the DHCP server typically holds the released IP address for a certain duration, in case the leaving MAC would return. As the DHCP server cannot know if the release is due to a temporal disconnection or a MAC rotation, the risk of scope address exhaustion exists even in cases where the IP address is released.

Routers keep track of which MAC address is on which interface. MAC rotation can cause MAC address cache exhaustion, but also the need for frequent ARP and inverse ARP exchanges.

In residential settings (environments type A), policies can be in place to control the traffic of some devices (e.g. parental control, block-list devices). These policies are often based on the device MAC address. Rotation of the MAC address removes the possibility for such control.

In residential settings (environments type A) and in enterprises (environments types D and E), device recognition and ranging may be used for IoT-related functionalities (door unlock, preferred light and temperature configuration, etc.) These functions often rely on the detection of the device wireless MAC address. MAC address rotation breaks the services based on such model.

In managed residential settings (environments types B) and in enterprises (environments types D and E), the network operator is often requested to provide IT support. With MAC address rotation, real time support is only possible if the user is able to provide the current MAC address. Service improvement support is not possible if the MAC address that the device had at the (past) time of the reported issue is not known at the time the issue is reported.

3.5. Scenario Mapping Table

Section 3.4 discusses different environments, different settings, and the expectations of users and network operators. Table 1 summarizes the expected degree of trust, network admin responsibility, complexity of supported network services and network support expectation from the user

Network Location	Trust Degree	Network Admin	Network Services	Network Support Expectation
Home	High	User	Medium	Low
Campus (BYOD)	Medium	IT	Complex	Medium
Enterprise (MDM)	High	IT	Complex	High
Hospitality	Low	IT	Simple	Medium
Public WiFi	Low	ISP	Simple	Low

Table 1: Scenario Mapping Table

For example: a Home network is considered to be trusted and safe. Users expect a simple procedure to connect to their home network. All devices in the home network trust each other. Privacy within the Layer 2 domain is not a major concern. The Home network can also include many IoT devices, which need to be simple to onboard and manage. The home user commonly expects the network operator to protect the home network from external threats (attacks from the Internet). The home user also commonly expects simple policy features (e.g., Parental Control). Most home users do not expect to need networking skills to manage their home network.

On the other end of the spectrum, Public Wi-Fi is often considered to be untrusted. Privacy is the number one concern for the user. Most users connect to Public Wi-Fi only require simple Internet connectivity service, and expect only limited to no technical support.

3.6. Requirements Formulation

The section describes the requirements for Randomized MAC-Address Changes:

- REQ1 The network must not make any assumption about client MAC address persistence. MAC address change must happen while allowing for service continuity. If a service is interrupted during the RCM process, there must be a formal mechanism for the client and the network to exchange about the interruption.
- REQ2 During duration of the services, the device should not change the identity. Any change of identity may result re-authentication and interrupt of the current network services.
- REQ3 Survey the current standards that use MAC address as a device identifier in the protocol. Make recommendation to the working groups to remove the dependency.
- REQ4 Work as liaison with external standard bodies such as IEEE, BBF and WBA to align with use cases and requirements.
- REQ5 Identify a secure mechanism to authenticate and exchange network identity to the device.
- REQ6 Identify a secure mechanism to inform the device about the type of network the device is connecting to (e.g. public Wi-Fi, enterprise, home), allowing the user to select the device identity (or identities) accordingly.

REQ7 Identify a secure mechanism for the network to request device identity. Upon successful authentication, the network may provide the device a temporary network-based marker to use the network services.

REQ8 Identify a secure mechanism for the device to notify the network prior to update the MAC address.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

Privacy considerations are discussed throughout this document.

6. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", BCP 72, RFC 3552, DOI 10.17487/RFC3552, July 2003, <<https://www.rfc-editor.org/info/rfc3552>>.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", RFC 5226, DOI 10.17487/RFC5226, May 2008, <<https://www.rfc-editor.org/info/rfc5226>>.

7. Informative References

- [IEEE.802-1D.1993] Institute of Electrical and Electronics Engineers, "Information technology - Telecommunications and information exchange between systems - Local area networks - Media access control (MAC) bridges", IEEE Standard 802.1D, July 1993.
- [IEEE.802-1Y.1990] Institute of Electrical and Electronics Engineers, "Source Routing Tutorial for End System Operation", IEEE Standard 802.1Y, September 1990.

- [IEEE.802.15.4P_2014]
IEEE, "IEEE Standard for local and metropolitan area networks - Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) - Amendment 7: Physical Layer for Rail Communications and Control (RCC)", IEEE 802.15.4p-2014, DOI 10.1109/ieeestd.2014.6809836, 2 May 2014, <<http://ieeexplore.ieee.org/servlet/opac?punumber=6809834>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://www.rfc-editor.org/info/rfc4941>>.
- [RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, DOI 10.17487/RFC5176, January 2008, <<https://www.rfc-editor.org/info/rfc5176>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

Authors' Addresses

Jerome Henry
Cisco Systems
United States of America

Email: jerhenry@cisco.com

Yiu L. Lee
Comcast
1800 Arch Street
Philadelphia, PA 19103
United States of America

Email: yiul_lee@comcast.com

Internet Area Working Group
Internet-Draft
Intended status: Informational
Expires: 7 September 2022

Y. Jia
D. Trossen
L. Iannone
Huawei
P. Mendes
Airbus
N. Shenoy
R.I.T.
L. Toutain
IMT-Atlantique
A. Y. Chen
Avinta
D. Farinacci
lispers.net
6 March 2022

Gap Analysis in Internet Addressing
draft-jia-intarea-internet-addressing-gap-analysis-02

Abstract

There exist many extensions to Internet addressing, as it is defined in [RFC0791] for IPv4 and [RFC8200] for IPv6, respectively. Those extensions have been developed to fill gaps in capabilities beyond the basic properties of Internet addressing. This document outlines those properties as a baseline against which the extensions are categorized in terms of methodology used to fill the gap together with examples of solutions doing so.

While introducing such extensions, we outline the issues we see with those extensions. This ultimately leads to consider whether or not a more consistent approach to tackling the identified gaps, beyond point-wise extensions as done so far, would be beneficial. The benefits are the ones detailed in the companion document [I-D.jia-intarea-scenarios-problems-addressing], where, leveraging on the gaps identified in this memo and scenarios provided in [I-D.jia-intarea-scenarios-problems-addressing], a clear problem statement is provided.

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 <https://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 7 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Properties of Internet Addressing	4
2.1. Property 1: Fixed Address Length	4
2.2. Property 2: Ambiguous Address Semantic	4
2.3. Property 3: Limited Address Semantic Support	5
3. Filling Gaps through Extensions to Internet Addressing Properties	5
3.1. Length Extensions	5
3.1.1. Shorter Address Length	6
3.1.2. Longer Address Length	8
3.1.3. Summary	10
3.2. Identity Extensions	10
3.2.1. Anonymous Address Identity	11
3.2.2. Authenticated Address Identity	14
3.2.3. Summary	15
3.3. Semantic Extensions	16
3.3.1. Utilizing Extended Address Semantics	17
3.3.2. Utilizing Existing or Extended Header Semantics	20
3.3.3. Summary	23
4. Overview of Approaches to Extend Internet Addressing	24

5. A System View on Address	26
6. Issues in Extensions to Internet Addressing	27
6.1. Limiting Address Semantics	27
6.2. Complexity and Efficiency	27
6.2.1. Repetitive encapsulation	28
6.2.2. Compounding issues with header compression	29
6.2.3. Introducing Path Stretch	29
6.2.4. Complicating Traffic Engineering	29
6.3. Security	30
6.4. Fragility	30
7. Summary of issues	31
8. Conclusions	33
9. Security Considerations	34
10. IANA Considerations	34
11. Informative References	34
Acknowledgments	44
Authors' Addresses	44

1. Introduction

[I-D.jia-intarea-scenarios-problems-addressing] outlines scenarios and problems in Internet addressing through presenting a number of cases of communication that have emerged over the many years of utilizing the Internet and for which various extensions to the network interface-centric addressing of IPv6 have been developed. In order to continue the discussion on the emerging needs for addressing, initiated with [I-D.jia-intarea-scenarios-problems-addressing], this memo aims at identifying gaps between the Internet addressing model and desirable features that have been added by various extensions, in various contexts.

The approach to identifying the gaps is guided by key properties of Internet addressing, outlined in Section 2, namely (i) the fixed length of the IP addresses, (ii) the ambiguity of IP addresses semantic, while still (iii) providing limited IP address semantic support. Those properties are derived directly as a consequence of the respective standards that provide the basis for Internet addressing, most notably [RFC0791] for IPv4 and [RFC8200] for IPv6, respectively.

Those basic properties, and the potential issues that arise from those properties, give way to extensions that have been proposed over the course of deploying new Internet technologies. Section 3 discusses those extensions, summarized as gaps against the basic properties in Section 4.

Finally, this memo outlines issues that arise with the extension-driven approach to the basic Internet addressing, discussed in Section 6, arguing that any requirements for solutions that would revise the basic Internet addressing would require to address those issues.

2. Properties of Internet Addressing

As the Internet Protocol adoption has grown towards the global communication system we know today, its characteristics have evolved subtly, with [RFC6250] documenting various aspects of the IP service model and its frequent misconceptions, including Internet addressing. In this section, the three most acknowledged properties related to `_Internet addressing_` are detailed. Those are (i) fixed IP address length, (ii) ambiguous IP address semantic, and (iii) limited IP address semantic support.

Section 3 elaborates on various extensions that aim to expand Internet addressing beyond those properties; those extensions are positioned as intentions to close perceived gaps against those key properties.

2.1. Property 1: Fixed Address Length

The fixed IP address length is specified as a key property of the design of Internet addressing, with 32 bits for IPv4 ([RFC0791]), and 128 bits for IPv6 ([RFC8200]), respectively. Given the capability of the hardware at the time of IPv4 design, a fixed length address was considered as a more appropriate choice for efficient packet forwarding. Although the address length was once considered to be variable during the design of Internet Protocol Next Generation ("IPng", cf., [RFC1752]) in the 1990s, it finally inherited the design of IPv4 and adopted a fixed length address towards the current IPv6. As a consequence, the 128-bit fixed address length of IPv6 is regarded as a balance between fast forwarding (i.e., fixed length) and practically boundless cyberspace (i.e., enabled by using 128-bit addresses).

2.2. Property 2: Ambiguous Address Semantic

Initially, the meaning of an IP address has been to identify an interface on a network device, although, when [RFC0791] was written, there were no explicit definitions of the IP address semantic.

With the global expansion of the Internet protocol, the semantic of the IP address is commonly believed to contain at least two notions, i.e., the explicit 'locator', and the implicit 'identifier'. Because of the increasing use of IP addresses to both identify a node and to

indicate the physical or virtual location of the node, the intertwined address semantics of identifier and locator was then gradually observed and first documented in [RFC2101] as 'locator/identifier overload' property. With this, the IP address is used as an identification for host and server, very often directly used, e.g., for remote access or maintenance.

2.3. Property 3: Limited Address Semantic Support

Although IPv4 [RFC0791] did not add any semantic to IP addresses beyond interface identification (and location), time has proven that additional semantics are desirable (c.f., the history of 127/8 [HISTORY127] or the introduction of private addresses [RFC1918]), hence, IPv6 [RFC4291] introduced some form of additional semantics based on specific prefix values, for instance link-local addresses or a more structured multicast addressing. Nevertheless, systematic support for rich address semantics remains limited and basically prefix-based.

3. Filling Gaps through Extensions to Internet Addressing Properties

Over the years, a plethora of extensions has been proposed in order to move beyond the native properties of IP addresses, outlined in the previous section. The development of those extensions can be interpreted as filling gaps between the original properties of Internet addressing and desired new capabilities that those developing the extensions identified as being missing and yet needed and desirable.

3.1. Length Extensions

Extensions in this subsection aim at extending the property described in Section 2.1, i.e., the fixed IP address length.

When IPv6 was designed, the main objective was to create an address space that would not lead to the same situation as IPv4, namely to address exhaustion. To this end, while keeping the same addressing model like IPv4, IPv6 adopted a 128-bit address length with the aim of providing a sufficient and future-proof address space. The choice was also founded on the assumption that advances in hardware and Moore's law would still allow to make routing and forwarding faster, and the IPv6 routing table manageable.

We observe, however, that the rise of new use cases but also the number of new, e.g., industrial/home or small footprint devices, was possibly unforeseen. Sensor networks and more generally the Internet of Things (IoT) emerged after the core body of work on IPv6, thus different from IPv6 assumptions, 128-bit addresses were costly in

certain scenarios. On the other hand, given the huge investments that IPv6 deployment involved, certain solutions are expected to increase the addressing space of IPv4 in a compatible way, and thus extend the lifespan of the sunk investment on IPv4.

At the same time, it may also be possible to use variable and longer address lengths to address current networking demands. For example in content delivery networks, longer addresses such as URLs are required to fetch content, an approach that Information-Centric Networking (ICN) applied for any data packet sent in the network, using information-based addressing at the network layer. Furthermore, as an approach to address the routing challenges faced in the Internet, structured addresses may be used in order to avoid the need for routing protocols. Using variable length addresses allow as well to have shorter addresses. So for requirements for smaller network layer headers, shorter addresses could be used, maybe alleviating the need to compress other fields of the header. Furthermore, transport layer port numbers can be considered short addresses, where the high order bits of the extended address is the public IP of a NAT. Hence, in IoT deployments, the addresses of the devices can be really small and based on the port number, but they all share the global address of the gateway to make each one have a globally unique address.

3.1.1. Shorter Address Length

3.1.1.1. Description:

In the context of IoT [RFC7228], where bandwidth and energy are very scarce resources, the static length of 128-bit for an IP address is more a hindrance than a benefit since 128-bit for an IP address may occupy a lot of space, even to the point of being the dominant part of a packet. In order to use bandwidth more efficiently and use less energy in end-to-end communication, solutions have been proposed that allow for very small network layer headers instead.

3.1.1.2. Methodology:

- * Header Compression/Translation: One of the main approaches to reduce header size in the IoT context is by compressing it. Such technique is based on a stateful approach, utilizing what is usually called a 'context' on the IoT sensor and the gateway for communications between an IoT device and a server placed somewhere in the Internet - from the edge to the cloud.

The role of the 'context' is to provide a way to 'compress' the original IP header into a smaller one, using shorter address information and/or dropping some field(s); the context here serves as a kind of dictionary.

- * Separate device from locator identifier: Approaches that can offer customized address length that is adequate for use in such constrained domains are preferred. Using different namespaces for the 'device identifier' and the 'routing' or 'locator identifier' is one such approach.

3.1.1.3. Examples

- * Header Compression/Translation: Considering one base station is supposed to serve hundreds of user devices, maximizing the effectiveness for specific spectrum directly improves user quality of experience. To achieve the optimal utilization of the spectrum resource in the wireless area, the RObust Header Compression (ROHC) [RFC5795] mechanism, which has been widely adopted in cellular network like WCDMA, LTE, and 5G, utilizes header compression to shrink existing IPv6 headers onto shorter ones.

Similarly, header compression techniques for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) have been around for several years now, constituting a main example of using the notion of a 'shared context' in order to reduce the size of the network layer header ([RFC6282], [RFC7400], [ITU9959]). More recently, other compression solutions have been proposed for Low Power Wide Area Networks (LPWAN - [RFC8376]). Among them, the Static Context Header Compression (SCHC - [RFC8724]) generalized the compression mechanism developed by 6lo. Instead of a standard compression behavior implemented in all 6lo nodes, SCHC introduces the notion of rule shared by two nodes. The SCHC compression technique is generic and can be applied to IPv6 and above layers. Regarding the nature of the traffic, IPv6 addresses (source and destination) can be elided, partially sent, or replaced by a small index. Instead of the versatile IP packet, SCHC defines new packet formats dedicated to specific applications. SCHC rules are equivalence functions mapping this format to standard IP packets.

Also, constraints coming from either devices or carrier links would lead to mixed scenarios and compound requirements for extraordinary header compression. For native IPv6 communications on DECT ULE and MS/TP Networks [RFC6282], dedicated compression mechanisms are specified in [RFC8105] and [RFC8163], while the transmission of IPv6 packets over NFC and PLC, specifications are being developed in [I-D.ietf-6lo-nfc] and [I-D.ietf-6lo-plc].

- * Separate device from locator identifier: Solutions such as proposed in [EIBP] and [I-D.ietf-lisp-rfc6830bis] can utilize a separation of device from locator, where only the latter is used for routing between the different domains using the same technology, therefore enabling the use of shorter addresses in the (possibly constrained) local environment. Device IDs used within such domains are carried as part of the payload by EIBP and hence can be of shorter size suited to the domain, while, for instance, in LISP a flexible address encoding [RFC8060] allows shorter addresses to be supported in the LISP control plane [I-D.ietf-lisp-rfc6833bis].

3.1.2. Longer Address Length

3.1.2.1. Description

Historically, obtaining adequate address space is considered as the primary and raw motivation to invent IPv6. Longer address (more than 32-bit of IPv4 address), which can accommodate almost inexhaustible devices, used to be considered as the surest direction in 1990s. Nevertheless, to protect the sunk cost of IPv4 deployment, certain efforts focus on IPv4 address space depletion question but engineer IPv4 address length in a more practical way. Such effort, i.e., NAT (Network Address Translation), unexpectedly and significantly slows IPv6 deployment because of its high cost-effectiveness in practice.

Another crucial need for longer address lengths comes from "semantic extensions" to IP addresses, where the extensions themselves do not fit within the length limitation of the IP address. Section 3.3 discusses extensions which extend address semantics that are not limited by the IP address length.

This sub-section focuses on address length extensions that aim at reducing the IPv4 addresses depletion, while Section 3.3, i.e., address semantic extensions, may still refer to extensions when longer address length are suitable to accommodate different address semantic. See Section 3.3 for details of semantic-driven address lengthening.

3.1.2.2. Methodology

- * Split address zone by network realm: This methodology first split the network realm into two types: one public realm (i.e., the Internet), and innumerable private realms (i.e., local networks, which may be embedded and/or having different scope). Then, it splits the IP address space into two type of zones: global address zone (i.e., public address) and local address zone (e.g., private address, reserved address). Based on this, it is assumed that in

public realm, all devices attached to it should be assigned an address that belongs to the global address zone. While for devices attached to private realms, only addresses belonging to the local address zone will be assigned. Local realms may have different scope or even be embedded one in another, like for instance, light switches local network being part of the building local network, which in turn connects to the Internet. In the local realms address may have a pure identification purpose. For instance in last example, addresses of the light switches identify the switches themselves, while the building local network is used to locate them.

Given that the local address zone is not globally unique, certain mechanisms are designed to express the relationship between the global address zone (in public realm) and the local address zone (in any private realm). In this case, global addresses are used for forwarding when a packet is in the public realm, and local addresses are used for forwarding when a packet is in a private realms.

3.1.2.3. Examples

- * Split address zone by network realm: Network Address Translation (NAT), which was first laid out in [RFC2663], using private address and a stateful address binding to translate between the realms. As outlined in [RFC2663], basic address translation is usually extended to include port number information in the translation process, supporting bidirectional or simple outbound traffic only. Because the 16-bits port number is used in the address translation, NAT theoretically increase IPv4 address length from 32-bit to 48-bit, i.e., 281 trillion address space.

Similarly, EzIP [EzIP] expects to utilize a reserved address block, i.e., 240/4, and an IPv4 header option to include it. Based on this, it can be regarded as EzIP is carrying a hierarchical address with two parts, where each part is a partial 32-bit IPv4 address. The first part is a public address residing in the "address field" of the header from globally routable IPv4 pool [IPv4pool], i.e., ca. 3.84 billion address space. The second part is the reserved address residing in "option field" and belongs to the 240/4 prefix, i.e., ca. $2^{28}=268$ million. Based on that, each EzIP deployment is tethered on the existing Internet via one single IPv4 address, and EzIP then have $3.84B * 268M$ address, ca. 1,000,000 trillion. Collectively, the 240/4 can also be used as end point identifier and form an overlay network providing services parallel to the current Internet, yet independent of the latter in other aspects.

Compared to NAT, EzIP is able to establish a communication session from either side of it, hence being completely transparent, and facilitating a full end-to-end networking configuration.

3.1.3. Summary

Table 1 summarizes methodologies and examples towards filling gaps on IP address length extensions.

	Methodology	Examples
Shorter Address Length	Header compression/ translation	6LoWPAN, ROHC, SCHC
	Separate device from locator identifier	EIBP, LISP, ILNP, HIP
Longer Address Length	Split address zone by network realm	NAT, EzIP

Table 1: Summary Length Extensions

3.2. Identity Extensions

Extensions in this subsection attempt extending the property described in Section 2.2, i.e., 'locator/identifier overload' of the ambiguous address semantic.

From the perspective of Internet users, on the one hand, the implicit identifier semantic results in a privacy issue due to network behavior tracking and association. Despite that IP address assignments may be dynamic, they are nowadays considered as 'personal data' and as such undergoes privacy protection regulations like General Data Protection Regulation ("GDPR" [GDPR]). Hence, additional mechanisms are necessary in order to protect end user privacy.

For network regulation of sensitive information, on the other hand, dynamically allocated IP addresses are not sufficient to guarantee device or user identification. As such, different address allocation systems, with stronger identification properties are necessary where security and authentication are at highest priority. Hence, in order to protect information security within a network, additional mechanism are necessary to identify the users or the devices attached to the network.

3.2.1. Anonymous Address Identity

3.2.1.1. Description

As discussed in Section 2.2, IP addresses reveal both 'network locations' as well as implicit 'identifier' information to both traversed network elements and destination nodes alike. This enables recording, correlation, and profiling of user behaviors and historical network traces, possibly down to individual real user identity. The IETF, e.g., in [RFC7258], has taken a clear stand on preventing any such pervasive monitoring means by classifying them as an attack on end users' right to be left alone (i.e., privacy). Regulations such as the EU's General Data Protection Regulation (GDPR) classifies, for instance, the 'online identifier' as personal data which must be carefully protected; this includes end users' IP addresses [GDPR].

Even before pervasive monitoring [RFC7258], IP addresses have been seen as something that some organizational owners of networked system may not want to reveal at the individual level towards any non-member of the same organization. Beyond that, if forwarding is based on semantic extensions, like other fields of the header, extension headers, or any other possible extension, if not adequately protected it may introduce privacy leakage and/or new attack vectors.

3.2.1.2. Methodology:

- * Traffic Proxy: Detouring the traffic to a trusted proxy is a heuristic solution. Since nodes between trusted proxy and destination (including the destination per se) can only observe the source address of the proxy, the 'identification' of the origin source can thereby be hidden. To obfuscate the nodes between origin and the proxy, the traffic on such route would be encrypted via a key negotiated either in-band or off-band. Considering that all applications' traffic in such route can be seen as a unique flow directed to the same 'unknown' node, i.e., the trusted proxy, eavesdroppers in such route have to make more efforts to correlate user behavior through statistical analysis even if they are capable of identifying the users via their source addresses. The protection lays in the inability to isolate single application specific flows. According to the methodology, such approach is IP version independent and works for both IPv4 and IPv6.
- * Source Address Rollover: Privacy issues related to address 'identifier' semantic can be mitigated through regular change (beyond the typical 24 hours lease of DHCP). Due to the semantics of 'identifier' that an IP address carries, such approach promotes

to change the source IP address at a certain frequency. Under such methodology, the refresh cycling window may reach to a balance between privacy protection and address update cost. Due to the limited space that IPv4 contains, such approach usually works for IPv6 only.

- * Private Address Spaces: Their introduction in [RFC1918] foresaw private addresses (assigned to specific address spaces by the IANA) as a means to communicate purely locally, e.g., within an enterprise, by separating private from public IP addresses. Considering that private addresses are never directly reachable from the Internet, hosts adopting private addresses are invisible and thus 'anonymous' for the Internet. Besides, hosts for purely local communication used the latter while hosts requiring public Internet service access would still use public IP addresses.
- * Address Translation: The aforementioned original intention for using private IP addresses, namely for purely local communication, resulted in a lack of flexibility in changing from local to public Internet access on the basis of what application would require which type of service.

If eventually every end-system in an organization would require some form of public Internet access in addition to local one, an adequate number of public Internet addresses would be required for providing to all end systems. Instead, address translation enables to utilize many private IP addresses within an organization, while only relying on one (or few) public IP addresses for the overall organization.

In principle, address translation can be applied recursively. This can be seen in modern broadband access where Internet providers may rely on carrier-grade address translation for all their broadband customers, who in turn employ address translation of their internal home or office addresses to those (private again) IP addresses assigned to them by their network provider.

Two benefits arise from the use of (private to public IP) address translation, namely (i) the hiding of local end systems at the level of the (address) assigned organization, and (ii) the reduction of public IP addresses necessary for communication across the Internet. While the latter has been seen for long as a driver for address translation, we focus on the first issue in this section, also since we see such privacy benefit as well as objective as still being valid in addressing systems like IPv6 where address scarcity is all but gone [GNATCATCHER].

- * Separate device from locator identifier: Solutions that make a clear separation between the routing locator and the identifier, can allow for a device ID of any size, which in turn can be encrypted by a network element deployed at the border of routing domain (e.g., access/edge router). Both source and end-domain addresses can be encrypted and transported, as in the routing domain, only the routing locator is used.

3.2.1.3. Examples:

- * Traffic Proxy: Although not initially designed as a traffic proxy approach, a Virtual Private Network (VPN [VPN]) is widely utilized for packets origin hiding as a traffic detouring methodology. As it evolved, VPN derivatives like WireGuard [WireGuard] have become a mainstream instance for user privacy and security enhancement.

With such methodology in mind, onion routing [ONION], instantiated in the TOR Project [TOR], achieves high anonymity through traffic hand over via intermediates, before reaching the destination. Since the architecture of TOR requires at least three proxies, none of them is aware of the entire route. Given that the proxies themselves can be deployed all over cyberspace, trust is not the prerequisite if proxies are randomly selected.

In addition, dedicated protocols are also expected to be customized for privacy improvement via traffic proxy. For example, Oblivious DNS over HTTPS (ODOH [ODOH]) use a third-party proxy to obscure identifications of user source addresses during DNS over HTTPS (DoH [RFC8484]) resolution. Similarly, Oblivious HTTP [OHTTP] involve proxy alike in the HTTP environment.

- * Source Address Rollover: As for source address rollover, it has been standardized that IP addresses for Internet users should be dynamic and temporary every time they are being generated [RFC8981]. This benefits from the available address space in the case of IPv6, through which address generation or assignment should be unpredictable and stochastic for outside observers.

More radically, [EPHEMERALv6] advocates an 'ephemeral address', changing over time, for each process. Through this, correlating user behaviors conducted by different identifiers (i.e., source address) becomes much harder, if not impossible, if based on the IP packet header alone.

- * Private Addresses: The use and assignment of private addresses for IPv4 is laid out in [RFC1918], while unique local addresses (ULAs) in IPv6 [RFC4193] take over the role of private address spaces in IPv4.

- * Network Address Translation: Given address translation can be performed several times in cascade, NATs may exist as part of existing customer premise equipment (CPE), such as a cable or an Ethernet router, with private wired/wireless connectivity, or may be provided in a carrier environment to further translate ISP-internal private addresses to a pool of (assigned) public IP addresses. The latter is often dynamically assigned to CPEs during its bootstrapping.
- * Separate device from locator identifier: EIBP [EIBP] utilizes a structured approach to addressing. It separates the routing ID from the device ID, where only the former is used for routing. As such, the device IDs can be encrypted, protecting the end device identity. Similarly, LISP uses separate namespaces for routing and identification allowing to 'hide' identifiers in encrypted LISP packets that expose only known routing information [RFC8061].

3.2.2. Authenticated Address Identity

3.2.2.1. Description

In some scenarios (e.g., corporate networks) it is desirable to being able authenticate IP addresses in order to prevent malicious attackers spoofing IP addresses. This is usually achieved by using a mechanism that allows to prove ownership of the IP address.

3.2.2.2. Methodology

- * Self-certified addresses: This method is usually based on the use of nodes' public/private keys. A node creates its own interface ID (IID) by using a cryptographic hash of its public key (with some additional parameters). Messages are then signed using the nodes' private key. The destination of the message will verify the signature through the information in the IP address. Self-certification has the advantage that no third party or additional security infrastructure is needed. Any node can generate its own address locally and then only the address and the public key are needed to verify the binding between the public key and the address.

- * Third party granted addresses: DHCP (Dynamic Host Configuration Protocol) is widely used to provide IP addresses, however, in its basic form, it does not perform any check and even an unauthorized user without the right to use the network can obtain an IP address. To solve this problem, a trusted third party has to grant access to the network before generating an address (via DHCP or other) that identifies the user. User authentication done securely either based on physical parameters like MAC addresses or based on an explicit login/password mechanism.

3.2.2.3. Examples

- * Self-certified Addresses: As an example of this methodology serves [RFC3972], defining IPv6 cryptographically Generated Addresses (CGA). A Cryptographically Generated Address is formed by replacing the least-significant 64 bits of an IPv6 address with the cryptographic hash of the public key of the address owner. Packets are then signed with the private key of the sender. Packets can be authenticate by the receiver by using the public key of the sender and the address of the sender. The original specifications have been already amended (cf., [RFC4581] and [RFC4982]) in order to support multiple (stronger) cryptographic algorithms.
- * Third party granted addresses: [RFC3118] defines a DHCP option through which authorization tickets can be generated and newly attached hosts with proper authorization can be automatically configured from an authenticated DHCP server. Solutions exist where separate servers are used for user authentication like [UA-DHCP] and [RFC4014]. The former proposing to enhance the DHCP system using registered user login and password before actually providing an IP address lease and recording the MAC address of the device the user used to sign-in. The latter, couples the RADIUS authentication protocol ([RFC2865]) with DHCP, basically piggybacking RADIUS attributes in a DHCP sub-option, with the DHCP server contacting the RADIUS server to authenticate the user.

3.2.3. Summary

Table 2, summarize the methodologies and the examples towards filling the gaps on identity extensions.

	Methodology	Examples
Anonymous Address Identity	Traffic Proxy	VPN, TOR, ODoH
	Source Address Rollover	SLAAC
	Private Address Spaces	ULA
	Address Translation	NAT
	Separate device from locator identifier	EIBP, LISP
Authenticated Address Identity	Self-certified Addresses	CGA
	Third party granted addresses	DHCP-Option

Table 2: Summary Identity Extensions

3.3. Semantic Extensions

Extensions in this subsection try extending the property described in Section 2.3, i.e., limited address semantic support.

As explained in Section 2.2, IP addresses carry both locator and identification semantic. Some efforts exist that try to separate these semantics either in different address spaces or through different address formats. Beyond just identification, location, and the fixed address size, other efforts extended the semantic through existing or additional header fields (or header options) outside the Internet address.

How much unique and globally routable an address should be? With the effect of centralization, edges communicate with (rather) local DCs, hence a unique address globally routable is not a requirement anymore. There is no need to use globally unique addresses all the time for communication, however, there is the need of having a unique address as a general way to communicate to any connected entity without caring what transmission networks the packets traverse.

3.3.1. Utilizing Extended Address Semantics

3.3.1.1. Description

Several extensions have been developed to extend beyond the limited IPv6 semantics. Those approaches may include to apply structure to the address, utilize specific prefixes, or entirely utilize the IPv6 address for different semantics, while re-encapsulating the original packet to restore the semantics in another part of the network. For instance, structured addresses have the capability to introduce delimiters to identify semantic information in the header, therefore not constraining any semantic by size limitations of the address fields.

We note here that extensions often start out as being proposed as an extended header semantic, while standardization may drive the solution to adopt an approach to accommodate their semantic within the limitations of an IP address. This section does include examples of this kind.

3.3.1.2. Methodology

*Semantic prefixes: Semantic prefixes are used to separate the IPv6 address space. Through this, new address families, such as for information-centric networking [HICN], service routing or other semantically rich addressing, can be defined, albeit limited by the prefix length and structure as well as the overall length limitation of the IPv6 address.

* Separate device/resource from locator identifier: The option to use separate namespaces for the device address would offer more freedom for the use of different semantics. For instance, the static binding of IP addresses to servers creates a strong binding between IP addresses and service/resources, which may be a limitation for large Content Distribution networks (CDNs) [FAYED21].

As an extreme form of separating resource from locator identifier, recent engineering approaches, described in [CLOUDFLARE_SIGCOMM], decouple web service (semantics) from the routing address assignments by using virtual hosting capabilities, thereby effectively mapping possibly millions of services onto a single IP address.

* Structured addressing: One approach to address the routing challenges faced in the Internet is the use of structured addresses, e.g., to void the need for routing protocols. Benefits of this approach can be significant, with the structured addresses

capturing the relative physical or virtual position of routers in the network as well as being variable in length. Key to the approach, however, is that the structured addresses capturing the relative physical or virtual position of routers in the network, or networks in an internetwork may not fit within the fixed and limited IP address length (cf., Section 3.1.2). Other structured approaches may be the use of application-specific structured binary components for identification, generalizing URL schema used for HTTP-level communication but utilized at the network level for traffic steering decisions.

- * Localized forwarding semantics: Layer 2 hardware, such as SDN switches, are limited to the use of specific header fields for forwarding decisions. Hence, devising new localized forwarding mechanisms may be based on re-using differently existing header fields, such as the IPv6 source/destination fields, to achieve the desired forwarding behavior, while encapsulating the original packets in order to be restored at the local forwarding network boundary. Networks in those solutions are limited by the size of the utilized address field, e.g., 256 bits for IPv6, thereby limiting the way such techniques could be used.

3.3.1.3. Examples

- * Semantic prefixes: Newer approaches to IP anycast suggest the use of service identification in combination with a binding IP address model [SFCANYCAST] as a way to allow for metric-based traffic steering decisions; approaches for Service Function Chaining (SFC) [RFC7665] utilize the Network Service Header (NSH) information and packet classification to determine the destination of the next service.

Another example of the usage of different packet header extensions based on IP addressing is Segment Routing. In this case, the source chooses a path and encodes it in the packet header as an ordered list of segments. Segments are encoded using new Routing Extensions Header type, the Segment Routing Header (SRH), which contains the Segment List, similar to what is already specified in [RFC8200], i.e., a list of segment ID (SID) that dictate the path to follow in the network. Such segment IDs are coded as 128 bit IPv6 addresses [RFC8986].

Approaches such as [HICN] utilize semantic prefixing to allow for ICN forwarding behavior within an IPv6 network. In this case, an HICN name is the hierarchical concatenation of a name prefix and a name suffix, in which the name prefix is encoded as an IPv6 128 bits word and carried in IPv6 header fields, while the name suffix is encoded in transport headers fields such as TCP. However, it

is a challenge to determine which IPv6 prefixes should be used as name prefixes. In order to know which IPv6 packets should be interpreted based on an ICN semantic, it is desirable to be able to recognize that an IPv6 prefix is a name prefix, e.g. to define a specific address family (AF_HICN, b0001::/16). This establishment of a specific address family allows the management and control plane to locally configure HICN prefixes and announce them to neighbors for interconnection.

- * Separate device from locator identifier: EIBP [EIBP] separates the routing locator from the device identifier, relaxing therefore any semantic constraints on the device identifier. Similarly, LISP uses a flexible encoding named LISP Canonical Address Format (LCAF [RFC8061]), which allows to associate to routing locators any possible form (and length) of identifier. ILNP [RFC6740] introduces as well a different semantic of IP addresses, while aligning to the IPv6 address format (128 bits). Basically, ILNP introduces a sharper logical separation between the 64 most significant bits and the 64 least significant bits of an IPv6 address. The former being a global locator, while the latter being an identifier that can have different semantics (rather than just being an interface identifier).
- * Structured addressing: Network topology captures the physical connectivity among devices in the network. There is a structure associated with the topology. Examples are the core-distribution-access router structure commonly used in enterprise networks and clos topologies that are used to provide multiple connections between Top of Rack (ToR) devices and multiple layers of spine devices. Internet service providers use a tier structure that defines their business relationships. A clear structure of connected networks can be noticed in the Internet. EIBP [EIBP] proposes to leverage the physical structure (or a virtual structure overlaid on the physical structure) to auto assign addresses to routers in a network or networks in an internetwork to capture their relative position in the physical/virtual topology. EIBP proposes to administratively identify routers/networks with a tier value based on the structure.
- * Localized forwarding semantics: Approaches such as those outlined in [REED] suggest using a novel forwarding semantic based on path information carried in the packet itself, said path information consists in a fixed size bit-field (see [REED] for more information on how to represent the path information in said bit-field). In order to utilize existing, e.g., SDN-based, forwarding switches, the direct use of the IPv6 source/destination address is suggested for building appropriate match-action rules (over the suitable binary information representing the local output ports),

while preserving the original IPv6 information in the encapsulated packet. As mentioned above, such use of the existing IPv6 address fields limits the size of the network to a maximum of 256 bits (therefore paths in the network over which such packets can be forwarded). [ICNIP], however, goes a step further by suggesting to use the local forwarding as direct network layer mechanism, removing the IP packet and only leaving the transport/application layer, with the path identifier constituting the network-level identifier albeit limited by using the existing IP header for backward compatibility reasons (the next section outlines the removal of this limitation).

3.3.2. Utilizing Existing or Extended Header Semantics

3.3.2.1. Description:

While the former sub-section explored extended address semantic, thereby limiting any such extended semantic with that of the existing IPv6 semantic and length, additional semantics may also be placed into the header of the packet or the packet itself, utilized for the forwarding decision to the appropriate endpoint according to the extended semantic.

Reasons for embedding such new semantics may be related to traffic engineering since it has long been shown that the IP address itself is not enough to steer traffic properly since the IP address itself is not semantically rich enough to adequately describe the forwarding decision to be taken in the network, not only impacting WHERE the packet will need to go but also HOW it will need to be sent.

3.3.2.2. Methodology:

- * In-Header extensions: One way to add additional semantics besides the address fields is to use other fields already present in the header.
- * Headers option extensions: Another mechanism to add additional semantics is to actually add additional fields, e.g., through Header Options in IPv4 or through Extension Headers in IPv6.
- * Re-encapsulation extension: A more radical approach for additional semantics is the use of a completely new header that is designed so to carry the desired semantics in an efficient manner (often as a shim header).
- * Structured addressing: Similar to the methodology that structures addresses within the limitations of the IPv6 address length, outlined in the previous sub-sections, structured addressing can

also be applied within existing or extended header semantics, e.g., utilizing a dedicated (extension) header to carry the structured address information.

- * Localized forwarding semantics: This set of solutions applies capabilities of newer (programmable) forwarding technology, such as [P4], to utilize any header information for a localized forwarding decision. This removes any limitation to use existing header or address information for embedding a new address semantic into the transferred packet.

3.3.2.3. Examples:

- * In-Header extensions: In order to allow additional semantic with respect to the pure Internet addressing, the original design of IPv4 included the field 'Type of Service' [RFC2474], while IPv6 introduced the 'Flow label' and the 'Traffic Class' [RFC8200]. In a certain way, those fields can be considered 'semantic extensions' of IP addresses, and they are 'in-header' because natively present in the IP header (differently from options and extension headers). However, they proved not to be sufficient. Very often a variety of network operation are performed on the well-known 5-tuple (source and destination addresses; source and destination port number; and protocol number). In some contexts all of the above mentioned fields are used in order to have a very fine grained solution ([RFC8939]).
- * Headers option extensions: Header options have been largely under-exploited in IPv4. However, the introduction of the more efficient extension header model in IPv6 along with technology progress made the use of header extensions more widespread in IPv6. Segment Routing re-introduced the possibility to add path semantic to the packet by encoding a loosely defined source routing ([RFC8402]). Similarly, in the aim to overcome the inherent shortcoming of the multi-homing in the IP context, SHIM6 ([RFC5533]) also proposed the use of an extension header able to carry multi-homing information which cannot be accommodated natively in the IPv6 header.

To serve a moving endpoint, mechanisms like Mobile IPv6 [RFC6275] are used for maintaining connection continuity by a dedicated IPv6 extension header. In such case, the IP address of the home agent in Mobile IPv6 is basically an identification of the on-going communication. In order to go beyond the interface identification model of IP, the Host Identity Protocol (HIP) tries to introduce an identification layer to provide (as the name says) host identification. The architecture here relies on the use of another type of extension header [RFC7401].

- * Re-encapsulation extension: Differently from the previous approach, re-encapsulation prepends complete new IP headers to the original packet introducing a completely custom shim header between the outer and inner header. This is the case for LISP, adding a LISP specific header right after an IP+UDP header ([I-D.ietf-lisp-rfc6830bis]). A similar design is used by VxLAN ([RFC7348]) and GENEVE ([RFC8926]), even if they are designed for a data center context. IP packets can also be wrapped with headers using more generic and semantically rich names, for instance with ICN [ICNIP].
- * Structured addressing: Solutions such as those described in the previous sub-section, e.g., EIBP [EIBP], can provide structured addresses that are not limited to the IPv6 address length but instead carry the information in an extension header to remove such limitation.

Also Information-Centric Networking (ICN) naming approaches usually introduce structures in the (information) names without limiting themselves to the IP address length; more so, ICN proposes its own header format and therefore radically breaks with not only IP addressing semantic but the format of the packet header overall. For this, approaches such as those described in [RFC8609] define a TLV-based binary application component structure that is carried as a 'name' part of the CCN messages. Such a name is a hierarchical structure for identifying and locating a data object, which contains a sequence of name components. Names are coded based on 2-level nested Type-Length-Value (TLV) encodings, where the name-type field in the outer TLV indicates this is a name, while the inner TLVs are name components including a generic name component, an implicit SHA-256 digest component and a SHA-256 digest of Interest parameters. For textual representation, URIs are normally used to represent names, as defined in [RFC3986].

In geographic addressing, position based routing protocols use the geographic location of nodes as their addresses, and packets are forwarded when possible in a greedy manner towards the destination. For this purpose, the packet header includes a field coding the geographic coordinates (x, y, z) of the destination node, as defined in [RFC2009]. Some proposals also rely on extra fields in the packet header to code the distance towards the destination, in which case only the geographic coordinates of neighbors are exchanged. This way the location of the destination is protected even if routing packets are eavesdropped.

- * Localized forwarding semantics: Unlike the original suggestion in [REED] to use existing SDN switches, the proliferation of P4 [P4] opens up the possibility to utilize a locally limited address semantic, e.g., expressed through the path identifier, as an entirely new header (including its new address) with an encapsulation of the IP packet for E2E delivery (including further delivery outside the localized forwarding network or positioning the limited address semantic directly as the network address semantic for the packet, i.e., removing any IP packet encapsulation from the forwarded packet, as done in [ICNIP]). Removing the IPv6 address size limitation by not utilizing the existing IP header for the forwarding decision also allows for extensible length approaches for building the path identifier with the potential for increasing the supported network size. On the downside, this approach requires to encapsulate the original IP packet header for communication beyond the local domain in which the new header is being used, such as discussed in the previous point above on 're-encapsulation extension'.

3.3.3. Summary

Table 3, summarize the methodologies and the examples towards filling the gaps on semantic extensions.

	Methodology	Examples
Utilizing Extended Address Semantics	Semantic prefixes	HICN
	Separate device from locator identifier	EIBP, ILNP, LISP, HIP
	Structured addressing	EIBP, ILNP
	Localized forwarding semantics	REED
Utilizing Existing or Extended Header Semantics	In-Header extensions	DetNet
	Headers option extensions	SHIM6, SRv6, HIP
	Re-encapsulation extension	VxLAN, ICNIP
	Structured addressing	EIBP
	Localized forwarding semantics	REED

Table 3: Summary Semantic Extensions

4. Overview of Approaches to Extend Internet Addressing

The following Table 4 describes the objectives of the extensions discussed in this memo with respect to the properties of Internet addressing (Section 2). As summarized, extensions may aim to extend one property of the Internet addressing, or extend other properties at the same time.

	Length Extension	Identity Extension	Semantic Extension
6LoWPAN	x		
ROHC	x		

EzIP	x		
TOR		x	
ODoH		x	
SLAAC		x	
CGA		x	x
NAT	x	x	
HICN		x	x
ICNIP	x	x	x
CCNx names	x	x	x
EIBP	x	x	x
Geo addressing	x		x
REED	x (with P4)		x
DetNet		x	
Mobile IP			x
SHIM6			x
SRv6			x
HIP		x	x
VxLAN		x	x
LISP		x	x
SFC		x	x

Table 4: Relationship between Extensions and Internet Addressing

5. A System View on Address

In the following, we investigate in which parts of the overall Internet system extensions have been proposed and developed. For this, we divide the possible innovation across two dimensions:

- * Horizontal: Internet edge vs core. The criticality, scale, investment on the core of the Internet makes it more difficult to introduce innovation, while at the edges there is more flexibility. As general purpose processors have drastically improved in performance, data-plane features can be implemented in software. At the edge of the Internet, it is easier to introduce innovation for several reasons: Economics, faster ROI because of faster deployment; No need of large scale deployment (and hence less standardization effort); less stakeholders involved (sometimes just one, see following point). Furthermore, the fact that the edge is a place where there is less coordination and cooperation from the core, is another factor that eases the innovation.
- * Vertical: at which layer of the protocol stack. The difficulty to innovate varies as well depending at which layer the innovation takes place. One thing is to innovate at application layer where the app developer has large degree of freedom, another is to innovate at network layer, which is more constrained because of its central point in the architecture. Innovation at higher layer sometimes leads to walled gardens (aka limited domains [RFC8799]). Indeed because of the centralization phenomena, an actor offering a certain service may very well develop and deploy a custom technology that does not need to be actually standardized because it is done for its own internal usage.
- * Horizontal vs Vertical Innovation:
 - In the public Internet, core innovation at lower layer is harder, often reduced to app-level innovation or building an overlay limited domain (aka a walled garden).
 - At the edges it is easier to innovate at lower layers (more vertical flexibility) but some form of adaptation is needed if global reachability is wanted.

Despite these two orthogonal dimensions, innovation does not happen either horizontally or vertically, rather in both dimensions simultaneously at various degree.

6. Issues in Extensions to Internet Addressing

While the extensions to the original Internet properties, discussed in Section 3, demonstrate the benefits of more flexibility in addressing, they also bring with them a number of issues, which are discussed in the following section. To this end, the problems hereafter outlined link to the approaches to extensions summarized in Section 4. These issues may not be present all the time and everywhere, since as explained in Section 5, extensions are developed and deployed in different part of the Internet, which may worsen things.

6.1. Limiting Address Semantics

Many approaches changing the semantics of communication, e.g., through separating host identification from network node identification [RFC7401], separating the device identifier from the routing locator ([EIBP], [I-D.ietf-lisp-introduction]), or through identifying content and services directly [HICN], are limited by the existing packet size and semantic constraints of IPv6, e.g., in the form of its source and destination network addresses.

While approaches such as [ICNIP] may override the addressing semantics, e.g., by replacing IPv6 source and destination information with path identification, a possible unawareness of endpoints still requires the carrying of other address information as part of the payload.

Also, the expressible service or content semantic may be limited, as in [HICN] or the size of supported networks [REED] due to relying on the limited bit positions usable in IPv6 addresses.

6.2. Complexity and Efficiency

A crucial issue is the additional complexity introduced for realizing the additional addressing semantics. This is particularly an issue since we see those additional semantics particularly at the edge of the Internet, utilizing the existing addressing semantic of the Internet to interconnect the domains that require those additional semantics.

Furthermore, any additional complexity often comes with an efficiency and cost penalty, particularly at the edge of the network, where resource constraints may play a significant role. Compression processes, taking [ROHC] as an example, require additional resources both for the sender generating the compressed header but also the gateway linking to the general Internet by re-establishing the full IP header.

Conversely, the performance requirements of core networks, in terms of packet processing speed, makes the accommodation of extensions to addressing often prohibitive. This is not only due to the necessary extra processing that is specific to the extension, but also due to the complexity that will need to be managed in doing so at significantly higher speeds than at the edge of the network. The observations on the dropping of packets with IPv6 extension headers in the real world is (partially) due to such a implementation complexity [RFC7872].

Another example for lowering the efficiency of packet forwarding is the routing in systems like TOR [TOR]. As detailed before, traffic in TOR, for anonymity purposes, should be handed over by at least three intermediates before reaching the destination. Frequent relaying enhances the privacy, however, because such kind of solutions are implemented at application level, they come at the cost of lower communication efficiency. May be a different privacy enhanced address semantic would enable efficient implementation of TOR-like solutions at network layer.

6.2.1. Repetitive encapsulation

Repetitive encapsulation is an issue since it bloats the packets size due to additional encapsulation headers. Addressing proposals such as those in [ICNIP] utilize path identification within an alternative forwarding architecture that acts upon the provided path identification. However, due to the limitation of existing flow-based architectures with respect to the supported header structures (in the form of IPv4 or IPv6 headers), the new routing semantics are being inserted into the existing header structure, while repeating the original, sender-generated header structure, in the payload of the packet as it traverses the local domain, effectively doubling the per-packet header overhead.

The problem is also present in a number of solutions tackling different issues, e.g., mobility [I-D.ietf-lisp-mn], DC networking ([RFC8926], [RFC7348], [I-D.ietf-intarea-gue]), traffic engineering [RFC8986], and privacy ([TOR], [SPHINX]). Certainly these solutions are able to avoid other issues, like path lengthening or privacy issues, as described before, but they come at the price of multiple encapsulations that reduce the effective payload. This, not only hampers efficiency in terms of header-to-payload ratio, but also introduces 'encapsulation points', which in turn add complexity to the (often edge) network as well as fragility due to the addition of possible failure points; this aspect is discussed in further details in Section 6.4.

6.2.2. Compounding issues with header compression

IP header overhead requires header compression in constrained environments, such as wireless sensor networks and IoT in general. Together with fragmentation, both tasks constitute significant energy consumption, as shown in [HEADER_COMP_ISSUES1], negatively impacting resource limited devices that often rely on battery for operation. Further, the reliance on the compression/decompression points creates a dependence on such gateways, which may be a problem for intermittent scenarios.

According to the implementation of `_contiki-ng_` [CONTIKI], an example of operating system for IoT devices, the source codes for 6LowPan requires at least 600Kb to include a header compression process. In certain use cases, such requirement can be an obstacle for extremely constrained devices, especially for the RAM and energy consumption.

6.2.3. Introducing Path Stretch

Mobile IP [RFC6275], which was designed for connection continuity in the face of moving endpoints, is a typical case for path stretch. Since traffic must follow a triangular route before arriving at the destination, such detour routing inevitably impacts transmission efficiency as well as latency.

6.2.4. Complicating Traffic Engineering

While many extensions to the original IP address semantic target to enrich the decisions that can be taken to steer traffic, according to requirements like QoS, mobility, chaining, compute/network metrics, flow treatment, path usage, etc., the realization of the mechanisms as individual solutions likely complicates the original goal of traffic engineering when individual solutions are being used in combination. Ultimately, this may even prevent the combined use of more than one mechanism and/or policy with a need to identify and prevent incompatibilities of mechanisms. Key here is not the issue arising from using conflicting traffic engineering policies, rather conflicting realizations of policies that may well generally work well alongside ([ROBUSTSDN], [TRANSACTIONSDN]).

This not only increases fragility, as discussed separately in Section 6.4, but also requires careful planning of which mechanisms to use and in which combination, likely needing human-in-the-loop approaches alongside possible automation approaches for the individual solutions.

6.3. Security

The properties described in Section 2 have, obviously, also consequences in terms of security and privacy related issues, as already mentioned in other parts of this document.

For instance, in the effort of being somehow backward compatible, HIP [RFC7401] uses a 128-bit Host Identity, which may be not sufficiently cryptographically strong in the future because of the limited size (future computational power may erode 128-bit security). Similarly, CGA [RFC3972] also aligns to the 128-bit limit, but may use only 59 bits of them, hence, the packet signature may not be sufficiently robust to attacks [I-D.rafiiee-6man-cga-attack].

IP addresses, even temporary ones meant to protect privacy, have been long recognized as a 'Personal Identification Information' that allows even to geolocate the communicating endpoints [RFC8280]. The use of temporary addresses provides sufficient privacy protection only if the renewal rate is high [EPHEMERALv6]. However, this causes additional issues, like the large overhead due to the Duplicate Address Detection, the impact on the Neighbor Discovery mechanism, in particular the cache, which can even lead to communication disruption. With such drawbacks, the extensions may even lead to defeat the target, actually lowering security rather than increasing it.

The introduction of alternative addressing semantics has also been used to help in (D)DoS attacks mitigation. This leverages on changing the service identification model so to avoid topological information exposure, making the potential disruptions likely remain limited [ADDRLESS]. However, this increased robustness to DDoS comes at the price of important communication setup latency and fragility, as discussed next.

6.4. Fragility

From the extensions discussed in Section 3, it is evident that having alternative or additional address semantic and formats available for making routing as well as forwarding decisions dependent on these, is common place in the Internet. This, however, adds many extension-specific translation/adaptation points, mapping the semantic and format in one context into what is meaningful in another context, but also, more importantly, creating a dependency towards an additional component, often without explicit exposure to the endpoints that originally intended to communicate.

For instance, the re-writing of IP addresses to facilitate the use of private address spaces throughout the public Internet, realized through network address translators (NATs), conflicts with the end-to-end nature of communication between two endpoints. Additional (flow) state is required at the NAT middle-box to smoothly allow communication, which in turn creates a dependency between the NAT and the end-to-end communication between those endpoints, thus increasing the fragility of the communication relation.

A similar situation arises when supporting constrained environments through a header compression mechanism, adding the need for, e.g., a ROHC [RFC5795] element in the communication path, with communication-related compression state being held outside the communicating endpoints. Failure will introduce some inefficiencies due to context regeneration, which may affect the communicating endpoints, increasing fragility of the system overall.

Such translation/adaptation between semantic extensions to the original 'semantic' of an IP address is generally not avoidable when accommodating more than a single universal semantic. However, the solution-specific nature of every single extension is likely to noticeably increase the fragility of the overall system, since individual extensions will need to interact with other extensions that may be deployed in parallel, but were not designed taking into account such deployment scenario (cf., [I-D.ietf-intarea-tunnels]). Considering that extensions to traditional per-hop-behavior (based on IP addresses) can essentially be realized over almost 'any' packet field, the possible number of conflicting behaviors or diverging interpretation of the semantic and/or content of such fields, among different extensions, may soon become an issue, requiring careful testing and delineation at the boundaries of the network within which the specific extension has been realized.

7. Summary of issues

Table 5, derived from Section 6, summarizes the issues related to each extension. While each extension involves at least one issue, some others, like ICNIP, may create several issues at the same time.

	Limiting Address Semantics	Complexity and Efficiency	Security	Fragility
6LoWPAN		x		x
ROHC		x		x

EzIP		x		
TOR		x		x
ODoH		x		
SLAAC		x		
CGA	x		x	
NAT		x		x
HICN	x			
ICNIP	x	x		
CCNx name	x			
EIBP				x
Geo addressing	x			x
REED	x			
DetNet		x		
Mobile IP		x		x
SHIM6				x
SRv6				x
HIP			x	x
VxLAN		x		
LISP		x		x
SFC		x		x

Table 5: Issues in Extensions to Internet Addressing

8. Conclusions

The examples of extensions discussed in Section 3 to the original Internet addressing scheme show that extensibility beyond the original model (and its underlying per-hop behavior) is a desired capability for networking technologies and has been so for a long time. Generally, we can observe that those extensions are driven by the requirements of stakeholders, expecting a desirable extended functionality from the introduction of the specific extension. If interoperability is required, those extensions require standardization of possibly new fields, new semantics as well as (network and/or end system) operations alike.

The issues we identified in this document with the extension-specific solution approach, point to the need for a discussion on Internet addressing, as formulated in the companion document [I-D.jia-intarea-scenarios-problems-addressing] that formalizes the problem statement through scenarios that highlight the shortcomings of the Internet addressing model.

It is our conclusion that the existence of the many extensions to the original Internet addressing is clear evidence for gaps that have been identified over time by the wider Internet community, each of which come with a raft of issues that we need to deal with daily: We believe that it is time to develop an architectural but more importantly a sustainable approach to make Internet addressing extensible in order to capture the many new use cases that will still be identified for the Internet to come.

To jumpstart any such effort from an addressing perspective, it will be key to suitably define what an address is at which layer of the overall system, let alone the network layer. We argue that any answer to this question must be derived from what features we may want from the network instead of being guided by the answers that the Internet can give us today, e.g., being a mere ephemeral token for accessing PoP-based services (as indicated in related arch-d mailing list discussions).

This is not to 'second guess' the market and its possible evolution, but to outline clear features from which to derive clear principles for a design. Any such design must not skew the technical capabilities of addressing to the current economic situation of the Internet since this bears the danger of locking down innovation capabilities as an outcome of those technical limitations introduced. Instead, addressing must be aligned with enabling the model of permissionless innovation that the IETF has been promoting, ultimately enabling the serendipity of new applications that has led to many of those applications we can see in the Internet today. Most

importantly, any inaction on our side in that regard will only compound the issues identified, eventually hampering the future Internet's readiness for those new uses.

9. Security Considerations

The present memo does not introduce any new technology and/or mechanism and as such does not introduce any security threat to the TCP/IP protocol suite.

As an additional note, and as discussed in this document, security and privacy aspects were not considered as part of the key properties for Internet addressing, which led to the introduction of a number of extensions intending to fix those gaps. The analysis presented in this memo (non-exhaustively) shows those issues are either solved in an ad-hoc manner at application level, or at transport layer, while at network level only few extensions tackling specific aspects exist, albeit often with limitations due to the adherence to the Internet addressing model and its properties.

10. IANA Considerations

This document does not include any IANA request.

11. Informative References

[ADDRLESS] Hao, S., Liu, R., Weng, Z., Chang, D., Bao, C., and X. Li, "Addressless: A new internet server model to prevent network scanning", PLOS ONE Vol. 16, pp. e0246293, DOI 10.1371/journal.pone.0246293, February 2021, <<https://doi.org/10.1371/journal.pone.0246293>>.

[CLOUDFLARE_SIGCOMM] Fayed, M., Bauer, L., Giotsas, V., Kerola, S., Majkowski, M., Odintsov, P., Sitnicki, J., Chung, T., Levin, D., Mislove, A., Wood, C., and N. Sullivan, "The ties that unbind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale", Proceedings of the 2021 ACM SIGCOMM 2021 Conference, DOI 10.1145/3452296.3472922, August 2021, <<https://doi.org/10.1145/3452296.3472922>>.

[CONTIKI] "Contiki-NG: The OS for Next Generation IoT Devices", n.d., <<https://github.com/contiki-ng/contiki-ng>>.

- [EIBP] Shenoy, S Chandraiah, P Willis, N., "A Structured Approach to Routing in the Internet", June 2021, <First Intl Workshop on Semantic Addressing and Routing for Future Networks>.
- [EPHEMERALv6] Gont, F. and G. Gont, "IPv6 Addressing Considerations", Work in Progress, Internet-Draft, draft-gont-v6ops-ipv6-addressing-considerations-01, 21 February 2021, <<https://www.ietf.org/archive/id/draft-gont-v6ops-ipv6-addressing-considerations-01.txt>>.
- [EzIP] Chen, A. Y., Ati, R. R., Karandikar, A., and D. R. Crowe, "Adaptive IPv4 Address Space", Work in Progress, Internet-Draft, draft-chen-ati-adaptive-ipv4-address-space-10, 8 December 2021, <<https://www.ietf.org/archive/id/draft-chen-ati-adaptive-ipv4-address-space-10.txt>>.
- [FAYED21] Fayed, M., Bauer, L., Giotsas, V., Kerola, S., Majkowski, M., Odintsov, P., Sitnicki, J., Chung, T., Levin, D., Mislove, A., Wood, C., and N. Sullivan, "The ties that unbind: decoupling IP from web services and sockets for robust addressing agility at CDN-scale", Proceedings of the 2021 ACM SIGCOMM 2021 Conference, DOI 10.1145/3452296.3472922, August 2021, <<https://doi.org/10.1145/3452296.3472922>>.
- [GDPR] Voigt, P. and A. von dem Bussche, "The EU General Data Protection Regulation (GDPR)", Springer International Publishing book, DOI 10.1007/978-3-319-57959-7, 2017, <<https://doi.org/10.1007/978-3-319-57959-7>>.
- [GNATCATCHER] "Global Network Address Translation Combined with Audited and Trusted CDN or HTTP-Proxy Eliminating Reidentification", n.d., <<https://github.com/bslassey/ip-blindness>>.
- [HEADER_COMP_ISSUES1] Mesrinejad, F., Hashim, F., Noordin, N., Rasid, M., and R. Abdullah, "The effect of fragmentation and header compression on IP-based sensor networks (6LoWPAN)", The 17th Asia Pacific Conference on Communications, DOI 10.1109/apcc.2011.6152926, October 2011, <<https://doi.org/10.1109/apcc.2011.6152926>>.

- [HICN] Muscariello, L., "Hybrid Information-Centric Networking: ICN inside the Internet Protocol", March 2018, <<https://datatracker.ietf.org/meeting/interim-2018-icnrg-01/materials/slides-interim-2018-icnrg-01-sessa-hybrid-icn-hicn-luca-muscariello>>.
- [HISTORY127] "History of 127/8 as localhost/loopback addresses", n.d., <<https://elists.isoc.org/pipermail/internet-history/2021-January/006920.html>>.
- [I-D.ietf-6lo-nfc] Choi, Y., Hong, Y., Youn, J., Kim, D., and J. Choi, "Transmission of IPv6 Packets over Near Field Communication", Work in Progress, Internet-Draft, draft-ietf-6lo-nfc-17, 23 August 2020, <<https://www.ietf.org/archive/id/draft-ietf-6lo-nfc-17.txt>>.
- [I-D.ietf-6lo-plc] Hou, J., Liu, B., Hong, Y., Tang, X., and C. E. Perkins, "Transmission of IPv6 Packets over PLC Networks", Work in Progress, Internet-Draft, draft-ietf-6lo-plc-10, 17 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-6lo-plc-10.txt>>.
- [I-D.ietf-intarea-gue] Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", Work in Progress, Internet-Draft, draft-ietf-intarea-gue-09, 26 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-gue-09.txt>>.
- [I-D.ietf-intarea-tunnels] Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-tunnels-10.txt>>.
- [I-D.ietf-lisp-introduction] Cabellos, A. and D. S. (Ed.), "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-introduction-15, 20 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-lisp-introduction-15.txt>>.

[I-D.ietf-lisp-mn]

Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", Work in Progress, Internet-Draft, draft-ietf-lisp-mn-11, 30 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-lisp-mn-11.txt>>.

[I-D.ietf-lisp-rfc6830bis]

Farinacci, D., Fuller, V., Meyer, D., Lewis, D., and A. Cabellos, "The Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6830bis-36, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6830bis-36.txt>>.

[I-D.ietf-lisp-rfc6833bis]

Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, "Locator/ID Separation Protocol (LISP) Control-Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6833bis-30, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6833bis-30.txt>>.

[I-D.jia-intarea-scenarios-problems-addressing]

Jia, Y., Trossen, D., Iannone, L., Shenoy, N., Mendes, P., 3rd, D. E. E., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-scenarios-problems-addressing-02, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-scenarios-problems-addressing-02.txt>>.

[I-D.rafiiee-6man-cga-attack]

Rafiiee, H. and C. Meinel, "Possible Attack on Cryptographically Generated Addresses (CGA)", Work in Progress, Internet-Draft, draft-rafiiee-6man-cga-attack-03, 8 May 2015, <<https://www.ietf.org/archive/id/draft-rafiiee-6man-cga-attack-03.txt>>.

[ICNIP]

Trossen, D., Robitzsch, S., Reed, M., Al-Naday, M., and J. Riihijarvi, "Internet Services over ICN in 5G LAN Environments", Work in Progress, Internet-Draft, draft-trossen-icnrg-internet-icn-5glan-04, 1 October 2020, <<https://www.ietf.org/archive/id/draft-trossen-icnrg-internet-icn-5glan-04.txt>>.

- [IPv4pool] "IANA IPv4 Address Space Registry", n.d.,
<<https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>>.
- [ITU9959] Badenhop, C., Fuller, J., Hall, J., Ramsey, B., and M. Rice, "Evaluating ITU-T G.9959 Based Wireless Systems Used in Critical Infrastructure Assets", IFIP Advances in Information and Communication Technology pp. 209-227, DOI 10.1007/978-3-319-26567-4_13, 2015,
<https://doi.org/10.1007/978-3-319-26567-4_13>.
- [ODoH] Kinnear, E., McManus, P., Pauly, T., Verma, T., and C. A. Wood, "Oblivious DNS Over HTTPS", Work in Progress, Internet-Draft, draft-pauly-dprive-oblivious-doh-11, 17 February 2022, <<https://www.ietf.org/archive/id/draft-pauly-dprive-oblivious-doh-11.txt>>.
- [OHTTP] Thomson, M. and C. A. Wood, "Oblivious HTTP", Work in Progress, Internet-Draft, draft-thomson-http-oblivious-02, 24 August 2021, <<https://www.ietf.org/archive/id/draft-thomson-http-oblivious-02.txt>>.
- [ONION] Goldschlag, D., Reed, M., and P. Syverson, "Onion routing", Communications of the ACM Vol. 42, pp. 39-41, DOI 10.1145/293411.293443, February 1999,
<<https://doi.org/10.1145/293411.293443>>.
- [P4] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and D. Walker, "P4: programming protocol-independent packet processors", ACM SIGCOMM Computer Communication Review Vol. 44, pp. 87-95, DOI 10.1145/2656877.2656890, July 2014,
<<https://doi.org/10.1145/2656877.2656890>>.
- [REED] Reed, M., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", 2016 IEEE International Conference on Communications (ICC), DOI 10.1109/icc.2016.7511036, May 2016,
<<https://doi.org/10.1109/icc.2016.7511036>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981,
<<https://www.rfc-editor.org/info/rfc791>>.

- [RFC1752] Bradner, S. and A. Mankin, "The Recommendation for the IP Next Generation Protocol", RFC 1752, DOI 10.17487/RFC1752, January 1995, <<https://www.rfc-editor.org/info/rfc1752>>.
- [RFC1918] Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G. J., and E. Lear, "Address Allocation for Private Internets", BCP 5, RFC 1918, DOI 10.17487/RFC1918, February 1996, <<https://www.rfc-editor.org/info/rfc1918>>.
- [RFC2009] Imielinski, T. and J. Navas, "GPS-Based Addressing and Routing", RFC 2009, DOI 10.17487/RFC2009, November 1996, <<https://www.rfc-editor.org/info/rfc2009>>.
- [RFC2101] Carpenter, B., Crowcroft, J., and Y. Rekhter, "IPv4 Address Behaviour Today", RFC 2101, DOI 10.17487/RFC2101, February 1997, <<https://www.rfc-editor.org/info/rfc2101>>.
- [RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <<https://www.rfc-editor.org/info/rfc2474>>.
- [RFC2663] Srisuresh, P. and M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", RFC 2663, DOI 10.17487/RFC2663, August 1999, <<https://www.rfc-editor.org/info/rfc2663>>.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <<https://www.rfc-editor.org/info/rfc2865>>.
- [RFC3118] Droms, R., Ed. and W. Arbaugh, Ed., "Authentication for DHCP Messages", RFC 3118, DOI 10.17487/RFC3118, June 2001, <<https://www.rfc-editor.org/info/rfc3118>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.

- [RFC4014] Droms, R. and J. Schnizlein, "Remote Authentication Dial-In User Service (RADIUS) Attributes Suboption for the Dynamic Host Configuration Protocol (DHCP) Relay Agent Information Option", RFC 4014, DOI 10.17487/RFC4014, February 2005, <<https://www.rfc-editor.org/info/rfc4014>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4581] Bagnulo, M. and J. Arkko, "Cryptographically Generated Addresses (CGA) Extension Field Format", RFC 4581, DOI 10.17487/RFC4581, October 2006, <<https://www.rfc-editor.org/info/rfc4581>>.
- [RFC4982] Bagnulo, M. and J. Arkko, "Support for Multiple Hash Algorithms in Cryptographically Generated Addresses (CGAs)", RFC 4982, DOI 10.17487/RFC4982, July 2007, <<https://www.rfc-editor.org/info/rfc4982>>.
- [RFC5533] Nordmark, E. and M. Bagnulo, "Shim6: Level 3 Multihoming Shim Protocol for IPv6", RFC 5533, DOI 10.17487/RFC5533, June 2009, <<https://www.rfc-editor.org/info/rfc5533>>.
- [RFC5795] Sandlund, K., Pelletier, G., and L-E. Jonsson, "The RObusT Header Compression (ROHC) Framework", RFC 5795, DOI 10.17487/RFC5795, March 2010, <<https://www.rfc-editor.org/info/rfc5795>>.
- [RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, DOI 10.17487/RFC6250, May 2011, <<https://www.rfc-editor.org/info/rfc6250>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <<https://www.rfc-editor.org/info/rfc6282>>.

- [RFC6740] Atkinson, R.J. and S.N. Bhatti, "Identifier-Locator Network Protocol (ILNP) Architectural Description", RFC 6740, DOI 10.17487/RFC6740, November 2012, <<https://www.rfc-editor.org/info/rfc6740>>.
- [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for Constrained-Node Networks", RFC 7228, DOI 10.17487/RFC7228, May 2014, <<https://www.rfc-editor.org/info/rfc7228>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<https://www.rfc-editor.org/info/rfc7258>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7400] Bormann, C., "6LoWPAN-GHC: Generic Header Compression for IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs)", RFC 7400, DOI 10.17487/RFC7400, November 2014, <<https://www.rfc-editor.org/info/rfc7400>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC7872] Gont, F., Linkova, J., Chown, T., and W. Liu, "Observations on the Dropping of Packets with IPv6 Extension Headers in the Real World", RFC 7872, DOI 10.17487/RFC7872, June 2016, <<https://www.rfc-editor.org/info/rfc7872>>.
- [RFC8060] Farinacci, D., Meyer, D., and J. Snijders, "LISP Canonical Address Format (LCAF)", RFC 8060, DOI 10.17487/RFC8060, February 2017, <<https://www.rfc-editor.org/info/rfc8060>>.

- [RFC8061] Farinacci, D. and B. Weis, "Locator/ID Separation Protocol (LISP) Data-Plane Confidentiality", RFC 8061, DOI 10.17487/RFC8061, February 2017, <<https://www.rfc-editor.org/info/rfc8061>>.
- [RFC8105] Mariager, P., Petersen, J., Ed., Shelby, Z., Van de Logt, M., and D. Barthel, "Transmission of IPv6 Packets over Digital Enhanced Cordless Telecommunications (DECT) Ultra Low Energy (ULE)", RFC 8105, DOI 10.17487/RFC8105, May 2017, <<https://www.rfc-editor.org/info/rfc8105>>.
- [RFC8163] Lynn, K., Ed., Martocci, J., Neilson, C., and S. Donaldson, "Transmission of IPv6 over Master-Slave/Token-Passing (MS/TP) Networks", RFC 8163, DOI 10.17487/RFC8163, May 2017, <<https://www.rfc-editor.org/info/rfc8163>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.
- [RFC8376] Farrell, S., Ed., "Low-Power Wide Area Network (LPWAN) Overview", RFC 8376, DOI 10.17487/RFC8376, May 2018, <<https://www.rfc-editor.org/info/rfc8376>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8484] Hoffman, P. and P. McManus, "DNS Queries over HTTPS (DoH)", RFC 8484, DOI 10.17487/RFC8484, October 2018, <<https://www.rfc-editor.org/info/rfc8484>>.
- [RFC8609] Mosko, M., Solis, I., and C. Wood, "Content-Centric Networking (CCNx) Messages in TLV Format", RFC 8609, DOI 10.17487/RFC8609, July 2019, <<https://www.rfc-editor.org/info/rfc8609>>.
- [RFC8724] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <<https://www.rfc-editor.org/info/rfc8724>>.

- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <<https://www.rfc-editor.org/info/rfc8926>>.
- [RFC8939] Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020, <<https://www.rfc-editor.org/info/rfc8939>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [RFC8986] Filsfils, C., Ed., Camarillo, P., Ed., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "Segment Routing over IPv6 (SRv6) Network Programming", RFC 8986, DOI 10.17487/RFC8986, February 2021, <<https://www.rfc-editor.org/info/rfc8986>>.
- [ROBUSTSDN] Canini, M., Kuznetsov, P., Levin, D., and S. Schmid, "A distributed and robust SDN control plane for transactional network updates", 2015 IEEE Conference on Computer Communications (INFOCOM), DOI 10.1109/infocom.2015.7218382, April 2015, <<https://doi.org/10.1109/infocom.2015.7218382>>.
- [ROHC] Fitzek, F., Rein, S., Seeling, P., and M. Reisslein, "RObust Header Compression (ROHC) Performance for Multimedia Transmission over 3G/4G Wireless Networks", Wireless Personal Communications Vol. 32, pp. 23-41, DOI 10.1007/s11277-005-7733-2, January 2005, <<https://doi.org/10.1007/s11277-005-7733-2>>.
- [SFCANYCAST] Wion, A., Bouet, M., Iannone, L., and V. Conan, "Distributed Function Chaining with Anycast Routing", Proceedings of the 2019 ACM Symposium on SDN Research, DOI 10.1145/3314148.3314355, April 2019, <<https://doi.org/10.1145/3314148.3314355>>.

- [SPHINX] Danezis, G. and I. Goldberg, "Sphinx: A Compact and Provably Secure Mix Format", 2009 30th IEEE Symposium on Security and Privacy, DOI 10.1109/sp.2009.15, May 2009, <<https://doi.org/10.1109/sp.2009.15>>.
- [TOR] "The Tor Project", n.d., <<https://www.torproject.org/>>.
- [TRANSACTIONSDN] Curic, M., Despotovic, Z., Hecker, A., and G. Carle, "Transactional Network Updates in SDN", 2018 European Conference on Networks and Communications (EuCNC), DOI 10.1109/eucnc.2018.8442793, June 2018, <<https://doi.org/10.1109/eucnc.2018.8442793>>.
- [UA-DHCP] Komori, T. and T. Saito, "The secure DHCP system with user authentication", 27th Annual IEEE Conference on Local Computer Networks, 2002. Proceedings. LCN 2002., DOI 10.1109/lcn.2002.1181774, n.d., <<https://doi.org/10.1109/lcn.2002.1181774>>.
- [VPN] Khanvilkar, S. and A. Khokhar, "Virtual private networks: an overview with performance evaluation", IEEE Communications Magazine Vol. 42, pp. 146-154, DOI 10.1109/mcom.2004.1341273, October 2004, <<https://doi.org/10.1109/mcom.2004.1341273>>.
- [WireGuard] Donenfeld, J., "WireGuard: Next Generation Kernel Network Tunnel", Proceedings 2017 Network and Distributed System Security Symposium, DOI 10.14722/ndss.2017.23160, 2017, <<https://doi.org/10.14722/ndss.2017.23160>>.

Acknowledgments

Thanks to all the people that shared insightful comments both privately to the authors as well as on various mailing list, especially on the INTArea Mailing List. Also thanks for the interesting discussions to Carsten Borman, Brian E. Carpenter.

Authors' Addresses

Yihao Jia
Huawei Technologies Co., Ltd
156 Beiqing Rd.
Beijing
100095
P.R. China
Email: jiayihao@huawei.com

Dirk Trossen
Huawei Technologies Duesseldorf GmbH
Riesstr. 25C
80992 Munich
Germany
Email: dirk.trossen@huawei.com

Luigi Iannone
Huawei Technologies France S.A.S.U.
18, Quai du Point du Jour
92100 Boulogne-Billancourt
France
Email: luigi.iannone@huawei.com

Paulo Mendes
Airbus
Willy-Messerschmitt Strasse 1
81663 Munich
Germany
Email: paulo.mendes@airbus.com

Nirmala Shenoy
Rochester Institute of Technology
New-York, 14623
United States of America
Email: nxsvks@rit.edu

Laurent Toutain
IMT-Atlantique
2 rue de la Chataigneraie
CS 17607
35576 Cesson-Sevigne Cedex
France
Email: laurent.toutain@imt-atlantique.fr

Abraham Y. Chen
Avinta Communications, Inc.
142 N. Milpitas Blvd.
Milpitas, CA, 95035-4401
United States of America
Email: AYChen@Avinta.com

Dino Farinacci
lispers.net
United States of America
Email: farinacci@gmail.com

Internet Area Working Group
Internet-Draft
Intended status: Informational
Expires: 7 September 2022

Y. Jia
D. Trossen
L. Iannone
Huawei
N. Shenoy
R.I.T.
P. Mendes
Airbus
D. Eastlake 3rd
Futurewei
P. Liu
China Mobile
D. Farinacci
lispers.net
6 March 2022

Challenging Scenarios and Problems in Internet Addressing
draft-jia-intarea-scenarios-problems-addressing-03

Abstract

The Internet Protocol (IP) has been the major technological success in information technology of the last half century. As the Internet becomes pervasive, IP has been replacing communication technology for many domain-specific solutions. However, domains with specific requirements as well as communication behaviors and semantics still exist and represent what [RFC8799] recognizes as "limited domains".

This document describes well-recognized scenarios that showcase possibly different addressing requirements, which are challenging to be accommodated in the IP addressing model. These scenarios highlight issues related to the Internet addressing model and call for starting a discussion on a way to re-think/evolve the addressing model so to better accommodate different domain-specific requirements.

The issues identified in this document are complemented and deepened by a detailed gap analysis in a separate companion document [I-D.jia-intarea-internet-addressing-gap-analysis].

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 <https://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 7 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Communication Scenarios in Limited Domains	5
2.1. Communication in Constrained Environments	5
2.2. Communication within Dynamically Changing Topologies	7
2.3. Communication among Moving Endpoints	10
2.4. Communication Across Services	13
2.5. Communication Traffic Steering	14
2.6. Communication with built-in security	15
2.7. Communication protecting user privacy	16
2.8. Communication in Alternative Forwarding Architectures	16
3. Desired Network Features	18
4. Issues in Addressing	21
5. Problem Statement	23
6. Security Considerations	25
7. IANA Considerations	25
8. References	25
8.1. Normative References	25
8.2. Informative References	25
Acknowledgments	33
Authors' Addresses	33

1. Introduction

The Internet Protocol (IP), positioned as the unified protocol at the (Internet) network layer, is seen by many as key to the innovation stemming from Internet-based applications and services. Even more so, with the success of TCP/IP protocol stack, IP has been gradually replacing existing domain-specific protocols, evolving into the core protocol of the entire communication eco-system. At its inception, roughly 40 years ago [RFC0791], the Internet addressing system, represented in the form of the IP address and its locator-based (topological) semantics, has brought the notion of a 'common namespace for all communication'. Compared to proprietary technology-specific solutions, such 'common namespace for all communication' advance ensures end-to-end communication from any device connected to the Internet to another.

However, use cases, associated services, node behaviors, and requirements on packet delivery have since been significantly extended, with the Internet technology being developed to accommodate them in the framework of addressing that stood at the beginning of the Internet's development. This evolution is reflected in the concept of "Limited Domains", first introduced in [RFC8799]. It refers to a single physical network, attached to or running in parallel with the Internet, or is defined by a set of users and nodes distributed over a much wider area, but drawn together by a single virtual network over the Internet. Key to a limited domain is that requirements, behaviors, and semantics could be noticeable local and, more importantly, specific to the limited domain. Very often, the realization of a limited domain is defined by specific communication scenario(s) and/or use case(s) that exhibit the domain-specific behaviors and pose the requirements that lead to the establishment of the limited domain. Identifying limited domains may sometime be not obvious because of blurry boundaries depending on the point of view. For instance, from an end user perspective there is no vision at all on limited domains, hence for end users the dichotomy Internet vs limited domains more transparent. In such cases, it is harder to ensure (and detect) that no limited domain specific semantics leak in the Internet or other limited domains.

One key architectural aspect, when communicating within limited domains, is that of addressing and, therefore, the address structure, as well as the semantic that is being used for packet forwarding (e.g., service identification, content location, device type). The topological location centrality of IP is fundamental when reconciling the often differing semantics for 'addressing' that can be found in those limited domains. The result of this fundamental role of the single IP addressing is that limited domains have to adopt specific solutions, e.g., translating/mapping/converting concepts, semantics, and ultimately, domain-specific addressing, into the common IP addressing used across limited domains.

This document advocates flexibility in addressing in order to accommodate limited domain specific semantics, while, if possible, ensuring a single holistic addressing scheme able to reduce, or even entirely remove, the need for aligning the address semantics of different limited domains, such as the current topological location semantic of the Internet. Ultimately, such holistic addressing could be beneficial to those communication scenarios realized within limited domains by improving efficiency, removing of constraints imposed by needing to utilize the limited semantics of IP addressing, and/or in other ways.

In other words, this document revolves around the following question:

"Should interconnected limited domains purely rely on IP addresses and therefore deal with the complexity of translating any semantic mismatch themselves, or should flexibility for supporting those limited domains be a key focus for an evolved Internet addressing?"

To that end, this document describes well-recognized scenarios in limited domains that could benefit from greater flexibility in addressing and overviews the problems encountered throughout these scenarios due to the lack of that flexibility. A detailed gap analysis can be found in {I-D.jia-intarea-internet-addressing-gap-analysis}}, which elaborates on the issues identified in this memo in reference to extensions to Internet addressing that have attempted to address those issues. The purpose of this memo is rather to stimulate discussion on the emerging needs for addressing at large with the possibility to fundamentally re-think the addressing in the Internet beyond the current objectives of IPv6 [RFC8200].

It is important to remark that any change in the addressing, hence at the data plane level, leads to changes and challenges at the control plane level, i.e., routing. The latter is an even harder problem than just addressing and might need more research efforts that are beyond the objective of this document, which focuses solely on the data plane.

2. Communication Scenarios in Limited Domains

The following sub-sections outline a number of scenarios, all of which belong to the concept of "limited domains" [RFC8799]. While the list of scenarios may look long, this document focuses on scenarios with a number of aspects that can be observed in those limited domains, captured in the sub-section titles. For each scenario, possible challenges are highlighted, which are then picked upon in Section 4, when describing more formally the existing shortcomings in current Internet addressing.

2.1. Communication in Constrained Environments

In a number of communication scenarios, such as those encountered in the Internet of Things (IoT), a simple, communication network demanding minimal resources is required, allowing for a group of IoT network devices to form a network of constrained nodes, with the participating network and end nodes requiring as little computational power as possible and having small memory requirements in order to reduce the total cost of ownership of the network. Furthermore, in the context of industrial IoT, real-time requirements and scalability make IP technology not naturally suitable as communication technology ([OCADO]).

In addition to IEEE 802.15.4, i.e., Low-Rate Wireless Personal Area Network [LR-WPAN], several limited domains exist through utilizing link layer technologies such as Bluetooth Low Energy (BLE) [BLE], Digital European Cordless Telecommunications (DECT) - Ultra Low Energy (ULE) [DECT-ULE], Master-Slave/Token-Passing (MS/TP) [BACnet], Near-Field-Communication (NFC) [ECMA-340], and Power Line Communication (PLC) [IEEE_1901.1].

The end-to-end principle (detailed in [RFC2775]) requires IP addresses (e.g., IPv6 [RFC8200]) to be used on such constrained nodes networks, allowing IoT devices using multiple communication technologies to talk on the Internet. Often, devices located at the edge of constrained networks act as gateway devices, usually performing header compression ([RFC4919]). To ensure security and reliability, multiple gateways must be deployed. IoT devices on the network must select one of those gateways for traffic passthrough by the devices on the (limited domain) network.

Given the constraints imposed on the computational and possibly also communication technology, the usage of a single addressing semantic in the form of a 128-bit endpoint identifier, i.e., IPv6 address, may pose a challenge when operating such networks.

Another type of (differently) constrained environment is an aircraft, which encompasses not only passenger communication but also the integration of real-time data exchange to ensure that processes and functions in the cabin are automatically monitored or actuated. The goal for any aircraft network is to be able to send and receive information reliably and seamlessly. From this perspective, the medium with which these packets of information are sent is of little consequence so long as there is a level of determinism to it. However, there is currently no effective method in implementing wireless inter- and intra-communications between all subsystems. The emerging wireless sensor network technology in commercial applications such as smart thermostat systems, and smart washer/dryer units could be transposed onto aircraft and fleet operations. The proposal for having an Wireless Avionics Intra-Communications (WAIC) system promises reduction in the complexity of electrical wiring harness design and fabrication, reduction in wiring weight, increased configuration, and potential monitoring of otherwise inaccessible moving or rotating aircraft parts. Similar to the IoT concept, WAIC systems consist of short-range communications and are a potential candidate for passenger entertainment systems, smoke detectors, engine health monitors, tire pressure monitoring systems, and other kinds of aircraft maintenance systems.

While there are still many obstacles in terms of network security, traffic control, and technical challenges, future WAIC can enable real-time seamless communications between aircraft and between ground teams and aircraft as opposed to the discrete points of data leveraged today in aircraft communications. For that, WAIC infrastructure should also be connected to outside IP based networks in order to access edge/cloud facilities for data storage and mining. However, the restricted capacity (energy, communication) of most aircraft devices (e.g. sensors) and the nature of the transmitted data - periodic transmission of small packets - may pose some challenges for the usage of a single addressing semantic in the form of a 128-bit endpoint identifier, i.e., an IPv6 address. Moreover, most of the aircraft applications and services are focused on the data (e.g. temperature of gas tank on left wing) and not on the topological location of the data source. This means that the current topological location semantic of IP addresses is not beneficial for aircraft applications and services.

Greater flexibility in Internet addressing may avoid complex and energy hungry operations, like header compression and fragmentation, necessary to translate protocol headers from one limited domain to another, while enabling semantics different from locator-based addressing may better support the communication that occurs in those environments.

2.2. Communication within Dynamically Changing Topologies

Communication may occur over networks that exhibit dynamically changing topologies. One such example is that of satellite networks, providing global Internet connections through a combination of inter-satellite and ground station communication. With the convergence of space-based and terrestrial networks, users can experience seamless broadband access, e.g., on cruise ships, flights, and within cars, often complemented by and seamlessly switching between Wi-Fi, cellular, or satellite based networks at any time [WANG19].

The satellite network service provider will plan the transmission path of user traffic based on the network coverage, satellite orbit, route, and link load, providing potentially high-quality Internet connections for users in areas that are not, or hard to be, covered by terrestrial networks. With large scale LEO (Low Earth Orbit) satellites, the involved topologies of the satellite network will be changing constantly while observing a regular flight pattern in relation to other satellites and predictable overflight patterns to ground users [CHEN21].

Although satellite bearer services are capable of transporting IPv4 and IPv6, as well as associated protocols such as IP Multicast, DNS services and routing information, no IP functionality is implemented on-board the spacecraft limiting the capability of leveraging for instance large scale satellite constellations.

One of the major constraints of deploying routing capability on board of a satellite is power consumption. Due to this, space routers may end up being intermittently powered up during a daytime sunlit pass. Another limitation of the first generation of IP routers in space was the lack of capability to remotely manage and upgrade software while in operation.

The limitations faced in early development of IP based satellite communication payloads, showed the need to develop a flexible networking solution that would enable delay tolerant communications in the presence of intermittent connectivity. Further, in order to reduce latency, which is the major impairment of satellite networks, there was a need of a networking solution able to perform in a scenario encompassing mobile devices with the capability of storing data, leading to a significant reduction of latency, which is the major impairment of satellite networks.

Moreover, due to the current IP addressing scheme and its focus on IP unicast addressing with extended deployment of IP multicast and some IP anycast, current deployments do not take advantage of the broadcast nature of satellite networks.

Moreover networking platforms based on a name (data or service) based addressing scheme would bring several potential benefits to satellite networks aiming to tackle their major challenges, including high propagation delay and changing network topology in the case of LEO constellations.

Another example is that of vehicular communication, where services may be accessed across vehicles, such as self-driving cars, for the purpose of collaborative objection recognition (e.g., for collision avoidance), road status conveyance (e.g., for pre-warning of road-ahead conditions), and other purposes. Communication may include Road Side Units (RSU) with the possibility to create ephemeral connections to those RSUs for the purpose of workload offloading, joint computation over multiple (vehicular) inputs, and other purposes [I-D.ietf-lisp-nexagon]. Communication here may exhibit a multi-hop nature, not just involving the vehicle and the RSU over a direct link. Those topologies are naturally changing constantly due to the dynamic nature of the involved communication nodes.

The advent of Flying Ad-hoc NETworks (FANETs) has opened up an opportunity to create new added-value services [CHRIKI19]. Although these networks share common features with vehicular ad hoc networks, they present several unique characteristics such as energy efficiency, mobility degree, the capability of swarming, and the potential large scale of swarm networks. Due to high mobility of FANET nodes, the network topology changes more frequently than in a typical vehicular ad hoc network. From a routing point of view, although ad-hoc reactive and proactive routing approaches can be used, there are other type of routing protocols that have been developed for FANETS, such as hybrid routing protocols and position based routing protocols, aiming to increase efficiency in large scale networks with dynamic topologies.

Both type of protocols challenge the current Internet addressing semantic: in the case of hybrid protocols, two different routing strategies are used inside and outside a network zone. While inside a zone packets are routed to a specific destination IP address, between zones, query packets are routed to a subset of neighbors as determined by a broadcast algorithm. In the case of position based routing protocol, the IP addressing scheme is not used at all, since packets are routed to a different identifier, corresponding to the geographic location of the destination and not its topological location. Hence, what is needed is to consolidate the geo-spatial addressing with that of a locator-based addressing in order to optimize routing policies across the zones.

Moreover most of the application/services deployed in FANETs tend to be agnostic of the topological location of nodes, rather focusing on the location of data or services. This distinction is even more important because in dynamic network such as FANET robust networking solutions may rely on the redundancy of data and services, meaning that they may be found in more than one device in the network. This in turn may bring into play a possible service-centric semantic for addressing the packets that need routing in the dynamic network towards a node providing said service (or content).

In the aforementioned network technologies, there is a significant difference between the high dynamics of the underlying network topologies, compared to the relative static nature of terrestrial network topology, as reported in [HANDLEY]. As a consequence, the notion of a topological network location becomes restrictive in the sense that not only the relation between network nodes and user endpoint may change, but also the relation between the nodes that form the network itself. This may lead to the challenge of maintaining and updating the topological addresses in this constantly changing network topology.

In attempts to utilize entirely different semantics for the addressing itself, geographic-based routing, such as in [CARTISEAN], has been proposed for MANETs (Mobile Ad-hoc NETWORKs) through providing geographic coordinates based addresses to achieve better routing performance, lower overhead, and lower latency [MANET1].

Flexibility in Internet addressing here would allow for accommodating such geographic address semantics into the overall Internet addressing, while also enabling name/content-based addressing, utilizing the redundancy of many network locations providing the possible data.

2.3. Communication among Moving Endpoints

When packet switching was first introduced, back in the 60s/70s, it was intended to replace the rigid circuit switching with a communication infrastructure that was more resilient to failures. As such, the design never really considered communication endpoints as mobile. Even in the pioneering ALOHA [ALOHA] system, despite considering wireless and satellite links, the network was considered static (with the exception of failures and satellites, which fall in what is discussed in Section 2.2). Ever since, a lot of efforts have been devoted to overcome such limitations once it became clear that endpoint mobility will become a main (if not THE main) characteristic of ubiquitous communication systems.

The IETF has for a long time worked on solutions that would allow extending the IP layer with mobility support. Because of the topological semantic of IP addresses, endpoints need to change addresses each time they visit a different network. However, because routing and endpoint identification is also IP address based, this leads to a communication disruption.

To cope with such a situation, sometimes, the transport layer gets involved in mobility solutions, either by introducing explicit in-band signaling to allow for communicating IP address changes (e.g., in SCTP [RFC5061] and MPTCP [RFC6182]), or by introducing some form of connection ID that allows for identifying a communication independently from IP addresses (e.g., the connection ID used in QUIC [RFC9000]).

Concerning network layer only solutions, anchor-based Mobile IP mechanisms have been introduced ([RFC5177], [RFC6626] [RFC5944], [RFC5275]). Mobile IP is based on a relatively complex and heavy mechanism that makes it hard to deploy and it is not very efficient. Furthermore, it is even less suitable than native IP in constrained environments like the ones discussed in Section 2.1.

Alternative approaches to Mobile IP often leverage the introduction of some form of overlay. LISP [I-D.ietf-lisp-introduction], by separating the topological semantic from the identification semantic of IP addresses, is able to cope with endpoint mobility by dynamically mapping endpoint identifiers with routing locators [I-D.ietf-lisp-mn]. This comes at the price of an overlay that needs its own additional control plane [I-D.ietf-lisp-rfc6833bis].

Similarly, the NV03 (Network Virtualization Overlays) Working Group, while focusing on Data Center environments, also explored an overlay-based solution for multi-tenancy purposes, but also resilient to mobility since relocating Virtual Machines (VMs) is common practice.

NVO3 considered for a long time several data planes that implement slightly different flavors of overlays ([RFC8926], [RFC7348], [I-D.ietf-intarea-gue]), but lacks an efficient control plane specifically tailored for DCs.

Alternative mobility architectures have also been proposed in order to cope with endpoint mobility outside the IP layer itself. The Host Identity Protocol (HIP) [RFC7401] introduced a new namespace in order to identify endpoints, namely the Host Identity (HI), while leveraging the IP layer for topological location. On the one hand, such an approach needs to revise the way applications interact with the network layer, by modifying the DNS (now returning an HI instead of an IP address) and applications to use the HIP socket extension. On the other hand, early adopters do not necessarily gain any benefit unless all communicating endpoints upgrade to use HIP. In spite of this, such a solution may work in the context of a limited domain.

Another alternative approach is adopted by Information-Centric Networking (ICN) [RFC7476]. By making content a first class citizen of the communication architecture, the "what" rather than the "where" becomes the real focus of the communication. However, as explained in the next sub-section, ICN can run either over the IP layer or completely replace it, which in turn can be seen as running the Internet and ICN as logically completely separated limited domains.

Unmanned Aircraft Systems (UAS) are examples of moving devices that require a stable mobility management scheme since they consist of a number of Unmanned Aerial Vehicles (UAV) subordinated to a Ground Control Station (GCS) [MAROJEVIC20]. The information produced by the different sensors and electronic devices available at each UAV is collected and processed by a software or hardware data acquisition unit, being transmitted towards the GCS, where it is inspected and/or analyzed. Analogously, control information transmitted from the GCS to the UAV enables the execution of control operations over the aircraft, such as changing the route planning or the direction pointed by a camera.

Although UAVs may have redundant links to maintain communications in long-range missions (e.g., satellite), most of the communications between the GCS and the UAVs take place over wireless data links, e.g., based on a radio line-of-sight technology, Wi-Fi or 3G/4G/5G. While in some scenarios, UAVs will operate always under the range of the same cellular base station, in missions with large range, UAVs will move between different cellular or wireless ground infrastructure, meaning that the UAV needs to upload its topological locator and re-start the ongoing communication sessions. In such cases, most of existing Mobile IP approaches may play a role, as well as approaches to split the UAV identifier and the topological locator, such as HIP.

However, while the industry is given the first steps towards evolved UAS architectures and communication models, the data-centric communication plays an increasing role, where information is named and decoupled from its location, and applications/services operate over these named data rather than on host-to-host communications.

In this context, the Data Distribution Service ([DDS]) has emerged as an industry-oriented open standard that follows this approach. The space and time decoupling allowed by DDS is very relevant in any dynamic and distributed system, since interacting entities are not forced to know each other and are not forced to be simultaneously present to exchange data. Time decoupling can significantly simplify the management of intermittent data-links, in particular for wireless connectivity between UAS, as well as facilitate seamless UAV mobility between GCSs. This model of communication, in turn, questions the locator-based addressing used in IP and instead utilizes a data-centric naming.

In the case of using TCP/IP, mobility of UAVs introduces a significant challenge. Consider the case where a GCS is receiving telemetry information from a specific UAV. Assuming that the UAV moves and changes its point of attachment to the network, it will have to configure a new IP address on its wireless interface. However, this is problematic, as the telemetry information is still being sent by to the previous IP address of the UAV. This simple example illustrates the necessity to deploy mobility management solutions to handle this type of situations.

However, mobility management solutions increase the complexity of the deployment and may impact the performance of data distribution, both in terms of signaling/data overhead and communication path delay. Considering the specific case of multicast data streams, mobility of content producers and consumers is inherently handled by multicast routing protocols, which are able to react to changes of location of mobile nodes by reconstructing the corresponding multicast delivery

trees. Nevertheless, this comes with a cost in terms of signaling and data overhead (data may still flow through branches of a multicast delivery tree where there are no receivers while the routing protocol is still converging).

Another alternative is to perform the mobility management of producers and consumers not at the application layer based on IP multicast trees, but on the network layer based on an Information Centric Network approach, which was already mentioned in this section.

Greater flexibility in addressing may help in dealing with mobility more efficiently, e.g., through an augmented semantic that may fulfil the mobility requirements [RFC7429] in a more efficient way or through moving from a locator- to a content or service-centric semantic for addressing.

2.4. Communication Across Services

As a communication infrastructure spanning many facets of life, the Internet integrates services and resources from various aspects such as remote collaboration, shopping, content production as well as delivery, education, and many more. Accessing those services and resources directly through URIs has been proposed by methods such as those defined in ICN [RFC7476], where providers of services and resources can advertise those through unified identifiers without additional planning of identifiers and locations for underlying data and their replicas. Users can access required services and resources by virtue of using the URI-based identification, with an ephemeral relationship built between user and provider, while the building of such relationship may be constrained with user- as well as service-specific requirements, such as proximity (finding nearest provider), load (finding fastest provider), and others.

While systems like ICN [CCN] provide an alternative to the topological addressing of IP, its deployment requires an overlay (over IP) or native deployment (alongside IP), the latter with dedicated gateways needed for translation. Underlay deployments are also envisioned in [RFC8763], where ICN solutions are being used to facilitate communication between IP addressed network endpoints or URI-based service endpoints, still requiring gateway solutions for interconnection with ICN-based networks as well as IP routing based networks (cf., [ICN5G][ICNIP]).

Although various approaches combining service and location-based addressing have been devised, the key challenge here is to facilitate a "natural", i.e., direct communication, without the need for gateways above the network layer.

Another aspect of communication across services is that of chaining individual services to a larger service. Here, an identifier would be used that serves as a link to next hop destination within the chain of single services, as done in the work on Service Function Chaining (SFC). With this, services are identified at the level of Layer 2/3 ([RFC7665], [RFC8754], [RFC8595]) or at the level of name-based service identifiers like URLs [RFC8677] although the service chain identification is carried as a Network Service header (NSH) [RFC7665], separate to the packet identifiers. The forwarding with the chain of services utilizes individual locator-based IP addressing (for L3 chaining) to communicate the chained operations from one Service Function Forwarder [RFC7665] to another, leading to concerns regarding overhead incurred through the stacking of those chained identifiers in terms of packet overhead and therefore efficiency in handling in the intermediary nodes.

Greater flexibility in addressing may allow for incorporating different information, e.g., service as well as chaining semantics, into the overall Internet addressing.

2.5. Communication Traffic Steering

Steering traffic within a communication scenario may involve at least two aspects, namely (i) limiting certain traffic towards a certain set of communication nodes and (ii) restraining the sending of packets towards a given destination (or a chain of destinations) with metrics that would allow the selection among one or more possible destinations.

One possibility for limiting traffic inside limited domains, towards specific objects, e.g., devices, users, or group of them, is subnet partition with techniques such as VLAN [RFC5517], VxLAN [RFC7348], or more evolved solution like TeraStream [TERASTREAM] realizing such partitioning. Such mechanisms usually involve significant configuration, and even small changes in network and user nodes could result in a repartition and possibly additional configuration efforts. Another key aspect is the complete lack of correlation of the topological address and any likely more semantic-rich identification that could be used to make policy decisions regarding traffic steering. Suitably enriching the semantics of the packet address, either that of the sender or receiver, so that such decision could be made while minimizing the involvement of higher layer mechanisms, is a crucial challenge for improving on network operations and speed of such limited domain traffic.

When making decisions to select one out of a set of possible destinations for a packet, IP anycast semantics can be applied albeit being limited to the locator semantic of the IP address itself.

Recent work in [SFCANYCAST] suggests utilizing the notion of IP anycast address to encode a "service identifier", which is dynamically mapped onto network locations where service instances fulfilling the service request may be located. Scenarios where this capability may be utilized are provided in [SFCANYCAST] and include, but are not limited to, scenarios such as edge-assisted VR/AR, transportation, smart cities, smart homes, smart wearables, and digital twins.

The challenge here lies in the possible encoding of not only the service information itself but the constraint information that helps the selection of the "best" service instance and which is likely a service-specific constraint in relation to the particular scenario. The notion of an address here is a conditional (on those constraints) one where this conditional part is an essential aspect of the forwarding action to be taken. It needs therefore consideration in the definition of what an address is, what is its semantic, and how the address structure ought to look like.

As outlined in the previous sub-section, chaining services are another aspect of steering traffic along a chain of constituent services, where the chain is identified through either a stack of individual identifiers, such as in Segment Routing [RFC8402], or as an identifier that serves as a link to next hop destination within the chain, such as in Service Function Chaining (SFC). The latter can be applied to services identified at the level of Layer 2/3 ([RFC7665], [RFC8754], [RFC8595]) or at the level of name-based service identifiers like URLs [RFC8677]. However, the overhead incurred through the stacking of those chained identifiers is a concern in terms of packet overhead and therefore efficiency in handling in the intermediary nodes.

Flexibility in addressing may enable more semantic rich encoding schemes that may help in steering traffic at hardware level and speed, without complex mechanisms usually resulting in handling packets in the slow path of routers.

2.6. Communication with built-in security

Today, strong security in the Internet is usually implemented as a general network service ([PILA], [RFC6158]). Among the various reasons for such approach is the limited semantic of current IP addresses, which do not allow to natively express security features or trust relationships. Efforts like Cryptographically Generated Addresses (CGA) [RFC3972], provide some security features by embedding a truncated public key in the last 57-bit of IPv6 address, thereby greatly enhancing authentication and security within an IP network via asymmetric cryptography and IPsec [RFC4301]. The

development of the Host Identity Protocol (HIP) [RFC7401] saw the introduction of cryptographic identifiers for the newly introduced Host Identity (HI) to allow for enhanced accountability, and therefore trust. The use of those HIs, however, is limited by the size of IPv6 128bit addresses.

Through a greater flexibility in addressing, any security-related key, certificate, or identifier could instead be included in a suitable address structure without any information loss (i.e., as-is, without any truncation or operation as such), avoiding therefore compromises such as those in HIP. Instead, CGAs could be created using full length certificates, or being able to support larger HIP addresses in a limited domain that uses it. This could significantly help in constructing a trusted and secure communication at the network layer, leading to connections that could be considered as absolute secure (assuming the cryptography involved is secure). Even more, anti-abuse mechanisms and/or DDoS protection mechanisms like the one under discussion in PEARG ([PEARG]) Research Group may leverage a greater flexibility of the overall Internet addressing, if provided, in order to be more effective.

2.7. Communication protecting user privacy

See Comments in Section "Issues".

2.8. Communication in Alternative Forwarding Architectures

The performance of communication networks has long been a focus for optimization due to the immediate impact on cost of ownership for communication service providers. Technologies like MPLS [RFC3031] have been introduced to optimize lower layer communication, e.g., by mapping L3 traffic into aggregated labels of forwarding traffic for the purposes of, e.g., traffic engineering.

Even further, other works have emerged in recent years that have replaced the notion of packets with other concepts for the same purpose of improved traffic engineering and therefore efficiency gains. One such area is that of Software Defined Networks (SDN) [RFC7426], which has highlighted how a majority of Internet traffic is better identified by flows, rather than packets. Based on such observation, alternate forwarding architectures have been devised that are flow-based or path-based. With this approach, all data belonging to the same traffic stream is delivered over the same path, and traffic flows are identified by some connection or path identifier rather than by complete routing information, possibly enabling fast hardware based switching (e.g. [DETNET], [PANRG]).

On the one hand, such a communication model may be more suitable for real-time traffic like in the context of Deterministic Networks ([DETNET]), where indeed a lot of work has focused on how to "identify" packets belonging to the same DETNET flow in order to jointly manage the forwarding within the desired deterministic boundaries.

On the other hand, it may improve the communication efficiency in constrained wireless environments (cf., Section 2.1), by reducing the overhead, hence increasing the number of useful bits per second per Hz.

Also, the delivery of information across similar flows may be combined into a multipoint delivery of a single return flow, e.g., for scenarios of requests for a video chunk from many clients being responded to with a single (multi-destination) flow, as outlined in [BIER-MC] as an example. Another opportunity to improve communication efficiency is being pursued in ongoing IETF/IRTF work to deliver IP- or HTTP-level packets directly over path-based or flow-based transport network solutions, such as in [TROSSEN][BIER-MC][ICNIP][ICN5G] with the capability to bundle unicast forward communication streams flexibly together in return path multipoint relations. Such capability is particularly opportune in scenarios such as chunk-based video retrieval or distributed data storage. However, those solutions currently require gateways to "translate" the flow communication into the packet-level addressing semantic in the peering IP networks. Furthermore, the use of those alternative forwarding mechanisms often require the encapsulation of Internet addressing information, leading to wastage of bandwidth as well as processing resources.

Providing an alternative way of forwarding data has also been the motivation for the efforts created in the European Telecommunication Standards Institute (ETSI), which formed an Industry Specification Group (ISG) named Non-IP Networking (NIN) [ETSI-NIN]. This group sets out to develop and standardize a set of protocols leveraging an alternative forwarding architecture, such as provided by a flow-based switching paradigm. The deployment of such protocols may be seen to form limited domains, still leaving the need to interoperate with the (packet-based forwarding) Internet; a situation possibly enabled through a greater flexibility of the addressing used across Internet-based and alternative limited domains alike.

As an alternative to IP routing, EIBP (Extended Internet Bypass Protocol) [EIBP] offers a communications model that can work with IP in parallel and entirely transparent and independent to any operation at network layer. For this, EIBP proposes the use of physical and/or virtual structures in networks and among networks to auto assign

routable addresses that capture the relative position of routers in a network or networks in a connected set of networks, which can be used to route the packets between end domains. EIBP operates at Layer 2.5 and provides encapsulation (at source domain), routing, and de-encapsulation (at destination domain) for packets. EIBP can forward any type of packets between domains. A resolver to map the domain ID to EIBP's edge router addresses is required. When queried for a specific domain, the resolver will return the corresponding edge router structured addresses.

EIBP decouples routing operations from end domain operations, offering to serve any domain, without point solutions to specific domains. EIBP also decouples routing IDs or addresses from end device/domain addresses. This allows for accommodation of new and upcoming domains. A domain can extend EIBP's structured addresses into the domain, by joining as a nested domain under one or more edge routers, or by extending the edge router's structure addresses to its devices.

A greater flexibility in addressing semantics may reduce the aforementioned wastage by accommodating Internet addressing in the light of such alternative forwarding architectures, instead enabling the direct use of the alternative forwarding information.

3. Desired Network Features

From the previous subsection, we recognize that Internet technologies are used across a number of scenarios, each of which brings their own (vertical) view on needed capabilities in order to work in a satisfactory manner to those involved.

In the following, we complement those vertical-specific insights with answers to the question of network features that end users (in the form of individuals or organizations alike) desire from the networked system at large. Answers to this question look at the network more from a horizontal perspective, i.e. not with a specific usage in mind beyond communication within and across networks. The text here summarizes the discussion that took place on the INT Area mailing list after IETF112 on this issue. For some of those identified features, we can already identify how innovations on addressing may impact the realization of a particular feature.

We then combine the insights from both scenario-specific and wider horizontal views for the identification of issues when realizing the specific capability of addressing, presented in Section 4.

1. Always-On: The world is getting more and more connected, leading to being connected to the Internet, anywhere, by any technology (e.g., cable, fiber, or radio), even simultaneously, "all the time", and, most importantly, automatically (without any switch turning). However, when defining "all the time" there is a clear and important difference to be made between availability and reliability vs "desired usage". In other words, "always on" can be seen as a desirable perception at the end user level or as a characteristic of the underlying system. From an end user perspective, clearly the former is of importance, not necessarily leading to an "always on" system notion but instead "always-app-available", merely requiring the needed availability and reliability to realize the perception of being "always on" (e.g., for earthquake alerts), possibly complemented by app-specific methods to realize the "always on" perception (e.g., using local caching rather than communication over the network).
2. Transparency: Being agnostic with respect to local domains network protocols (Bluetooth, ZigBee, Thread, Airdrop, Airplay, or any others) is key to provide an easy and straightforward method for contacting people and devices without any knowledge of network issues, particularly those specific to network-specific solutions. While having a flexible addressing model that accommodates a wide range of use cases is important, the centrality of the IP protocol remains key as a mean to provide global connectivity.
3. Multi-homing: Seamless multi-homing capability for the host is key to best use the connectivity options that may be available to an end user, e.g., for increasing resilience in cases of failures of one available option. Protocols like LISP, SHIM6, QUIC, MPTCP, SCTP (to cite a few) have been successful at providing this capability in an incremental way, but too much of that capability is realized within the application, making it hard to leverage across all applications. While today each transport protocol has its own way to perform multi-address discovery, the network layer should provide the multi-homing feature (e.g., SHIM6 can be used to discover all addresses on both ends), and then leave the address selection to the transport. With that, multi-address discovery remains a network feature exposed to the upper layers. This may also mean to update the Socket API (which may be actually the first thing to do), which does not necessarily mean to expose more network details to the applications but instead be more address agnostic yet more expressive.

4. **Mobility:** A lot of work has been put in MobileIP ([RFC5944],[RFC6275]) to provide seamless and lossless communications for moving nodes (vehicle, satellites). However, it has never been widely deployed for several reasons, like complexity of the protocol and the fact that the problem has often been tackled at higher layers, with applications resilient to address changes. However, similar to multi-homing, solving the problem at higher layers means that each and every transport protocol and application have their own way to deal with mobility, leading to similar observations as those for the previous multi-homing aspect.
5. **Security and Privacy:** The COVID-19 pandemic has boosted end users' desire to be protected and protect their privacy. The balance among privacy, security, and accountability is not simple to achieve. There exist different views on what those properties should be, however the network should provide the means to provide what is felt as the best trade-off for the specific use case.
6. **Performance:** While certainly desirable, "performance" is a complex issue that depends on the objectives of those building for but also paying for performance. Examples are (i) speed (shorter paths/direct communications), (ii) bandwidth (10petabit/s for a link), (iii) efficiency (less overlays/encapsulations), (iv) high efficacy or sustainability (avoid waste). From an addressing perspective, length/format/semantics that may adapt to the specific use case (e.g. use short addresses for low power IoT, or, where needed, longer for addresses embedding certificates for strong authentication, authorization and accountability) may contribute to the performance aspects that end users desire, such as reducing waste through not needed encapsulation or needed conversion at network boundaries.
7. **Availability, Reliability, Predictability:** These three properties are important to enable wide-range of services and applications according to the desired usage (cf. point 1).
8. **Do not do harm:** Access to the Internet is considered a human right [RFC8280]. Access to and expression through it should align with this core principle. This issue transcends through a variety of previously discussed 'features' that are desired, such as privacy, security but also availability and reliability. However, lifting the feature of network access onto a basic rights level also brings in the aspect of "do not do harm" through the use of the Internet with respect to wider societal objectives. Similar to other industries, such as electricity or cars, preventing harm usually requires an interplay of

commercial, technological, and regulatory efforts, such as the enforcement of seat belt wearing to reduce accident death. As a first step, the potential harmfulness of a novel method must be recognized and weighted against the benefits of its introduction and use. One increasingly important consideration in the technology domain is "sustainability" of resource usage for an end user's consumption of and participation in Internet services. As an example, Distributed Ledger Technologies (DLT) are seen as an important tool for a variety of applications, including Internet decentralization ([DINRG]). However, the non-linear increase in energy consumption means that extending proof-of-work systems to the entire population of the planet would not only be impractical but also possibly highly wasteful, not just at the level of computational but also communication resource usage [DLT-draft]. This poses the question on how novel methods for addressing may improve on sustainability of such technologies, particularly if adopted more widely.

9. Maximum Transmission Unit (MTU): One long standing issue in the Internet is related to the MTU and how to discover the path MTU in order to avoid fragmentation ([I-D.ietf-6man-mtu-option], [I-D.templin-6man-aero]). While it makes sense to always leverage as much performance from local systems as possible, this should come without sacrificing the ability to communicate with all systems. Having a solid solution to solve the issue would make the overall interconnection of systems more robust.

4. Issues in Addressing

The desired properties outlined in the previous section have implications that go beyond addressing and need to be tackled from a larger architectural point of view. Such a discussion is left as future action, limiting the present document at discussing only the addressing viewpoint and identifying shortcomings perceived from this perspective.

There are a number of issues that we can identify from the communication scenarios in Section 2 and the network features generally desire from the network, presented in Section 3. We do not claim to be exhaustive in our list:

1. Limiting Alternative Address Semantics: Several communication scenarios pursue the use of alternative semantics of what constitute an 'address' of a packet traversing the Internet, which may fall foul of the defined network interface semantic of IP addresses.

2. **Hampering Security:** Aligning with the semantic and length limitations of IP addressing may hamper the security objectives of any new semantic, possibly leading to detrimental effects and possible other workarounds (at the risk of introducing fragility rather than security).
1. **Hampering Privacy:**
 - * Easy individual identification
 - * Flow linkability
 - * App/Activity profiling
2. **Complicating Traffic Engineering:** Utilizing a plethora of non-address inputs into the traffic steering decision in real networks complicates traffic engineering in that it makes the development of suitable policies more complex, while also leading to possible contention between methods being used.
3. **Hampering Efficiency:** Extending IP addressing through point-wise solutions also hampers efficiency, e.g., through needed re-encapsulation (therefore increasing the header processing overhead as well as header-to-payload ratio), through introducing path stretch, or through requiring compression techniques to reduce the header proportion of large addresses when operating in constrained environments.
4. **Fragility:** The introduction of point solutions, each of which comes with possibly own usages of address or packet fields, together with extension-specific operations, increases the overall fragility of the resulting system, caused, for instance, through contention between feature extensions that were neither foreseen in the design nor tested during the implementation phase.
5. **Extensibility:** Accommodating new requirements through ever new extensions as an extensibility approach to addressing compounds aspects discussed before, i.e., fragility, efficiency etc. It complicates engineering due to the clearly missing boundaries against which contentions with other extensions could be managed. It complicates standardization since extension-based extensibility requires independent, and often lengthy, standardization processes. And ultimately, deployments are complicated due to backward compatibility testing required for any new extension being integrated into the deployed system.

The table below shows how the above identified issues do arise somehow in our outlined communication scenarios in Section 2. This overview will be deepened in more details in the gap analysis document [I-D.jia-intarea-internet-addressing-gap-analysis].

	Issue 1	Issue 2	Issue 3	Issue 4	Issue 5	Issue 6
Constrained Environments				x	x	x
Dynamically Changing Topologies	x		x	x	x	x
Moving Endpoints	x		x	x	x	x
Across Services	x		x	x	x	x
Traffic Steering	x		x	x	x	x
Built-in Security	x	x		x	x	x
Alternative Forwarding Architectures	x			x		x

Table 1: Issues Involved in Challenging Scenarios

5. Problem Statement

This document identifies a number of scenarios as well as general features end users would want from the network, positioning the existing Internet addressing structure itself as a potential hindrance in solving key problems for Internet service provisioning. Such problems include supporting new, e.g., service-oriented, scenarios more efficiently, with improved security and efficient traffic engineering, as well as large scale mobility. We can observe that those new forms of communication are particularly driven by the conceptual framework of limited domains, realizing the requirements of stakeholders for an optimized communication in those limited domains, while still utilizing the Internet for interconnection as

well as for access to the wealth of existing Internet services.

This co-existence of optimized LD-level as well as Internet communication creates a tussle between those requirements on addressing stemming from those limited domains and those coming from the Internet in the form of agreed IPv6 semantics. This tussle directly refers back to our introductory question on flexibility in addressing (or leaving the problem to limited domain solutions to deal with). It is also captured in the discussion on where new features are being introduced, i.e. at the edge or core of the Internet.

But more importantly, the question on 'what is an address anyway' (derived from what features we may want from the network) should not be guided by the answers that the Internet can give us today, e.g., being a mere ephemeral token for accessing PoP-based services (as indicated in related arch-d mailing list discussions), but instead what features could be enabled by a particular view of what an address is. However, that is not to 'second guess' the market and its possible evolution, but to outline clear features from which to derive clear principles for a design.

For this, it is important to recognize that skewing the technical capabilities of any feature, let alone addressing, to the current economic situation of the Internet bears the danger of locking down innovation capabilities as an outcome of those technical limitations being introduced. Instead, addressing must align with enabling the model of permissionless but compatible innovation that the IETF has been promoting, ultimately enabling the serendipity of new applications that has led to many of those applications we can see in the Internet today.

At this stage, this document does not provide a definite answer nor does it propose or promote specific solutions to the problems here portrayed. Instead, this document aims at stimulating discussion on the emerging needs for addressing, with the possibility to fundamentally re-think the addressing in the Internet beyond the current objectives of IPv6, in order to provide the flexibility to suitably support the many new forms of communication that will emerge. Addressing can be rather flexible and can be of any form that applications may need. There is no limitation on the address to preclude any future applications.

To complement the problem statement in this document, the companion gap analysis document [I-D.jia-intarea-internet-addressing-gap-analysis] deepens the issues identified in Section 4 along key properties of today's Internet addressing.

6. Security Considerations

The present memo does not introduce any new technology and/or mechanism and as such does not introduce any security threat to the TCP/IP protocol suite.

Nevertheless, it is worth to observe whether or not greater flexibility of addressing (as suggested in previous sections) would allow to introduce fully featured security in endpoint identification, potentially able to eradicate the spoofing problem, as one example. Furthermore, it may be used to include application gateways' certificates in order to provide more efficiency, e.g., using web certificates also in the addressing of web services. While increasing security, privacy protection may also be improved.

7. IANA Considerations

This document does not include an IANA request.

8. References

8.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.

8.2. Informative References

[ALOHA] Kuo, F., "The ALOHA System", ACM SIGCOMM Computer Communication Review Vol. 25, pp. 41-44, DOI 10.1145/205447.205451, January 1995, <<https://doi.org/10.1145/205447.205451>>.

[BACnet] "BACnet-A Data Communication Protocol for Building Automation and Control Networks", ANSI/ASHRAE Standard 135-2016, January 2016, <https://www.techstreet.com/ashrae/standards/ashrae-135-2016?product_id=1918140>.

[BIER-MC] Trossen, D., Rahman, A., Wang, C., and T. Eckert, "Applicability of BIER Multicast Overlay for Adaptive Streaming Services", Work in Progress, Internet-Draft, draft-ietf-bier-multicast-http-response-06, 10 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06.txt>>.

- [BLE] "Bluetooth Specification", Bluetooth SIG Working Groups, n.d., <<https://www.bluetooth.com/specifications>>.
- [CARTISEAN] Hughes, L., Shumon, K., and Y. Zhang, "Cartesian Ad Hoc Routing Protocols", Ad-Hoc, Mobile, and Wireless Networks pp. 287-292, DOI 10.1007/978-3-540-39611-6_27, 2003, <https://doi.org/10.1007/978-3-540-39611-6_27>.
- [CCN] Jacobson, V., Smetters, D., Thornton, J., Plass, M., Briggs, N., and R. Braynard, "Networking named content", Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09, DOI 10.1145/1658939.1658941, 2009, <<https://doi.org/10.1145/1658939.1658941>>.
- [CHEN21] Chen, Y., Li, H., Liu, J., Wu, Q., and Z. Lai, "GAMS: An IP Address Management Mechanism in Satellite Mega-constellation Networks", 2021 International Wireless Communications and Mobile Computing (IWCMC), DOI 10.1109/iwcmc51323.2021.9498722, June 2021, <<https://doi.org/10.1109/iwcmc51323.2021.9498722>>.
- [CHRIKI19] Chriki, A., Touati, H., Snoussi, H., and F. Kamoun, "FANET: Communication, mobility models and security issues", Computer Networks Vol. 163, pp. 106877, DOI 10.1016/j.comnet.2019.106877, November 2019, <<https://doi.org/10.1016/j.comnet.2019.106877>>.
- [DDS] AL-Madani, B., Elkhider, S., and S. El-Ferik, "DDS-Based Containment Control of Multiple UAV Systems", Applied Sciences Vol. 10, pp. 4572, DOI 10.3390/app10134572, July 2020, <<https://doi.org/10.3390/app10134572>>.
- [DECT-ULE] "Digital Enhanced Cordless Telecommunications (DECT); Common Interface (CI); Part 1: Overview", ETSI European Standard, EN 300 175-1, V2.6.1, May 2009, <https://www.etsi.org/deliver/etsi_en/300100_300199/30017501/02.06.01_60/en_30017501v020601p.pdf>.
- [DETNET] "Deterministic Networking (DetNet)", n.d., <<https://datatracker.ietf.org/wg/detnet/about/>>.
- [DINRG] "Decentralized Internet Infrastructure - DINRG", n.d., <<https://datatracker.ietf.org/rg/dinrg/about/>>.

[DLT-draft]

Trossen, D., Guzman, D., Bride, M. M., and X. Fan, "Impact of DLTs on Provider Networks", Work in Progress, Internet-Draft, draft-trossen-rtgwg-impact-of-dlts-01, 2 March 2022, <<https://www.ietf.org/archive/id/draft-trossen-rtgwg-impact-of-dlts-01.txt>>.

[ECMA-340] EECMA-340, "Near Field Communication - Interface and Protocol (NFCIP-1) 3rd Ed.", June 2013.

[EIBP] Shenoy, S Chandraiah, P Willis, N., "A Structured Approach to Routing in the Internet", June 2021, <First Intl Workshop on Semantic Addressing and Routing for Future Networks>.

[ETSI-NIN] ETSI - European Telecommunication Standards Institute, "Non-IP Networking - NIN", n.d., <<https://www.etsi.org/technologies/non-ip-networking>>.

[HANDLEY] Handley, M., "Delay is Not an Option: Low Latency Routing in Space", Proceedings of the 17th ACM Workshop on Hot Topics in Networks, DOI 10.1145/3286062.3286075, November 2018, <<https://doi.org/10.1145/3286062.3286075>>.

[I-D.ietf-6man-mtu-option]

Hinden, R. M. and G. Fairhurst, "IPv6 Minimum Path MTU Hop-by-Hop Option", Work in Progress, Internet-Draft, draft-ietf-6man-mtu-option-13, 28 February 2022, <<https://www.ietf.org/archive/id/draft-ietf-6man-mtu-option-13.txt>>.

[I-D.ietf-intarea-gue]

Herbert, T., Yong, L., and O. Zia, "Generic UDP Encapsulation", Work in Progress, Internet-Draft, draft-ietf-intarea-gue-09, 26 October 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-gue-09.txt>>.

[I-D.ietf-lisp-introduction]

Cabellos, A. and D. S. (Ed.), "An Architectural Introduction to the Locator/ID Separation Protocol (LISP)", Work in Progress, Internet-Draft, draft-ietf-lisp-introduction-15, 20 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-lisp-introduction-15.txt>>.

[I-D.ietf-lisp-mn]

Farinacci, D., Lewis, D., Meyer, D., and C. White, "LISP Mobile Node", Work in Progress, Internet-Draft, draft-ietf-lisp-mn-11, 30 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-lisp-mn-11.txt>>.

[I-D.ietf-lisp-nexagon]

Barkai, S., Fernandez-Ruiz, B., Tamir, R., Rodriguez-Natal, A., Maino, F., Cabellos-Aparicio, A., and D. Farinacci, "Network-Hexagons: H3-LISP GeoState & Mobility Network", Work in Progress, Internet-Draft, draft-ietf-lisp-nexagon-19, 14 September 2021, <<https://www.ietf.org/archive/id/draft-ietf-lisp-nexagon-19.txt>>.

[I-D.ietf-lisp-rfc6833bis]

Farinacci, D., Maino, F., Fuller, V., and A. Cabellos, "Locator/ID Separation Protocol (LISP) Control-Plane", Work in Progress, Internet-Draft, draft-ietf-lisp-rfc6833bis-30, 18 November 2020, <<https://www.ietf.org/archive/id/draft-ietf-lisp-rfc6833bis-30.txt>>.

[I-D.jia-intarea-internet-addressing-gap-analysis]

Jia, Y., Trossen, D., Iannone, L., Shenoy, N., and P. Mendes, "Gap Analysis in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-internet-addressing-gap-analysis-01, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-internet-addressing-gap-analysis-01.txt>>.

[I-D.templin-6man-aero]

Templin, F. L., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero-39, 22 February 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-aero-39.txt>>.

[ICN5G]

Ravindran, R., Suthar, P., Trossen, D., Wang, C., and G. White, "Enabling ICN in 3GPP's 5G NextGen Core Architecture", Work in Progress, Internet-Draft, draft-irtf-icnrg-5gc-icn-04, 10 January 2021, <<https://www.ietf.org/archive/id/draft-irtf-icnrg-5gc-icn-04.txt>>.

- [ICNIP] Trossen, D., Robitzsch, S., Reed, M., Al-Naday, M., and J. Riihijarvi, "Internet Services over ICN in 5G LAN Environments", Work in Progress, Internet-Draft, draft-trossen-icnrg-internet-icn-5gln-04, 1 October 2020, <<https://www.ietf.org/archive/id/draft-trossen-icnrg-internet-icn-5gln-04.txt>>.
- [IEEE_1901.1] "Standard for Medium Frequency (less than 15 MHz) Power Line Communications for Smart Grid Applications", IEEE 1901.1 IEEE-SA Standards Board, May 2018, <<https://ieeexplore.ieee.org/document/8360785>>.
- [LR-WPAN] "IEEE 802.15.4 - IEEE Standard for Low-Rate Wireless Networks", IEEE 802.15 WPAN Task Group 4, May 2020, <https://standards.ieee.org/standard/802_15_4-2020.html>.
- [MANET1] Abdallah, A., Abdallah, E., Bsoul, M., and A. Ootom, "Randomized geographic-based routing with nearly guaranteed delivery for three-dimensional ad hoc network", International Journal of Distributed Sensor Networks Vol. 12, pp. 155014771667125, DOI 10.1177/1550147716671255, October 2016, <<https://doi.org/10.1177/1550147716671255>>.
- [MAROJEVIC20] Marojevic, V., Guvenc, I., Dutta, R., Sichitiu, M., and B. Floyd, "Advanced Wireless for Unmanned Aerial Systems: 5G Standardization, Research Challenges, and AERPAAW Architecture", IEEE Vehicular Technology Magazine Vol. 15, pp. 22-30, DOI 10.1109/mvt.2020.2979494, June 2020, <<https://doi.org/10.1109/mvt.2020.2979494>>.
- [OCADO] "Ocado Technologys robot warehouse a Hive of IoT innovation", n.d., <<https://techmonitor.ai/tech-leaders/ocado-technology-robot-hive-innovation>>.
- [PANRG] "Path Aware Networking Research Group - PANRG", n.d., <<https://datatracker.ietf.org/rg/panrg/about/>>.
- [PEARG] "Privacy Enhancements and Assessments Research Group - PEARG", n.d., <<https://irtf.org/pearg>>.
- [PILA] Krahenbuhl, C., Legner, M., Bitterli, S., and A. Perrig, "Pervasive Internet-Wide Low-Latency Authentication", 2021 International Conference on Computer Communications and Networks (ICCCN), DOI 10.1109/icccn52240.2021.9522235, July 2021, <<https://doi.org/10.1109/icccn52240.2021.9522235>>.

- [RFC2775] Carpenter, B., "Internet Transparency", RFC 2775, DOI 10.17487/RFC2775, February 2000, <<https://www.rfc-editor.org/info/rfc2775>>.
- [RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, DOI 10.17487/RFC3031, January 2001, <<https://www.rfc-editor.org/info/rfc3031>>.
- [RFC3972] Aura, T., "Cryptographically Generated Addresses (CGA)", RFC 3972, DOI 10.17487/RFC3972, March 2005, <<https://www.rfc-editor.org/info/rfc3972>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4919] Kushalnagar, N., Montenegro, G., and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", RFC 4919, DOI 10.17487/RFC4919, August 2007, <<https://www.rfc-editor.org/info/rfc4919>>.
- [RFC5061] Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M. Kozuka, "Stream Control Transmission Protocol (SCTP) Dynamic Address Reconfiguration", RFC 5061, DOI 10.17487/RFC5061, September 2007, <<https://www.rfc-editor.org/info/rfc5061>>.
- [RFC5177] Leung, K., Dommety, G., Narayanan, V., and A. Petrescu, "Network Mobility (NEMO) Extensions for Mobile IPv4", RFC 5177, DOI 10.17487/RFC5177, April 2008, <<https://www.rfc-editor.org/info/rfc5177>>.
- [RFC5275] Turner, S., "CMS Symmetric Key Management and Distribution", RFC 5275, DOI 10.17487/RFC5275, June 2008, <<https://www.rfc-editor.org/info/rfc5275>>.
- [RFC5517] HomChaudhuri, S. and M. Foschiano, "Cisco Systems' Private VLANs: Scalable Security in a Multi-Client Environment", RFC 5517, DOI 10.17487/RFC5517, February 2010, <<https://www.rfc-editor.org/info/rfc5517>>.
- [RFC5944] Perkins, C., Ed., "IP Mobility Support for IPv4, Revised", RFC 5944, DOI 10.17487/RFC5944, November 2010, <<https://www.rfc-editor.org/info/rfc5944>>.

- [RFC6158] DeKok, A., Ed. and G. Weber, "RADIUS Design Guidelines", BCP 158, RFC 6158, DOI 10.17487/RFC6158, March 2011, <<https://www.rfc-editor.org/info/rfc6158>>.
- [RFC6182] Ford, A., Raiciu, C., Handley, M., Barre, S., and J. Iyengar, "Architectural Guidelines for Multipath TCP Development", RFC 6182, DOI 10.17487/RFC6182, March 2011, <<https://www.rfc-editor.org/info/rfc6182>>.
- [RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, DOI 10.17487/RFC6275, July 2011, <<https://www.rfc-editor.org/info/rfc6275>>.
- [RFC6626] Tsirtsis, G., Park, V., Narayanan, V., and K. Leung, "Dynamic Prefix Allocation for Network Mobility for Mobile IPv4 (NEMOv4)", RFC 6626, DOI 10.17487/RFC6626, May 2012, <<https://www.rfc-editor.org/info/rfc6626>>.
- [RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <<https://www.rfc-editor.org/info/rfc7348>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <<https://www.rfc-editor.org/info/rfc7426>>.
- [RFC7429] Liu, D., Ed., Zuniga, JC., Ed., Seite, P., Chan, H., and CJ. Bernardos, "Distributed Mobility Management: Current Practices and Gap Analysis", RFC 7429, DOI 10.17487/RFC7429, January 2015, <<https://www.rfc-editor.org/info/rfc7429>>.
- [RFC7476] Pentikousis, K., Ed., Ohlman, B., Corujo, D., Boggia, G., Tyson, G., Davies, E., Molinaro, A., and S. Eum, "Information-Centric Networking: Baseline Scenarios", RFC 7476, DOI 10.17487/RFC7476, March 2015, <<https://www.rfc-editor.org/info/rfc7476>>.

- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<https://www.rfc-editor.org/info/rfc7665>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8280] ten Oever, N. and C. Cath, "Research into Human Rights Protocol Considerations", RFC 8280, DOI 10.17487/RFC8280, October 2017, <<https://www.rfc-editor.org/info/rfc8280>>.
- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8595] Farrel, A., Bryant, S., and J. Drake, "An MPLS-Based Forwarding Plane for Service Function Chaining", RFC 8595, DOI 10.17487/RFC8595, June 2019, <<https://www.rfc-editor.org/info/rfc8595>>.
- [RFC8677] Trossen, D., Purkayastha, D., and A. Rahman, "Name-Based Service Function Forwarder (nSFF) Component within a Service Function Chaining (SFC) Framework", RFC 8677, DOI 10.17487/RFC8677, November 2019, <<https://www.rfc-editor.org/info/rfc8677>>.
- [RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <<https://www.rfc-editor.org/info/rfc8754>>.
- [RFC8763] Rahman, A., Trossen, D., Kutscher, D., and R. Ravindran, "Deployment Considerations for Information-Centric Networking (ICN)", RFC 8763, DOI 10.17487/RFC8763, April 2020, <<https://www.rfc-editor.org/info/rfc8763>>.
- [RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [RFC8926] Gross, J., Ed., Ganga, I., Ed., and T. Sridhar, Ed., "Geneve: Generic Network Virtualization Encapsulation", RFC 8926, DOI 10.17487/RFC8926, November 2020, <<https://www.rfc-editor.org/info/rfc8926>>.

- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [SFCANYCAST] Wion, A., Bouet, M., Iannone, L., and V. Conan, "Distributed Function Chaining with Anycast Routing", Proceedings of the 2019 ACM Symposium on SDN Research, DOI 10.1145/3314148.3314355, April 2019, <<https://doi.org/10.1145/3314148.3314355>>.
- [TERASTREAM] "Deutsche Telekom tests TeraStream, the network of the future, in Croatia", n.d., <<https://www.telekom.com/en/media/media-information/archive/deutsche-telekom-tests-terastream-the-network-of-the-future-in-croatia-358444>>.
- [TROSSEN] Trossen, D., Sarela, M., and K. Sollins, "Arguments for an information-centric internetworking architecture", ACM SIGCOMM Computer Communication Review Vol. 40, pp. 26-33, DOI 10.1145/1764873.1764878, April 2010, <<https://doi.org/10.1145/1764873.1764878>>.
- [WANG19] Wang, P., Zhang, J., Zhang, X., Yan, Z., Evans, B., and W. Wang, "Convergence of Satellite and Terrestrial Networks: A Comprehensive Survey", IEEE Access Vol. 8, pp. 5550-5588, DOI 10.1109/access.2019.2963223, 2020, <<https://doi.org/10.1109/access.2019.2963223>>.

Acknowledgments

Thanks to all the people that shared insightful comments both privately to the authors as well as on various mailing list, especially on the INTArea Mailing List. Also thanks for the interesting discussions to Stewart Bryant, Ron Bonica, Toerless Eckert, Brian E. Carpenter, Kiran Makhijani, Fred Templin.

Authors' Addresses

Yihao Jia
Huawei Technologies Co., Ltd
156 Beiqing Rd.
Beijing
100095
P.R. China
Email: jiayihao@huawei.com

Dirk Trossen
Huawei Technologies Duesseldorf GmbH
Riesstr. 25C
80992 Munich
Germany
Email: dirk.trossen@huawei.com

Luigi Iannone
Huawei Technologies France S.A.S.U.
18, Quai du Point du Jour
92100 Boulogne-Billancourt
France
Email: luigi.iannone@huawei.com

Nirmala Shenoy
Rochester Institute of Technology
New-York, 14623
United States of America
Email: nxsvks@rit.edu

Paulo Mendes
Airbus
Willy-Messerschmitt Strasse 1
81663 Munich
Germany
Email: paulo.mendes@airbus.com

Donald E. Eastlake 3rd
Futurewei Technologies
2386 Panoramic Circle
Apopka, FL, 32703
United States of America
Email: d3e3e3@gmail.com

Peng Liu
China Mobile
32 Xuanwumen West Ave
Xicheng, Beijing
100053
P.R. China
Email: liupengygy@chinamobile.com

Dino Farinacci
lispers.net
United States of America
Email: farinacci@gmail.com

Independent Submission
Internet-Draft
Intended status: Informational
Expires: December 12, 2021

K. Makhijani
L. Dong
Futurewei
June 10, 2021

Requirements and Scenarios for Industry Internet Addressing
draft-km-industrial-internet-requirements-00

Abstract

Industry Control Networks host a diverse set of non-internet protocols for different purposes. Even though they operate in a controlled environment, one end of industrial control applications run over internet technologies (IT) and another over operational technology (OT) protocols. This memo discusses the challenges and requirements relating to convergence of OT and IT networks. One particular problem in convergence is figuring out reachability between these networks.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://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 12, 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
2. Terminology	3
2.1. Acronymns	4
3. Industrial Network Reference Architecture	4
3.1. Communication Patterns	5
3.2. Industry Control Network Nuances (current state)	5
4. Problem Statement	6
4.1. Heterogenity	7
4.2. Automation Impact	7
4.2.1. Scale	8
4.2.2. Stretch Control Fabric to Edge and Cloud	8
4.2.3. Reliability	8
4.2.4. Resilience	8
4.3. OT/IT Convergence	8
4.4. Data oriented networking	9
4.5. Virtualization	9
5. Address Space Requirements	9
5.1. Short Device Addressing	9
5.2. Meaningful Addresses	10
5.3. Device name based Addresses	10
5.4. Adoption of Lean Network Layer	10
5.5. Multi-semantic behavior	10
5.6. Interoperability with IP-world machines	11
6. Relationship with Activities in IETF	11
6.1. Deterministic Networks (DetNet WG)	11
6.2. IoT OPS	11
6.3. LPWAN	11
6.4. Recent Addressing related work	12
7. IANA Considerations	12
8. Security Considerations	12
9. Acknowledgements	12
10. Informative References	12
Authors' Addresses	13

1. Introduction

An industry control network interconnects devices used to operate, control and monitor physical equipment in industrial environments. These networks are increasingly becoming complex as the emphasis on convergence of OT/IT grows to improve the automation. On one side of Industrial internet are the inventory management, supply chain and simulation software and the other side are the control devices

operating on machines. Operational Technologies (OT) networks are more often tied to set of non-internet protocols such as Modbus, Profibus, CANbus, Profinet, etc. There are more than 100 different protocols each with it's own packet format and are used in the industry.

It is expected that integration between the IT and OT will provide numerous benefits in terms of improved productivity, efficiency of operations by providing end to end visibility and control. Industry control applications also expect to operate at cloud scale by virtualization of several modules (especially PLCs) leading to new set of network requirements.

One aspect of industry control is the delivery of data associated with the Real-time, deterministic and reliability characteristics over local-area and wide-area networks. This type of inter-operability functionality and study is already covered in DETNET working group. The other aspect is reachability and interconnection keeping heterogeneity of communication interfaces and a variety of services in mind. This document focuses on the latter part only.

OT networks have been traditionally separate from the IT networks. It allowed OT network experts to manage and control processes without much dependency on changes in the external networks. This is an important to consideration when dealing with the industry control networks to maintain them in a controlled environment leveraging the limited-domain networks [LDN] concept for an independent network control.

The purpose of this document is to discuss the reachability and interconnection characteristics, challenges and new requirements emerging from large-scale integration of IT and OT.

2. Terminology

- o Industrial Control Networks: The industrial control networks are interconnection of equipments used for the operation, control or monitoring of machines in the industry environment. It involves different level of communications - between fieldbus devices, digital controllers and software applications
- o Industry Automation: Mechanisms that enable machine to machine communication by use of technologies that enable automatic control and operation of industrial devices and processes leading to minimizing human intervention.
- o Human Machine Interface: An interface between the operator and the machine. The communication interface relays I/O data back and

forth between an operator's terminal and HMI software to control and monitor equipment.

2.1. Acronyms

- o HMI: Human Machine Interface

3. Industrial Network Reference Architecture

In the scope of this document the following reference industrial network will be used to provide structure to the discussion. In the Fig. Figure 1 below, a hierarchy of communications is shown. At the lowest level, PLCs operate and control field devices; above that Human Machine Interface (HMI) interconnects with different PLCs to program and control underlying field devices. HMI itself, sends data up to applications for consumption in that industry vertical.

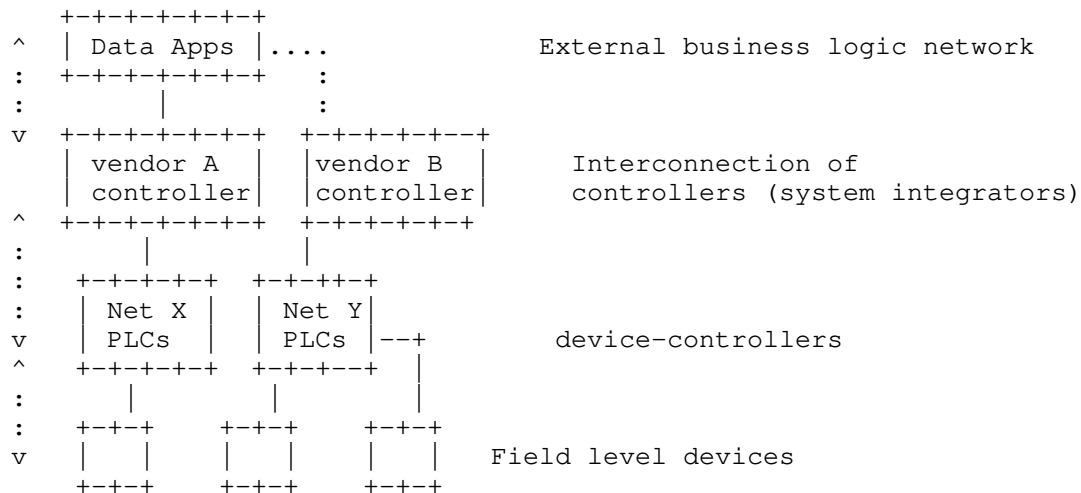


Figure 1: Hierarchy of Functions Industrial Control Networks

Unlike commercial networks that uniformly run IP protocols, the communication links run different protocols at along the different level of the hierarchy. One of the key requirement from new industrial applications is the integration of different types of communication protocols including Modbus, Profinet, Profibus, ControlNet, CANOpen etc.

A vertically integration system involves a network between the external business applications and higher controllers (for e.g. SCADA, HMI, or system integrators) is IP based. The second level of networks between the controllers can be either IP or non-IP

(Profibus, BACNet, etc.). The lowest field-level networks between industrial controllers and field-level may be any of the fieldbus or device control protocols (More details of the industry networks can be found in [SURV]).

3.1. Communication Patterns

The following communication patterns are commonly observed:

- o controller to controller: A communication between multi-vendor controller maybe required by system integrators to work in complex systems.
- o controller to field level devices: This is a fieldbus communication between device such as I/O modules, motors, controllers. This communication represent.
- o Device to device: allows direct communication between wired industrial devices and wireless devices to enhance automation use cases. For an exmaple, use of camera to visually monitor and detect anamolies in other devices.
- o controller to compute: vertical communication between a controller and compute integrates IP-based technologies with non-IP since OT product systems and solutions are not connected with IP based networks.

A certain level of inter-operability is required to exchange data between the above endpoints from different vendors. One of the challange is that Ethernet (which unifies IT standards) that's not always possible in Industry networks.

3.2. Industry Control Network Nuances (current state)

The Industry control networks are engineered for the idustry verticals they belong to and depict unique properties as below:

- o location bound: The Control Device's location or the network they are attached to is predetermined and changes rarely. However, the network resources may not get efficiently utilized to avoid contention between them.
- o security by separation: Typically, security is enhanced by keeping IT/OTnetworks separate. The operators control how data goes in and out of a site through firewalls and policies.
- o data growth: Even though the size of network remains the same, data generated is much higher. For example, cameras installed for

visual inspection to determine the quality of manufactured product generates a high bandwidth demand.

- o Wired device constraints: A bulk of machines are over wired network, their constraints vary from LPWAN and IoT devices which is an active area of standardization work. device lifetime, or power-requirements are not typical constraints. Instead direct process control mechanisms are more important.
- o Real-time behavior: The control devices require realtime as well as deterministic behavior between a controller (such as an HMI station) to the equipment. The DetNet working group covers several aspects.

The goal of the document is not to reinvent the Industry control infrastructure. See section Section 6 on related standards work. It is meant to exclude the already covered by other WGs.

Since a device connects to network through its address, the document explores different address specific nuances in control devices - such as management, device discovery and integration requirements. This document concerns with the identification of and role networks, specifically from the organization of industry control devices.

The goal of this document is to outline some of the challenges and improvement of connectivity aspects of Industry control networks.

4. Problem Statement

In industrial networks, a good number of devices still communicate over a serial or field bus (although Ethernet is being gradually adopted). The operations on these devices are performed by writing provide direct access to operation-control. i.e what operation to perform is embedded in the type of interface itself. For instance, Profibus, Modbus networks are implicitly latency sensitive and short control-command based.

ModBus

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| address | Function  code  | data |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

CANBus

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
| message id | data |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

Profibus - todo.

Since they are localized in an area such as factory floor or a site, such networks have evolved independently and are separated from the IT applications. The emerging trend requires a seamless integration with intelligent software, sophisticated compute platforms and other operational aspects as highlighted below:

4.1. Heterogeneity

A typical industry control network has devices of different communication interfaces such as Fieldbus (PROFIBUS, Modbus, and HART), Ethernet (generic Ethernet/IP, PROFINET, and Modbus-TCP), and also wireless (Bluetooth, Wireless HART, and IoT). These interfaces vary at the physical and link layers and because they integrate with their own application technologies providing interoperability between these devices remains a challenge. This also makes difficult to adopt to modern integration technologies.

Fieldbus client-server architecture is widely deployed. It delivers commands deterministically from a controller to the device and vice-versa. Interfaces of this kind have typically shorter addresses (upto 256 devices on a single bus in Modbus).

Some of the servers also behave as protocol gateways and interconnect different type of protocols. For example when a modbus device is being controlled by a profinet server, an gateway function will translate modbus data or encapsulate it over IP (if the controller supports it).

In a Gateway-centric approach, gateways are in charge of protocol translations between the devices with different interfaces. This requires packing and unpacking of data in the source and destination formats at the attached gateways. Note: As an example, a Modbus device does not know whether to send command to Profibus PLC or Modbus PLC. The gateway device attaches to performs the translation. This is even worse with encapsulations, where the entire frame is carried over IP.

This is not ideal for latency sensitive applications. Although hardware wise, gateways need to be equipped with all the interface, it is more efficient to only perform data link conversion.

4.2. Automation Impact

Automation of processes in industry relies on control sophisticated technologies such as machine learning, big data, etc. with minimal human intervention. Automation needs to support scale, reliability and resilience at large-scale.

4.2.1. Scale

Automation control at small scale applications with well defined task has been possible. In order to improve production, and eliminate stoppages and minimizing human intervention.

When the number or density of devices, and processes increase there is a need to schedule, route, and coordinate over multiple control environments.

4.2.2. Stretch Control Fabric to Edge and Cloud

The industry control networks can be extended to cloud or edge compute platforms. Since these networks are not equipped with compute intensive servers. Now extending the communication to the edge and cloud nodes increases the distance requiring traditional L2 networks to be adopted to L3 network designs.

Design decisions will require to choose different transit strategies (this maybe layer 1, 2, 3 technologies or even network slices). It also influence the security policies.

4.2.3. Reliability

Production efficiency is inversely related to number of defects in a process. System reliability is determined through measurements of its instantaneous state.

Automation processes need to ensure that system is performing in an expected state and is capable of reporting anomalies fast and accurately (i.e. packet drops or jitter leading to poor quality product).

4.2.4. Resilience

TBD.

4.3. OT/IT Convergence

Most of the factory floors are not equipped with IT servers to perform compute intensive tasks. Yet an IP-based device need to connect with non-IP interface to control those devices.

Often real-time response is necessary for example, in closed-loop control systems direct communication is desired to avoid any additional packet processing delay or overheads at the source and destination gateways, equipping IP to all OT devices and abandoning

the existing investment and depolyment could result in the following obvious problems.

- o Many of the standard IP based protocols maybe too much overhead for OT devices.
- o Cannot preserve communication characteristics of devices (different device addressing scheme, realtime, IRT, message identifiers, Bus-like properties).
- o It relies heavily on hierarchy network stack (network layer, transport layer, application), where as OT devices do not have any, they generally operate at data link layer or below.

4.4. Data oriented networking

Industry verticals keep data and control on the manufacturing floor, on a closed system. There is no easy way to forward this data to enterprise level software. On premise micro data centers or edge computing are new infrastructure pieces that will impact the design of current industrial networks.

4.5. Virtualization

Traditional Industry control infrastructure is not virtualized. Virtualization will enable deployment of new functionality in a flexible manner.

- o Virtual PLCs are considered an important component functionality customization of digital-twin realization.
- o virtualization enables edge and cloud native computing by moving and instantiating workflows at different locations.

Implications that PLCs are no longer one-hop away.

5. Address Space Requirements

5.1. Short Device Addressing

Shorter addresses are inherent to industry control systems to provide implicit determinism.

Note: The motivation for short address is to preseve the legacy attributes of fieldbus control devices. It is not related low-power or resource constraints.

A large volume of the messages are of sizes shorter than the size of IP headers (v4, v6) themselves. The header tax will be very high over industry control networks.

5.2. Meaningful Addresses

The industry control floors are built bottom-up. The devices are carefully wired and connected to controllers. In a hierarchical network design, a particular type of machine can be reached in a structured manner by adding subnet or location to the address structures.

5.3. Device name based Addresses

HMI might require human readable address that is understandable to human operators or application end users. For example, a device address could be associated with its location, type of applications, attached objects etc. The network needs to support the resolution and routing based on such device addresses, which is more user friendly. On the other hand, grouping devices based on their addresses shall be easily implemented to enable group operation and communication.

5.4. Adoption of Lean Network Layer

Challenge of Industrial network device address is that it communicates to a physical device address. Traditionally, in a limited environment there was no need for network layer or expressing network specific service, access control.

- o If a sensor is broken, it will require reprogramming of controller and re-aligning with the new address. The benefit of network layer, removes this restriction.
- o Note that, using IP stack is not suitable because these devices perform specific functions and any overhead in transport or large addressing can add to processing delays.
- o Several other IP suite protocols such as device discovery should be revisited.

5.5. Multi-semantic behavior

OT networks, at least at site level are organized at much smaller scale than typical IP-capable networks. There is in turn a fixed hierarchy of networks w.r.t. location in a plant.

5.6. Interoperability with IP-world machines

To develop further on different type of address format support. From smaller address of legacy devices to IT based applications with IP address.

(OT-Address)--->(Industry Control)--->(IP-Address)
(control dev) (network) (application)

Preferably allow OT devices to understand IP-addresses for the servers they connect to.

6. Relationship with Activities in IETF

6.1. Deterministic Networks (DetNet WG)

The Deterministic Networking (DetNet) [DETNET-ARCH] is working on using IP for long-range connectivity with bounded latency in industry control networks . Its data plane [DETNET-DP] takes care of forwarding aspects and most close to Industry control networks but the focus is on the controlled latency, low packet loss & delay variation, and high reliability functions. Not dealing with interconnection of devices.

In layer 2 domain, similar functionality is covered by TSN Ethernet [IEEE802.1TSNTG].

6.2. IoT OPS

IoT operations group discusses device security, privacy, and bootstrapping and device onboarding concepts. Among the device provisioning one of the object is network identifier. We understand that the IoT OPs does not exclude evaluation of industry IoT or control devices requirements. Given the specific functions described above it maybe necessary to configure more than an identifier, i.e. server or controller information or specific address scope and structure.

6.3. LPWAN

The LPWAN has focussed on low-power and constrained devices. There are compression related approaches that may apply are [SCHC] or [ROHC]. To be evaluated for process control devices.

6.4. Recent Addressing related work

Some of the work initiated on the addressing include solutions such as [FlexIP], [Flexible_IP], [FHE], and [SOIP].

Recently, a broader area of problem statement and challenges in [CHALLENGE].

7. IANA Considerations

This document requires no actions from IANA.

8. Security Considerations

This document introduces no new security issues.

9. Acknowledgements

10. Informative References

[CHALLENGE] Jia, Y., Trossen, D., Iannone, L., 3rd, D. E. E., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", draft-jia-intarea-scenarios-problems-addressing-00 (work in progress), February 2021.

[DETNET-ARCH]
Finn, N., Thubert, P., Varga, B., and J. Farkas,
"Deterministic Networking Architecture", RFC 8655,
DOI 10.17487/RFC8655, October 2019,
<<https://www.rfc-editor.org/info/rfc8655>>.

[DETNET-DP]
Varga, B., Ed., Farkas, J., Berger, L., Fedyk, D., and S. Bryant, "Deterministic Networking (DetNet) Data Plane: IP", RFC 8939, DOI 10.17487/RFC8939, November 2020,
<<https://www.rfc-editor.org/info/rfc8939>>.

[FHE] Jiang, S., Li, G., and B. Carpenter, "Asymmetric IPv6 for Resource-constrained IoT Networks", draft-jiang-asymmetric-ipv6-04 (work in progress), November 2020.

[Flexible_IP]
Jia, Y., Chen, Z., and S. Jiang, "Flexible IP: An Adaptable IP Address Structure", draft-jia-flex-ip-address-structure-00 (work in progress), October 2020.

- [FlexIP] Moskowitz, R., Li, G., and S. Ren, "FlexIP Addressing", draft-moskowitz-flexip-addressing-00 (work in progress), January 2019.
- [IEEE802.1TSNTG] "IEEE, "Time-Sensitive Networking (TSN) Task Group", 2018, <<https://1.ieee802.org/tsn>>.
- [LDN] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <<https://www.rfc-editor.org/info/rfc8799>>.
- [ROHC] Jonsson, L-E., Pelletier, G., and K. Sandlund, "The RObust Header Compression (ROHC) Framework", RFC 4995, DOI 10.17487/RFC4995, July 2007, <<https://www.rfc-editor.org/info/rfc4995>>.
- [SCHC] Minaburo, A., Toutain, L., Gomez, C., Barthel, D., and JC. Zuniga, "SCHC: Generic Framework for Static Context Header Compression and Fragmentation", RFC 8724, DOI 10.17487/RFC8724, April 2020, <<https://www.rfc-editor.org/info/rfc8724>>.
- [SOIP] Carpenter, B., Jiang, S., and G. Li, "Service Oriented Internet Protocol", draft-jiang-service-oriented-ip-03 (work in progress), May 2020.
- [SURV] Galloway, B. and G. Hancke, "Introduction to Industrial Control Networks", IEEE Communications Surveys & Tutorials Vol. 15, pp. 860-880, DOI 10.1109/surv.2012.071812.00124, 2013.

Authors' Addresses

Kiran Makhijani
Futurewei

Email: kiran.ietf@gmail.com

Lijun Dong
Futurewei
Central Expy
Santa Clara, CA 95050
United States of America

Email: lijun.dong@futurewei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 17 August 2022

L. Han, Ed.
R. Li
A. Retana
Futurewei Technologies, Inc.
M. Chen
L. Su
China Mobile
N. Wang
University of Surrey
13 February 2022

Problems and Requirements of Satellite Constellation for Internet
draft-lhan-problems-requirements-satellite-net-02

Abstract

This document presents the detailed analysis about the problems and requirements of satellite constellation used for Internet. It starts from the satellite orbit basics, coverage calculation, then it estimates the time constraints for the communications between satellite and ground-station, also between satellites. How to use satellite constellation for Internet is discussed in detail including the satellite relay and satellite networking. The problems and requirements of using traditional network technology for satellite network integrating with Internet are finally outlined.

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 <https://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 17 August 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology	3
3. Overview	5
4. Basics of Satellite Constellation	6
4.1. Satellite Orbit	6
4.2. Coverage of LEO and VLEO Satellites and Minimum Number Required	6
4.3. Real Deployment of LEO and VLEO for Satellite Network . .	9
5. Communications for Satellite Constellation	10
5.1. Dynamic Ground-station-Satellite Communication	11
5.2. Dynamic Inter-satellite Communication	12
5.2.1. Inter-satellite Communication Overview	12
5.2.2. Satellites on Adjacent Orbit Planes with Same Altitude	15
5.2.3. Satellites on Adjacent Orbit Planes with Different Altitude	17
6. Use Satellite Network for Internet	19
7. Problems and Requirements for Satellite Constellation for Internet	22
7.1. Common Problems and Requirements	22
7.2. Satellite Relay	25
7.2.1. One Satellite Relay	25
7.2.2. Multiple Satellite Relay	26
7.3. Satellite Networking	28
7.3.1. L2 or L3 network	28
7.3.2. Inter-satellite-Link Lifetime	28
7.3.3. Problems for Traditional Routing Technologies	29
8. IANA Considerations	33
9. Contributors	33
10. Acknowledgements	33
11. References	33
11.1. Normative References	33
11.2. Informative References	34
Appendix A. Change Log	36
Authors' Addresses	36

1. Introduction

Satellite constellation for Internet is emerging. Even there is no constellation network established completely yet at the time of the publishing of the draft (June 2021), some basic internet service has been provided and has demonstrated competitive quality to traditional broadband service.

This memo will analyze the challenges for satellite network used in Internet by traditional routing and switching technologies. It is based on the analysis of the dynamic characters of both ground-station-to-satellite and inter-satellite communications and its impact to satellite constellation networking.

The memo also provides visions for the future solution, such as in routing and forwarding.

The memo focuses on the topics about how the satellite network can work with Internet. It does not focus on physical layer technologies (wireless, spectrum, laser, mobility, etc.) for satellite communication.

2. Terminology

LEO	Low Earth Orbit with the altitude from 180 km to 2000 km.
VLEO	Very Low Earth Orbit with the altitude below 450 km
MEO	Medium Earth Orbit with the altitude from 2000 km to 35786 km
GEO	Geosynchronous orbit with the altitude 35786 km
GSO	Geosynchronous satellite on GEO
ISL	Inter Satellite Link
ISLL	Inter Satellite Laser Link
EIRP	Effective isotropic radiated power
P2MP	Point to Multiple Points
GS	Ground Station, a device on ground connecting the

satellite. In the document, GS will hypothetically provide L2 and/or L3 functionality in addition to process/send/receive radio wave. It might be different as the reality that the device to process/send/receive radio wave and the device to provide L2 and/or L3 functionality could be separated.

SGS	Source ground station. For a specified flow, a ground station that will send data to a satellite through its uplink.
DGS	Destination ground station. For a specified flow, a ground station that is connected to a local network or Internet, it will receive data from a satellite through its downlink and then forward to a local network or Internet.
PGW	Packet Gateway
UPF	User Packet Function
PE router	Provider Edge router
CE router	Customer Edge router
P router	Provider router
LSA	Link-state advertisement
LSP	Link-State PDUs
L1	Layer 1, or Physical Layer in OSI model [OSI-Model]
L2	Layer 2, or Data Link Layer in OSI model [OSI-Model]
L3	Layer 3, or Network Layer in OSI model [OSI-Model], it is also called IP layer in TCP/IP model
BGP	Border Gateway Protocol [RFC4271]
eBGP	External Border Gateway Protocol, two BGP peers have different Autonomous Number
iBGP	Internal Border Gateway Protocol, two BGP peers have same Autonomous Number

IGP Interior gateway protocol, examples of IGPs include Open Shortest Path First (OSPF [RFC2328]), Routing Information Protocol (RIP [RFC2453]), Intermediate System to Intermediate System (IS-IS [RFC7142]) and Enhanced Interior Gateway Routing Protocol (EIGRP [RFC7868]).

3. Overview

The traditional satellite communication system is composed of few GSO and ground stations. For this system, each GSO can cover 42% Earth's surface [GEO-Coverage], so as few as three GSO can provide the global coverage theoretically. With so huge coverage, GSO only needs to amplify signals received from uplink of one ground station and relay to the downlink of another ground station. There is no inter-satellite communications needed. Also, since the GSO is stationary to the ground station, there is no mobility issue involved.

Recently, more and more LEO and VLEO satellites have been launched, they attract attentions due to their advantages over GSO and MEO in terms of higher bandwidth, lower cost in satellite, launching, ground station, etc. Some organizations [ITU-6G][Surrey-6G][Nttddocomo-6G] have proposed the non-terrestrial network using LEO, VLEO as important parts for 6G to extend the coverage of Internet. SpaceX has started to build the satellite constellation called StarLink that will deploy over 10 thousand LEO and VLEO satellites finally [StarLink]. China also started to request the spectrum from ITU to establish a constellation that has 12992 satellites [China-constellation]. European Space Agency (ESA) has proposed "Fiber in the sky" initiative to connect satellites with fiber network on Earth [ESA-HyDRON].

When satellites on MEO, LEO and VLEO are deployed, the communication problem becomes more complicated than for GSO. This is because the altitude of MEO/LEO/VLEO satellites are much lower. As a result, the coverage of each satellite is much smaller than for GSO, and the satellite is not relatively stationary to the ground. This will lead to:

1. More satellites than GSO are needed to provide the global coverage. Section 4.2 will analyze the coverage area, and the minimum number of satellites required to cover the earth surface.
2. The point-to-point communication between satellite and ground station will not be static. Mobility issue has to be considered. Detailed analysis will be done in Section 5.1.

3. The inter-satellite communication is needed, and all satellites need to form a network. details are described in Section 5.2.

In addition to above context, Section 7 will address the problem and requirements when satellite constellation is joining Internet.

As the 1st satellite constellation company in history, the SpaceX/StarLink will be inevitably mentioned in the draft. But it must be noted that all information about SpaceX/StarLink in the draft are from public. Authors of the draft have no relationship or relevant inside knowledge of SpaceX/Starlink.

4. Basics of Satellite Constellation

This section will introduce some basics for satellite such as orbit parameters, coverage estimation, minimum number of satellite and orbit plane required, real deployments.

4.1. Satellite Orbit

The orbit of a satellite can be either circular or elliptic, it can be described by following Keplerian elements [KeplerianElement]:

1. Inclination (i)
2. Longitude of the ascending node (Ω)
3. Eccentricity (e)
4. Semimajor axis (a)
5. Argument of periapsis (ω)
6. True anomaly (ν)

For a circular orbit, two parameters, Inclination and Longitude of the ascending node, will be enough to describe the orbit.

4.2. Coverage of LEO and VLEO Satellites and Minimum Number Required

The coverage of a satellite is determined by many physical factors, such as spectrum, transmitter power, the antenna size, the altitude of satellite, the air condition, the sensitivity of receiver, etc. EIRP could be used to measure the real power distribution for coverage. It is not deterministic due to too many variants in a real environment. The alternative method is to use the minimum elevation angle from user terminals or gateways to a satellite. This is easier and more deterministic. [SpaceX-Non-GEO] has suggested originally

the minimum elevation angle of 35 degrees and deduced the radius of the coverage area is about 435km and 1230km for VLEO (altitude 335.9km) and LEO (altitude 1150km) respectively. The details about how the coverage is calculated from the satellite elevation angle can be found in [Satellite-coverage].

Using this method to estimate the coverage, we can also estimate the minimum number of satellites required to cover the earth surface.

It must be noted, SpaceX has recently reduced the required minimum elevation angle from 35 degrees to 25 degrees. The following analysis still use 35 degrees.

Assume there is multiple orbit planes with the equal angular interval across the earth surface (The Longitude of the ascending node for sequential orbit plane is increasing with a same angular interval). Each orbit plane will have:

1. The same altitude.
2. The same inclination of 90 degree.
3. The same number of satellites.

With such deployment, all orbit planes will meet at north and south pole. The density of satellite is not equal. Satellite is more dense in the space above the polar area than in the space above the equator area. Below estimations are made in the worst covered area, or the area of equator where the satellite density is the minimum.

Figure 1 illustrates the coverage area on equator area, and each satellite will cover one hexagon area. The figure is based on plane geometry instead of spherical geometry for simplification, so, the orbit is parallel approximately.

Figure 2 shows how to calculate the radius (R_c) of coverage area from the satellite altitude (A_s) and the elevation angle (b).

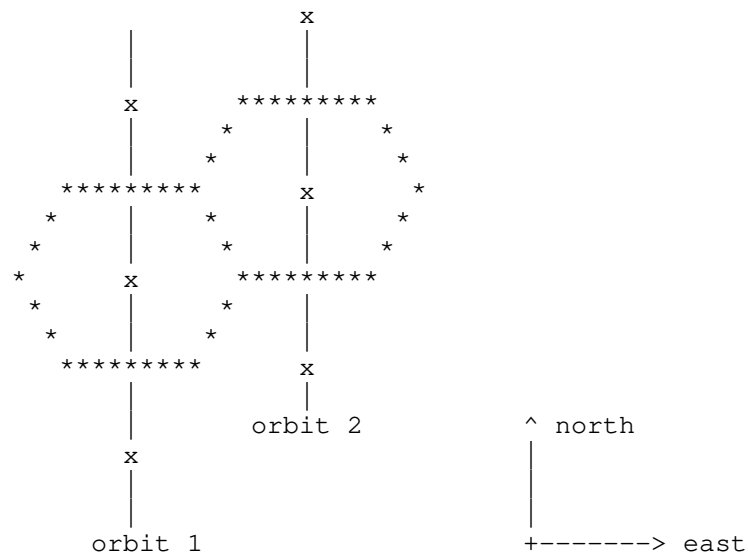


Figure 1: Satellite coverage on ground

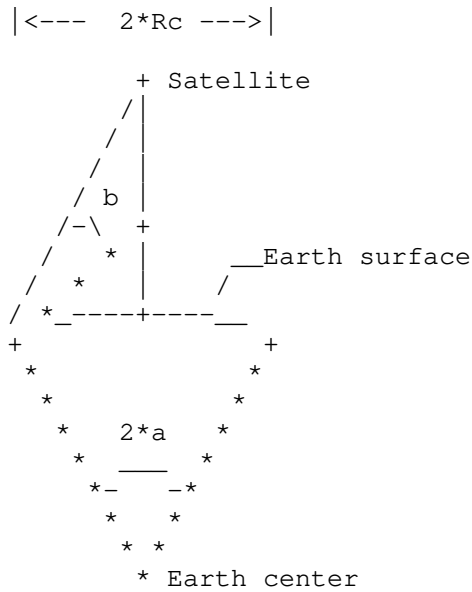


Figure 2: Satellite coverage estimation

- x The vertical projection of satelllite to Earth
- Re The radius of the Earth, Re=6378(km)

As The altitude of a satellite

Rc The radius (arc length) of the coverage, or, the arc length of hexagon center to its 6 vertices. $Rc = Re * (a * \pi) / 180$

a The cap angle for the coverage area (the RC arc). $a = \arccos((Re / (Re + As)) * \cos(b)) - b$.

b The least elevation angle that a ground station or a terminal can communicate with a satellite, $b = 35$ degree.

Ns The minimum number of satellites on one orbit plane, it is equal to the number of the satellite's vertical projection on Earth, so, $Ns = 180 / (a * \cos(30))$

No The minimum number of orbit (with same inclination), it is equal to the number of the satellite orbit's vertical projection, so, $No = 360 / (a * (1 + \sin(30)))$

For a example of two type of satelllite LEO and VEO, the coverages are calculated as in Table 1:

Parameters	VLEO1	VLEO2	LEO1	LEO2
As (km)	335.9	450	1100	1150
a (degree)	3.907	5.078	10.681	11.051
Rc (km)	435	565	1189	1230
Ns	54	41	20	19
No	62	48	23	22

Table 1: Satellite coverage estimation for LEO and VLEO examples

4.3. Real Deployment of LEO and VLEO for Satellite Network

Obviously, the above orbit parameter setup is not optimal since the sky in the polar areas will have the highest density of satellite.

In the real deployment, to provide better coverage for the areas with denser population, to get redundancy and better signal quality, and to make the satellite distance within the range of inter-satellite communication (2000km [Laser-communication-range]), more than the minimum number of satellites are launched. For example, different orbit planes with different inclination/altitude are used.

Normally, all satellites are grouped by orbit planes, each group has a number of orbit planes and each orbit plane has the same orbit parameters, so, each orbit in the same group will have:

1. The same altitude
2. The same inclination, but the inclination is less than 90 degrees. This will result in the empty coverage for polar areas and better coverage in other areas. See the orbit picture for phrase 1 for [StarLink].
3. The same number of satellites
4. The same moving direction for all satellites

The proposed deployment of SpaceX can be seen in [SpaceX-Non-GEO] for StarLink.

The China constellation deployment and orbit parameters can be seen in [China-constellation].

5. Communications for Satellite Constellation

Unlike the communication on ground, the communication for satellite constellation is much more complicated. There are two mobility aspects, one is between ground-station and satellite, another is between satellites.

In the traditional mobility communication system, only terminal is moving, the mobile core network including base station, front haul and back haul are static, thus an anchor point, i.e., PGW in 4G or UPF in 5G, can be selected for the control of mobility session. Unfortunately, when satellite constellation joins the static network system of Internet on ground, there is no such anchor point can be selected since the whole satellite constellation network is moving.

Another special aspect that can impact the communication is that the fast moving speed of satellite will cause frequent changes of communication peers and link states, this will make big challenges to the network side for the packet routing and delivery, session control and management, etc.

5.1. Dynamic Ground-station-Satellite Communication

All satellites are moving and will lead to the communication between ground station and satellite can only last a certain period of time. This will greatly impact the technologies for the satellite networking. Below illustrates the approximate speed and the time for a satellite to pass through its covered area.

In Table 2, VLEO1 and LEO3 have the lowest and highest altitude respectively, VLEO2 is for the highest altitude for VLEO. We can see that longest communication time of ground-station-satellite is less than 400 seconds, the longest communication time for VLEO ground-station-satellite is less than 140 seconds.

The "longest communication time" is for the scenario that the satellite will fly over the receiver ground station exactly above the head, or the ground station will be on the diameter line of satellite coverage circular area, see Figure 1.

Re The radius of the Earth, $Re=6378(km)$

As The altitude of a satellite

AL The arc length(in km) of two neighbor satellite on the same orbit plane, $AL=2*\cos(30)*(Re+As)*(a*pi)/180$

SD The space distance(in km) of two neighbor satellite on the same orbir plane, $SD=2*(Re+As)*\sin(AL/(2*(Re+As)))$.

V the velocity (in m/s) of satellite, $V=\sqrt{G*M/(Re+As)}$

G Gravitational constant, $G=6.674*10^{(-11)}(m^3/(kg*s^2))$

M Mass of Earth, $M=5.965*10^{24}(kg)$

T The time (in second) for a satellite to pass through its cover area, or, the time for the station-satellite communication. $T=ALs/V$

Parameters	VLEO1	VLEO2	LEO1	LEO2	LEO3
As (km)	335.9	450	1100	1150	1325
a (degree)	3.907	5.078	10.681	11.051	12.293
AL (km)	793	1048	2415	2515	2863
SD (km)	792.5	1047.2	2404	2503.2	2846.1
V (km/s)	7.7	7.636	7.296	7.272	7.189
T (s)	103	137	331	346	398

Table 2: The time for the ground-station-satellite communication

5.2. Dynamic Inter-satellite Communication

5.2.1. Inter-satellite Communication Overview

In order to form a network by satellites, there must be an inter-satellite communication. Traditionally, inter-satellite communication uses the microwave technology, but it has following disadvantages:

1. Bandwidth is limited and only up to 600M bps [Microwave-vs-Laser-communication].
2. Security is a concern since the microwave beam is relatively wide and it is easy for 3rd party to sniff or attack.
3. Big antenna size.
4. Power consumption is high.
5. High cost per bps.

Recently, laser is used for the inter-satellite communication, it has following advantages, and will be the future for inter-satellite communication.

1. Higher bandwidth and can be up to 10G bps [Microwave-vs-Laser-communication].

2. Better security since the laser beam size is much narrower than microwave, it is harder for sniffing.
3. The size of optical lens for laser is much smaller than microwave's antenna size.
4. Power saving compared with microwave.
5. Lower cost per bps.

The range for satellite-to-satellite communications has been estimated to be approximately 2,000 km currently [Laser-communication-range].

From Table 2, we can see the Space Distance (SD) for some LEO (altitude over 1100km) are exceeding the ceiling of the range of laser communication, so, the satellite and orbit density for LEO need to be higher than the estimation values in the Table 1.

Assume the laser communication is used for inter-satellite communication, then we can analyze the lifetime of inter-satellite communication when satellites are moving. The Figure 3 illustrates the movement and relative position of satellites on three orbits. The inclination of orbit planes is 90 degrees.

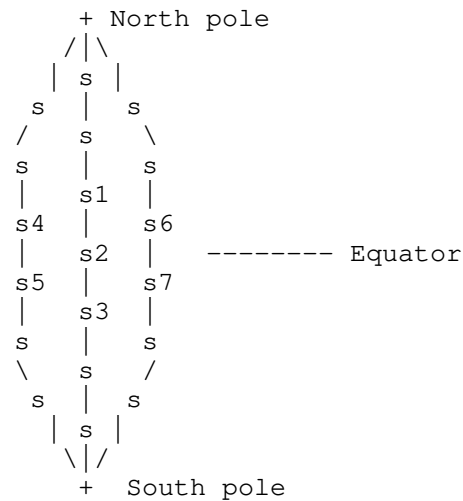
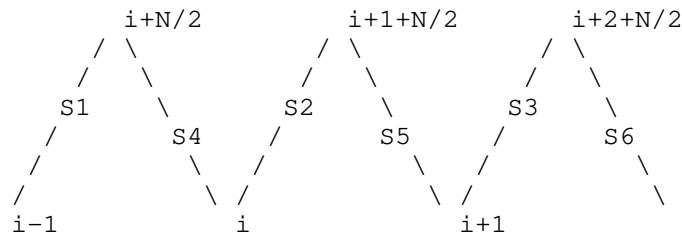


Figure 3: Satellite movement

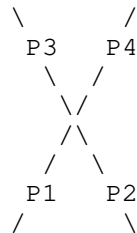
There are four scenarios:

1. For satellites within the same orbit
The satellites in the same orbit will move to the same direction with the same speed, thus the interval between satellites is relatively steady. Each satellite can communicate with its front and back neighbor satellite as long as satellite's orbit is maintained in its life cycle. For example, in Figure 3, s2 can communication with s1 and s3.
2. For satellites between neighbor orbits in the same group at non-polar areas
The orbits for the same group will share the same orbit altitude and inclination. So, the satellite speed in different orbit are also same, but the moving direction may be same or different. Figure 4 illustrates this scenario. When the moving direction is the same, it is similar to the scenario 1, the relative position of satellites in different orbit are relatively steady as long as satellite's orbit is maintained in its life cycle. When the moving direction is different, the relative position of satellites in different orbit are un-steady, this scenario will be analyzed in more details in Section 5.2.2.
3. For satellites between neighbor orbits in the same group at polar areas
For satellites between neighbor orbits with the same speed and moving direction, the relative position is steady as described in #2 above, but the steady position is only valid at areas other than polar area. When satellites meet in the polar area, the relative position will change dramatically. Figure 5 shows two satellites meet in polar area and their ISL facing will be swapped. So, if the range of laser pointing angle is 360 degrees and tracking technology supports, the ISL will not be flipping after passing polar area; Otherwise, the link will be flipping and inter-satellite communication will be interrupted.
4. For satellites between different orbits in the different group
The orbits for the different group will have different orbit altitude, inclination and speed. So, the relative position of satellite is not static. The inter-satellite communication can only last for a while when the distance between two satellite is within the limit of inter-satellite communication, that is 2000km for laser [Laser-communication-range], this scenario will be analyzed in more details in Section 5.2.3



- * The total number of orbit planes are N
- * The number $(i-1, i, i+1, \dots)$ represents the Orbit index
- * The bottom numbers $(i-1, i, i+1)$ are for orbit planes on which satellites $(S1, S2, S3)$ are moving from bottom to up.
- * The top numbers $(i+N/2, i+1+N/2, i+2+N/2)$ are for orbit planes on which satellites $(S4, S5, S6)$ are moving from up to bottom.

Figure 4: Two satellites with same altitude and inclination (i) move in the same or opposite direction



- * Two satellites $S1$ and $S2$ are at position $P1$ and $P2$ at time $T1$
- * $S1$'s right facing ISL connected to $S2$'s left facing ISL
- * $S1$ and $S2$ move to the position $P4$ and $P3$ at time $T2$
- * $S1$'s left facing ISL connected to $S2$'s right facing ISL

Figure 5: Two satellites meeting in the polar area will change its facing of ISL

5.2.2. Satellites on Adjacent Orbit Planes with Same Altitude

For satellites on different orbit planes with same altitude, the estimation of the lifetime when two satellite can communicate are as follows.

Figure 6 illustrates a general case that two satellites move and intersect with an angle A .

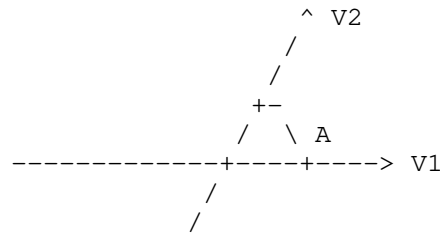


Figure 6: Two satellites (speed vector V1 and V2) intersect with angle A

More specifically, for orbit planes with the inclination angle i , Figure 7 illustrates two satellites move in the opposite direction and intersect with an angle $2*i$.

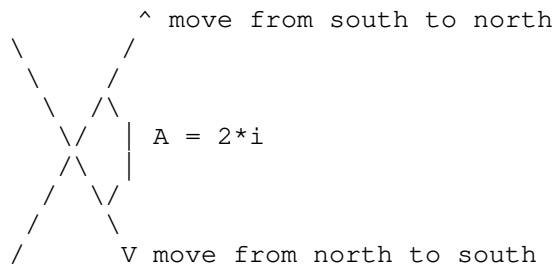


Figure 7: Two satellites with same altitude and inclination (i) intersect with angle $A=2*i$

Follows are the math to calculate the lifetime of communication. Table 3 are the results using the math for two satellites with different altitudes and different inclination angles.

D1 The laser communication limit, $D1=2000\text{km}$
[Laser-communication-range]

A The angle between two orbit's vertical projection on Earth.
 $A=2*i$

V1 The speed vector of satellite on orbit1

V2 The speed vector of satellite on orbit2

$|V|$ the magnitude of the difference of two speed vector V1 and V2, $|V|=|V1-V2|=\sqrt{(V1-V2*\cos(A))^2+(V2*\sin(A))^2}$. For satellites with the same altitude and inclination angle i , $V1=V2$, so, $|V|=V1*\sqrt{2-2*\cos(2*i)}=2V1*\sin(i)$

T The lifetime two satellites can communicate, or the time of two satellites' distance is within the range of communication, $T = 2 \cdot D_l / |V|$.

i (degree)	80	80	65	65	50	50
Alt (km)	500	800	500	800	500	800
V (km/s)	14.98	14.67	13.79	13.5	11.66	11.41
T(s)	267	273	290	296	343	350

Table 3: The lifetime of communication for two LEOs (with two altitudes and three inclination angles)

5.2.3. Satellites on Adjacent Orbit Planes with Different Altitude

For satellites on different orbit planes with different altitude, the estimation of the lifetime when two satellite can communicate are as follows.

Figure 8 illustrates two satellites (with the altitude difference D_a) move and intersect with an angle A .

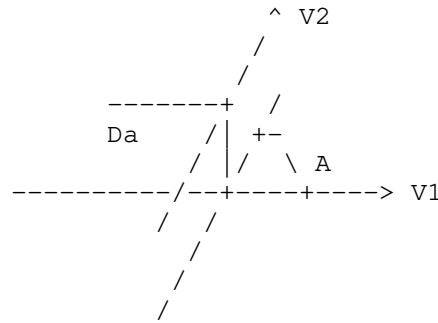


Figure 8: Satellite (speed vector V_1 and V_2 , Altitude difference D_a) intersects with Angle A

Follows are the math to calculate the lifetime of communication

D_l The laser communication limit, $D_l=2000\text{km}$
[Laser-communication-range]

D_a Altitude difference (in km) for two orbit planes

A The angle between two orbit's vertical projection on Earth

V1 The speed vector of satellite on orbit 1

V2 The speed vector of satellite on orbit 2

$|V|$ the magnitude of the difference of two speed vector V1 and V2, $|v| = |V1 - V2| = \sqrt{(V1 - V2 \cos(A))^2 + (V2 \sin(A))^2}$

T The lifetime two satellites can communicate, or the time of two satellites' distance is within the range of communication, $T = 2 \times \sqrt{D1^2 - Da^2} / |V|$

Using formulas above, below is the estimation for the life of communication of two satellites when they intersect. Table 4 and Table 5 are for two VLEOs with the difference of 114.1km for altitude. (VLEO1 and VLEO2 on Table 2). Table 6 and Table 7 are for two LEOs with the difference of 175km for altitude (LEO2 and LEO3 on Table 2).

Parameters	VLEO1	VLEO2
As (km)	335.9	450
V (km/s)	7.7	7.636

Table 4: Two VLEO with different altitude and speed

A (degree)	0	10	45	90	135	180
$ V $ (km/s)	0.065	1.338	5.869	10.844	14.169	15.336
T(s)	61810	2984	680	368	282	260

Table 5: Two VLEO intersects with different angle and the life of communication

Parameters	LEO1	LEO2
As (km)	1150	1325
V (km/s)	7.272	7.189

Table 6: Two LEO with different altitude and speed

A (degree)	0	10	45	90	135	180
V (km/s)	0.083	1.263	5.535	10.226	13.360	14.461
T(s)	47961	3155	720	390	298	276

Table 7: Two LEO intersects with different angle and the life of communication

6. Use Satellite Network for Internet

Since there is no complete satellite network established yet, all following analysis is based on the predictions from the traditional GEO communication. The analysis also learnt how other type of network has been used in Internet, such as Broadband access network, Mobile access network, Enterprise network and Service Provider network.

As a criteria to be part of Internet, any device connected to any satellite should be able to communicate with any public IP4 or IPv6 address in Internet. There could be three types of methods to deliver IP packet from source to destination by satellite:

1. Data packet is relayed between ground station and satellite.
For this method, there is no inter-satellite communication and networking. Data packet is bounced once or couple times between ground stations and satellites until the packet arrives at the destination in Internet.
2. Data packet is delivered by inter-satellite networking.
For this method, the data packet traverses with multiple satellites and inter-satellite networking is used to deliver the packet to the destination in Internet.

3. Both satellite relay and inter-satellite networking are used. For this method, the data packet is relayed in some segments and traverse with multiple satellites in other segments. It is a combination of the method 1 and method 2.

Using the above methods, follows are typical deployment scenarios that a Satellite network is integrated with Internet:

1. The end user terminal access Internet through satellite relay (Figure 9 for one satellite relay, Figure 10 for multiple satellite relay).
2. The end user terminal access Internet through inter-satellite-networking (Figure 11).
3. The local network access Internet through satellite relay (Figure 12 for one satellite relay, Figure 13 for multiple satellite relay).
4. The local network access Internet through inter-satellite-networking (Figure 14).
5. The End user terminal or local network access Internet through satellite network and Mobile Access Network, From mobile access network to satellite network or From satellite network to mobile access network, Satellite network includes inter satellite network and relay network (Figure 15 for mobile access network to satellite network, Figure 16 for satellite netowk to mobile access network).

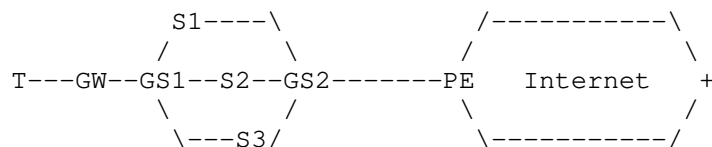


Figure 9: End user terminal access Internet through one satellite relay

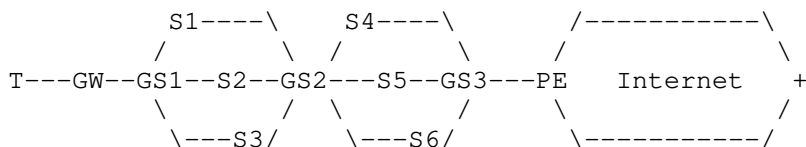


Figure 10: End user terminal access Internet through multiple satellite relay

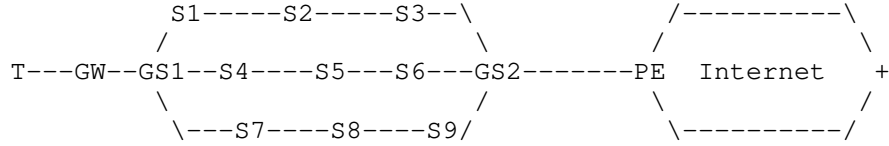


Figure 11: End user terminal access Internet through inter-satellite-networking

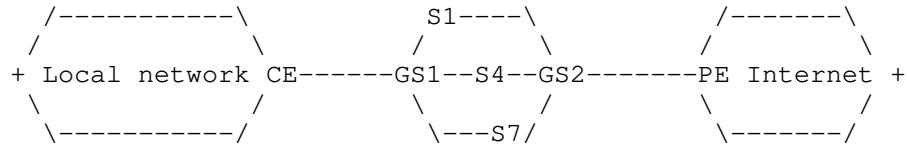


Figure 12: Local network access Internet through one satellite relay

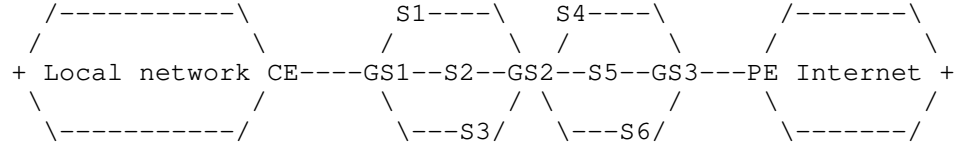


Figure 13: Local network access Internet through multiple satellite relay

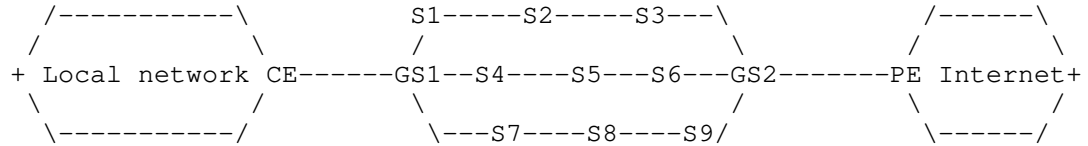


Figure 14: Local network access Internet through inter-satellite-networking

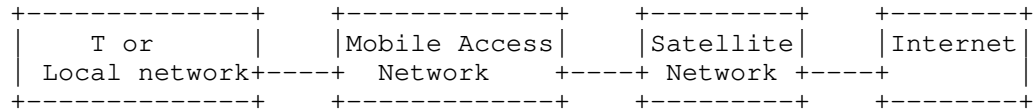


Figure 15: End user terminal or local network access Internet through Mobile Access Network and Satellite Network

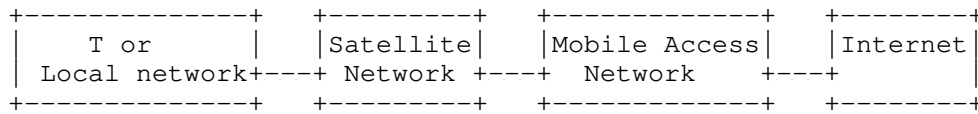


Figure 16: End user terminal or local network access Internet through Satellite Network and Mobile Access Network

In above Figure 9 to Figure 16, the meaning of symbols are as follows:

T	The end user terminal
GW	Gateway router
GS1, GS2, GS3	Ground station with L2/L3 routing/switch functionality.
S1 to S9	Satellites
PE	Provider Edge Router
CE	Customer Edge Router

7. Problems and Requirements for Satellite Constellation for Internet

As described in Section 6, satellites in a satellite constellation can either relay internet traffic or multiple satellites can form a network to deliver internet traffic. More detailed analysis are in following sub sections. There might have multiple solutions for each method described in Section 6, following contexts only discuss the most plausible solution from networking perspectives.

Section 7.1 will list the common problems and requirements for both satellite relay and satellite networking.

Section 7.2 and Section 7.3 will describe key problems, requirement and potential solution from the networking perspective for these two cases respectively.

7.1. Common Problems and Requirements

For both satellite relay and satellite networking, satellite-ground-station must be used, so, the problems and requirements for the satellite-ground-station communication is common and will apply for both methods.

When one satellite is communicating with ground station, the satellite only needs to receive data from uplink of one ground station, process it and then send to the downlink of another ground station. Figure 9 illustrates this case. Normally microwave is used for both links.

Additionally, from the coverage analysis in Section 4.2 and real deployment in Section 4.3, we can see one ground station may communicate with multiple satellites. Similarly, one satellite may communicate with multiple ground stations. The characters for satellite-ground-station communication are:

1. Satellite-ground-station communication is P2MP.
Since microwave physically is the carrier of broadcast communication, one satellite can send data while multiple ground stations can receive it. Similarly, one ground station can send data and multiple satellites can receive it.
2. Satellite-ground-station communication is in open space and not secure.
Since electromagnetic fields for microwave physically are propagating in open space. The satellite-ground-station communication is also in open space. It is not secure naturally.
3. Satellite-ground-station communication is not steady.
Since the satellite is moving with high speed, from Section 5.1, the satellite-ground-station communication can only last a certain period of time. The communication peers will keep changing.
4. Satellite-to-Satellite communication is not steady.
For some satellites, even they are in the same altitude and move in the same speed, but they move in the opposite direction, from Section 5.2.2, the satellite-to-satellite communication can only last a certain period of time. The communication peers will keep changing.
5. Satellite-to-Satellite distance is not steady.
For satellites with the same altitude and same moving direction, even their relative position is steady, but the distance between satellites are not steady. This will lead to the inter-satellite-communication's bandwidth and latency keep changing.

6. Satellite physical resource is limited.
Due to the weight, complexity and cost constraint, the physical resource on a satellite, such as power supply, memory, link speed, are limited. It cannot be compared with the similar device on ground. The design and technology used should consider these factors and take the appropriate approach if possible.

The requirements of satellite-ground-station communication are:

- R1. The bi-directional communication capability
Both satellites and ground stations have the bi-directional communication capability
- R2. The identifier for satellites and ground stations
Satellites and ground stations should have Ethernet and/or IP address configured for the device and each link. More detailed address configuration can be seen in each solution.
- R3. The capability to decide where the IP packet is forwarded to.
In order to send Internet traffic or IP data to destination correctly, satellites and ground station must have Ethernet hub or switching or IP routing capability. More detailed capability can be seen in each solution.
- R4. The protocol to establish the satellite-ground-station communication.
For security and management purpose, the satellite-ground-station communication is only allowed after both sides agree through a protocol. The protocol should be able to establish a secured channel for the communication when a new communication peer comes up. Each ground station should be able to establish multiple channels to communicate with multiple satellites. Similarly, each satellite should be able to establish multiple channels to communicate to multiple ground stations.
- R5. The protocol to discover the state of communication peer.
The discover protocol is needed to detect the state of communication peer such as peer's identity, the state of the peer and other info of the peer. The protocol must be running securely without leaking the discovered info.
- R6. The internet data packet is forwarded securely.
When satellite or ground station is sending the IP packet to its peer, the packet must be relayed securely without leaking the user data.

R7. The internet data packet is processed efficiently on satellite

Due to the resource constraint on a satellite, the packet may need more efficient mechanism to be processed on satellite. The process on satellite should be very minimal and offloaded to ground as much as possible.

7.2. Satellite Relay

One of the reasons to use satellite constellation for internet access is it can provide shorter latency than using the fiber underground. But using ISL for inter-satellite communication is the premise for such benefit in latency. Since the ISL is still not mature and adopted commercially, satellite relay is a only choice currently for satellite constellation used for internet access. In [UCL-Mark-Handley], detailed simulations have demonstrated better latency than fiber network by satellite relay even the ISL is not present.

7.2.1. One Satellite Relay

One satellite relay is the simplest method for satellite constellation to provide Internet service. By this method, IP traffic will be relayed by one satellite to reach the DGS and go to Internet.

The solution option and associated requirements are:

S1. The satellite only does L1 relay or the physical signal process.

For this solution, a satellite only receives physical signal, amplify it and broadcast to ground stations. It has no further process for packet, such as L2 packet compositing and processing, etc. All packet level work is done only at ground station. The requirements for the solution are:

R1-1. SGS and BGS are configured as IP routing node. Routing protocol is running in SGS and BGS

SGS and BGS is a IP peer for a routing protocol (IGP or BGP). SGS will send internet traffic to DGS as next hop through satellite uplink and downlink.

R1-2. DGS must be connected with Internet.

DGS can process received packet from satellite and forward the packet to the destination in Internet.

In addition to the above requirements, following problem should be solved:

P1-1. IP continuity between two ground stations

This problem is that two ground stations are connected by one satellite relay. Since the satellite is moving, the IP continuity between ground stations is interrupted by satellite changing periodically. Even though this is not killing problem from the view point that IP service traditionally is only a best effort service, it will benefit the service if the problem can be solved. Different approaches may exist, such as using hands off protocols, multipath solutions, etc.

S2. The satellite does the L2 relay or L2 packet process.

For this solution, IP packet is passing through individual satellite as an L2 capable device. Unlike in the solution S1, satellite knows which ground station it should send based on packet's destination MAC address after L2 processing. The advantage of this solution over S1 is it can use narrower beam to communicate with DGS and get higher bandwidth and better security. The requirements for the solution are:

R2-1. Satellite must have L2 bridge or switch capability

In order to forward packet to properly, satellite should run some L2 process such as MAC learning, MAC switching. The protocol running on satellite must consider the fast movement of satellite and its impact to protocol convergence, timer configuration, table refreshment, etc.

R2-2. same as R1-1 in S1

R2-3. same as R1-2 in S1

In addition to the above requirements, the problem P1-1 for S1 should also apply.

7.2.2. Multiple Satellite Relay

For this method, packet from SGS will be relayed through multiple intermediate satellites and ground station until reaching a DGS.

This is more complicated than one satellite relay described in Section 7.2.1.

One general solution is to configure both satellites and ground-stations as IP routing nodes, proper routing protocols are running in this network. The routing protocol will dynamically determine forwarding path. The obvious challenge for this solution is that all links between satellite and ground station are not static, according to the analysis in Section 5.1, the lifetime of each link may last

only couple of minutes. This will result in very quick and constant topology changes in both link state and IP adjacency, it will cause the distributed routing algorithms may never converge. So this solution is not feasible.

Another plausible solution is to specify path statically. The path is composed of a serials of intermediate ground stations plus SGS and DGS. This idea will make ground stations static and leave the satellites dynamic. It will reduce the fluctuation of network path, thus provide more steady service. One variant for the solution is whether the intermediate ground stations are connected to Internet. Separated discussion is as below:

S1. Manual configuring routing path and table

For this solution, the intermediate ground stations and DGS are specified and configured manually during the stage of network planning and provisioning. Following requirements apply:

R1-1. Specify a path from SGS to DGS via a list of intermediate ground stations.

The specified DGS must be connected with internet. Other specified intermediate ground stations does not have to

R1-2. All Ground stations are configured as IP routing node. Static routing table on all ground stations must be pre-configured, the next hop of routes to Internet destination in any ground station is configured to going through uplink of satellite to the next ground station until reaching the DGS.

R1-3. All Satellites are configured as either L1 relay or L2 relay.

The Satellite can be configured as L1 relay or L2 relay described in S1 and S2 respectively in Section 7.2.1

In addition to the above requirements, the problem P1-1 in Section 7.2.1 should also apply.

S2. Automatic decision by routing protocol.

This solution is only feasible after the IP continuity problem (P1-1 in Section 7.2.1) is solved. Following requirements apply:

R2-1. All Ground stations are configured as IP routing node.
Proper routing protocols are configured as well.

The satellite link cost is configured to be lower than the ground link. In such a way, the next hop of routes for the IP forwarding to Internet destination in any ground station will be always going through the uplink of satellite to the next ground station until reaching the DGS.

R2-2. All Satellites are configured as either L1 relay or L2 relay.

The Satellite can be configured as L1 relay or L2 relay described in S1 and S2 respectively in Section 7.2.1

In addition to the above requirements, the problem P1-1 in Section 7.2.1 should also apply.

7.3. Satellite Networking

In the draft, satellite Network is defined as a network that satellites are inter-connected by inter-satellite links (ISL). One of the major difference of satellite network with the other type of network on ground (telephone, fiber, etc.) is its topology and links are not stationary, some new issues have to be considered and solved. Follows are the factors that impact the satellite networking.

7.3.1. L2 or L3 network

The 1st question to answer is should the satellite network be configured as L2 or L3 network? As analyzed in Section 4.2 and Section 4.3, since there are couple of hundred or over ten thousand satellites in a network, L2 network is not a good choice, instead, L3 or IP network is more appropriate for such scale of network.

7.3.2. Inter-satellite-Link Lifetime

If we assume the orbit is circular and ignore other trivial factors, the satellite speed is approximately determined by the orbit altitude as described in the Section 5.1. The satellite orbit can determine if the dynamic position of two satellites is within the range of the inter-satellite communication. That is 2000km for laser communication [Laser-communication-range] by Inter Satellite Laser Link (ISLL).

When two satellites' orbit planes belong to the same group, or two orbit planes share the same altitude and inclination, and when the satellites move in the same direction, the relative positions of two satellites are relatively stationary, and the inter-satellite communication is steady. But when the satellites move in the

opposite direction, the relative positions of two satellites are not stationary, the communication lifetime is couple of minutes. The Section 5.2.2 has analyzed the scenario.

When two satellites' orbit planes belong to the different group, or two orbit planes have different altitude, the relative position of two satellite are unstable, and the inter-satellite communication is not steady. As described in Section 5.2, The life of communication for two satellites depends on the following parameters of two satellites:

1. The speed vectors.
2. The altitude difference
3. The intersection angle

From the examples shown in Table 4 to Table 7, we can see that the lifetime of inter-satellite communication for the different group of orbit planes are from couple of hundred seconds to about 18 hours. This fact will impact the routing technologies used for satellite network and will be discussed in Section 7.3.3.

7.3.3. Problems for Traditional Routing Technologies

When the satellite network is integrated with Internet by traditional routing technologies, following provisioning and configuration (see Figure 17) will apply:

1. The ground stations connected to local network and internet are treated as PE router for satellite network (called PE_GS1 and PE_GS2 in the following context), and all satellites are treated as P router.
2. All satellites in the network and ground stations are configured to run IGP.
3. The eBGP is configured between PE_GS and its peered network's PE or CE.

The work on PE_GS1 are:

- * The local network routes are received at PE_GS1 from CE by eBGP. The routes are redistributed to IGP and then IGP flood them to all satellites. (Other more efficient methods, such as iBGP or BGP reflectors are hard to be used, since the satellite is moving and there is no easy way to configure a full meshed iBGP session for all satellites, or configure one satellite as BGP reflector in satellite network.)
- * The internet routes are redistributed from IGP to eBGP running on PE_GS1, and eBGP will advertise them to CE.

The work on PE_GS2 are:

- * The Internet routes are received at PE_GS2 from PE by eBGP. The routes are redistributed to IGP and then IGP flood them to all satellites. (Similar as in PE_GS1, Other more efficient methods, such as iBGP or BGP reflector cannot be used.)
- * The local network routes are redistributed from IGP to eBGP running on PE_GS2, and eBGP will advertise them to Internet.

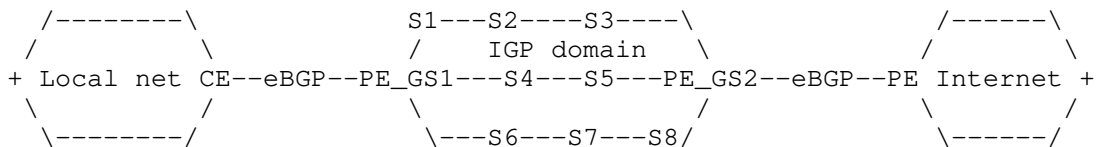


Figure 17: Local access Internet through inter-satellite-networking

Local access Internet through inter-satellite-networking

On PE-GS1, due to the fact that IGP link between PE_GS1 and satellite is not steady; this will lead to following routing activity:

1. When one satellite is connecting with PE_GS1, the satellite and PE_GS1 form a IGP adjacency. IGP starts to exchange the link state update.
2. The local network routes received by eBGP in PE_GS1 from CE are redistributed to IGP, and IGP starts to flood link state update to all satellites.
3. Meanwhile, the Internet routes learnt from IGP in PE_GS1 will be redistributed to eBGP. eBGP starts to advertise to CE.
4. Every satellite will update its routing table (RIB) and forwarding table (FIB) after IGP finishes the SPF algorithm.

5. When the satellite is disconnecting with PE-GS1, the IGP adjacency between satellite and PE_GS1 is gone. IGP starts to exchange the link state update.
6. The routes of local network and satellite network that were redistributed to IGP in step 2 will be withdrawn, and IGP starts to flood link state update to all satellites.
7. Meanwhile, the Internet routes previously redistributed to eBGP in step 3 will also be withdrawn. eBGP starts to advertise route withdraw to CE.
8. Every satellite will update its routing table (RIB) and forwarding table (FIB) after the SPF algorithm.

Similarly on PE_GS2, due to the fact that IGP link between PE_GS2 and satellite is not steady; this will lead to following routing activity:

1. When one satellite is connecting with PE_GS2, the satellite and PE_GS2 form a IGP adjacency. IGP starts to exchange the link state update.
2. The Internet routes previously received by eBGP in PE_GS2 from PE are redistributed to IGP, IGP starts to flood the new link state update to all satellites.
3. Meanwhile, the routes of local network and satellite network learnt from IGP in PE_GS2 will be redistributed to eBGP. eBGP starts to advertise to Internet peer PE.
4. Every satellite will update its routing table (RIB) and forwarding table (FIB) after IGP finishes the SPF algorithm.
5. When the satellite is disconnecting with PE-GS2, the IGP adjacency between satellite and PE_GS2 is gone. IGP starts to exchange the link state update.
6. The internet routes previously redistributed to IGP in step 2 will be withdrawn, and IGP starts to flood link state update to all satellites
7. Meanwhile, the routes of local network and satellite network previously redistributed to eBGP in step 3 will also be withdrawn. eBGP starts to advertise route withdraw to PE.
8. Every satellite will update its routing table (RIB) and forwarding table (FIB) after the SPF algorithm.

For the analysis of detailed events above, the estimated time interval between event 1 and 5 for PE_GS1 and PE_GS2 can use the analysis in Section 5.1. For example, it is about 398s for LEO and 103s for VLEO. Within this time interval, the satellite network including all satellites and two ground stations must finish the works from 1 to 4 for PE_GS1 and PE_GS2. The normal internet IPv6 and IPv4 BGP routes size are about 850k v4 routes + 100K v6 routes [BGP-Table-Size]. There are couple critical problems associated with the events:

P1. Frequent IGP update for its link cost

Even for satellites in different orbit with the steady relative positions, the distance between satellites is keep changing. If the distance is used as the link cost, it means the IGP has to update the link cost frequently. This will make IGP keep running and update its routing table.

P2. Frequent IGP flooding for the internet routes

Whenever the IGP adjacency changes (step 1 and 5 for PE_GS2), it will trigger the massive IGP flooding for the link state update for massive internet routes learnt from eBGP. This will result in the IGP re-convergency, RIB and FIB update.

P3. Frequent BGP advertisement for the internet routes

Whenever the IGP adjacency changes (step 3 and 7 for PE_GS1), it will trigger the massive BGP advertisement for the internet routes learnt from IGP. This will result in the BGP re-convergency, RIB and FIB update. BGP convergency time is longer than IGP. The document [BGP-Converge-Time1] has shown that the BGP convergence time varies from 50sec to couple of hundred seconds. The analysis [BGP-Converge-Time2] indicated that per entry update takes about 150us, and it takes $O(75s)$ for 500k routes, or $O(150s)$ for 1M routes.

P4. More frequent IGP flooding and BGP update in whole satellite network

To provide the global coverage, a satellite constellation will have many ground stations deployed. For example, StarLink has applied for the license for up to one million ground stations [StarLink-Ground-Station-Fcc], in which, more than 50 gateway ground stations (equivalent to the PE_GS2) have been registered [SpaceX-Ground-Station-Fcc] and deployed in U.S. [StarLink-GW-GS-map]. It is expected that the gateway ground station will grow quickly to couple of thousands [Tech-Comparison-LEOs]. This means almost each satellite in the satellite network would have a ground station connected. , Due to the fact that all satellites are moving, many IGP adjacency changes may occur in a shorter period of time described in Section 5.1 and result in the problem P1 and P2 constantly occur.

P5. Service is not steady

Due to the problems P1 to P3, the service provider of satellite constellation is hard to provide a steady service for broadband service by using inter-satellite network and traditional routing technologies.

As a summary, the traditional routing technology is problematic for large scale inter-satellite networking for Internet. Enhancements on traditional technologies, or new technologies are expected to solve the specific issues associated with satellite networking.

8. IANA Considerations

This memo includes no request to IANA.

9. Contributors

10. Acknowledgements

11. References

11.1. Normative References

- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/info/rfc4271>>.
- [RFC2328] Moy, J., "OSPF Version 2", STD 54, RFC 2328, DOI 10.17487/RFC2328, April 1998, <<https://www.rfc-editor.org/info/rfc2328>>.

- [RFC7142] Shand, M. and L. Ginsberg, "Reclassification of RFC 1142 to Historic", RFC 7142, DOI 10.17487/RFC7142, February 2014, <<https://www.rfc-editor.org/info/rfc7142>>.
- [RFC2453] Malkin, G., "RIP Version 2", STD 56, RFC 2453, DOI 10.17487/RFC2453, November 1998, <<https://www.rfc-editor.org/info/rfc2453>>.
- [RFC7868] Savage, D., Ng, J., Moore, S., Slice, D., Paluch, P., and R. White, "Cisco's Enhanced Interior Gateway Routing Protocol (EIGRP)", RFC 7868, DOI 10.17487/RFC7868, May 2016, <<https://www.rfc-editor.org/info/rfc7868>>.

11.2. Informative References

- [KeplerianElement] "Keplerian elements", <https://en.wikipedia.org/wiki/Orbital_elements>.
- [GEO-Coverage] "Coverage of a geostationary satellite at Earth", <<https://www.planetary.org/space-images/coverage-of-a-geostationary>>.
- [Nttdocomo-6G] "NTTDP COM 6G White Paper", <https://www.nttdocomo.co.jp/english/binary/pdf/corporate/technology/whitepaper_6g/DOCOMO_6G_White_PaperEN_20200124.pdf>.
- [ITU-6G] "ITU 6G vision", <https://www.itu.int/dms_pub/itu-s/opb/itu_jnl/S-ITUJNL-JFETF.V1I1-2020-P09-PDF-E.pdf>.
- [Surrey-6G] "Surrey 6G vision", <<https://www.surrey.ac.uk/sites/default/files/2020-11/6g-wireless-a-new-strategic-vision-paper.pdf>>.
- [OSI-Model] "OSI Model", <https://en.wikipedia.org/wiki/OSI_model>.
- [StarLink] "Star Link", <<https://en.wikipedia.org/wiki/Starlink>>.
- [China-constellation] "China Constellation", <<https://www.itu.int/ITU-R/space/asreceived/Publication/DisplayPublication/23706>>.

[ESA-HydRON]

"HydRON: Fiber in the sky",
<https://www.esa.int/ESA_Multimedia/Videos/2021/04/HydRON_Fibre_in_the_sky>.

[SpaceX-Non-GEO]

"FCC report: SPACEX V-BAND NON-GEOSTATIONARY SATELLITE SYSTEM", <<https://fcc.report/IBFS/SAT-LOA-20170301-00027/1190019.pdf>>.

[Satellite-coverage]

Alan R.Washburn, Department of Operations Research, Naval Postgraduate School, "Earth Coverage by Satellites in Circular Orbit",
<<https://faculty.nps.edu/awashburn/Files/Notes/EARTHCOV.pdf>>.

[Microwave-vs-Laser-communication]

International Journal for Research in Applied Science and Engineering Technology (IJRASET), "Comparison of Microwave and Optical Wireless Inter-Satellite Links",
<<https://www.ijraset.com/files/serve.php?FID=7815>>.

[Laser-communication-range]

"Interferometric optical communications can potentially lead to robust, secure, and naturally encrypted long-distance laser communications in space by taking advantage of the underlying physics of quantum entanglement.",
<<https://www.laserfocusworld.com/optics/article/16551652/interferometry-quantum-entanglement-physics-secures-spacetospace-interferometric-communications>>.

[BGP-Table-Size]

"BGP in 2020 - BGP table",
<<https://blog.apnic.net/2021/01/05/bgp-in-2020-the-bgp-table/>>.

[BGP-Converge-Time1]

"BGP in 2020 - BGP Update Churn",
<<https://labs.apnic.net/?p=1397>>.

[BGP-Converge-Time2]

"Bringing SDN to the Internet, one exchange point at the time",
<<https://www.cs.princeton.edu/courses/archive/fall14/cos561/docs/SDX.pdf>>.

- [StarLink-Ground-Station-Fcc]
"APPLICATION FOR BLANKET LICENSED EARTH STATIONS",
<<https://fcc.report/IBFS/SES-LIC-INTR2019-00217/1616678>>.
- [SpaceX-Ground-Station-Fcc]
"List of SpaceX applications for ground stations",
<<https://fcc.report/IBFS/Company/Space-Exploration-Technologies-Corp-SpaceX>>.
- [Tech-Comparison-LEOs]
"A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband",
<<http://www.mit.edu/~portillo/files/Comparison-LEO-IAC-2018-slides.pdf>>.
- [StarLink-GW-GS-map]
"StarLink gateway ground station map",
<https://www.google.com/maps/d/u/0/viewer?mid=1Hlx8jZs8vfjy60TvKgpbYs_grargieVw>.
- [UCL-Mark-Handley]
"Using ground relays for low-latency wide-area routing in megaconstellations",
<<https://discovery.ucl.ac.uk/id/eprint/10090242/1/hotnets-ucl.pdf>>.

Appendix A. Change Log

- * Initial version, 07/03/2021
- * 01 version, 10/20/2021

Authors' Addresses

Lin Han (editor)
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America

Email: lhhan@futurewei.com

Richard Li
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America

Email: rli@futurewei.com

Alvaro Retana
Futurewei Technologies, Inc.
2330 Central Expy
Santa Clara, CA 95050,
United States of America

Email: alvaro.retana@futurewei.com

Meiling Chen
China Mobile
32, Xuanwumen West
BeiJing 100053
China

Email: chenmeiling@chinamobile.com

Li Su
China Mobile
32, Xuanwumen West
BeiJing 100053
China

Email: suli@chinamobile.com

Ning Wang
University of Surrey
Guildford
Surrey, GU2 7XH
United Kingdom

Email: n.wang@surrey.ac.uk

INTAREA
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

C. Sarathchandra
M. Kheirkhah
M. Ghassemian
InterDigital Europe, Ltd.
July 12, 2021

Tactile Internet Service Requirements
draft-sarathchandra-tactile-internet-01

Abstract

The Tactile Internet refers to a new communication paradigm, which can provide low-latency, reliable and secure transmission for real-time information such as control, touch, and sensing/actuation in emerging tactile internet applications like teleoperation, immersive virtual reality, and haptics communications. The main goal of this document is: 1) to briefly introduce tactile internet background and use cases; 2) to identify potential service requirements that can be addressed at the IETF or researched at the IRTF.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
2. Terminology	3
3. Abbreviations List	3
4. Use Cases	3
4.1. Industry	4
4.2. Healthcare	4
4.3. Entertainment	4
4.4. Training	5
5. User Equipment Capabilities	5
6. TI Service Requirements	6
6.1. Haptic Media Type	6
6.2. Ultra-Low Latency	6
6.3. Ultra-High Reliability	7
6.4. Synchronization	7
6.5. Application-Network Interaction	7
6.6. Multi-Modal Coordinated Parallel Transmission	8
6.7. Personalised Multi-Modal Experiences	8
7. IANA Considerations	9
8. Security Considerations	9
9. Conclusion	9
10. Acknowledgments	10
11. References	10
11.1. Normative References	10
11.2. Informative References	10
Authors' Addresses	12

1. Introduction

Tactile Internet (TI) was defined as a new wave of innovation after the successful Internet of Things (IoT) [ITU-T2014]. In fact, Tactile Internet (TI) can be regarded as a new ICT paradigm with extreme emphasises and service requirements on multiple performance metrics such as latency, availability, reliability, and security. TI finds its application in many emerging application scenarios, including, but not limited to, Industry, Robotics and Telepresence, eXtended Reality (e.g., Augmented Reality, Virtual Reality and Mixed Reality), Healthcare, Gaming, and Teleoperation.

These extreme service requirements from TI applications pose new challenges to both communication and computing. Although existing networking architecture and protocols can support some of these

service requirements partially (e.g., 5G URLLC [URLLC-3GPP]), a still pending question is whether and how a holistic and systematic approach should be developed in order to efficiently support TI applications. Moreover, IEEE 1918.1 standards working group [IEEE19181] on TI is formed to investigate aspects related to TI applications, architecture and haptic encoding.

2. Terminology

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 [RFC2119].

3. Abbreviations List

- o TI - Tactile Internet
- o TD - Tactile Devices
- o UE - User Equipment
- o URLLC - Ultra-Reliable Low-Latency Communications
- o AR - Augmented Reality
- o VR - Virtual Reality
- o PPE - Personal Protective Equipment
- o ISOBMFF - ISO Base Media File Format
- o QoE - Quality of Experience
- o QoS - Quality of Service
- o AES - Advanced Encryption Standard
- o WEP - Wired Equivalent Privacy
- o WPA - Wi-Fi Protected Access

4. Use Cases

This section aims to introduce the reader to distinct, although not exhaustive, TI applications which are widely being discussed in the TI research community.

4.1. Industry

Automation, smart factories and remote operation are some of key industry use cases that are enabled by TI [IndustryTI]. Moreover, repair and maintenance in remote areas, in high-risk scenarios requiring high precision requires multi-modal [TactileMultimodal-3GPP] and low latency communication provided by TI. For example, in such scenarios, human operators can control machinery (e.g., robots) remotely and perform complex operations [IndustryRobot], where either it is too dangerous for humans to be present, or it's not possible for the experts to be physically present at the environment where the operations are conducted. The controlled machinery may be equipped with various sensors for providing information about the environment to the operator, while it may also be equipped with required actuators for performing corresponding tasks as instructed by the constructor over the TI. TI may also enable the transmission of critical information (e.g., alerts) to human users (e.g., through connected PPE as AR and haptic data) who perform operations in high-risk environments. Alerts may be automatically generated based on information gathered from sensors, or sent by human users, over the TI.

4.2. Healthcare

Key health applications of TI include, tele-surgery [Independent], tele-mentoring, tele-rehabilitation and tele-diagnosis [TIAijaz2019]. Specifically, minimising the invasive nature of surgery has been a focus of the health technology industry and has currently been widely used due to the small tissue damage and fast recovery it incurs. Today, surgeons use surgical robots for performing highly precise operations. Providing tactile feedback is specifically critical for performing operations which require high precision manipulation. Although, it is not always possible to get specialist surgeons on site for performing operations on patients, TI enables surgeons to perform such critical operations remotely, where it requires only the machinery (high precision robots) to be co-located alongside the patient.

4.3. Entertainment

The advancements in Augmented Reality (AR) & Virtual Reality (VR) technology as well as the increased number of applications developed for user entertainment (e.g., VR gaming, VR tourism, VR art) have significantly increased the interest for further improving the immersive experiences those application provide. VR applications enable human users, or a collection of human users to interact with a virtual environment where the provided immersive experience is similar to that of a real physical interaction. Haptic feedback is a

key element in such interactions, allowing the user to experience the sense of touch along with audio and visual (e.g., users perceiving the effect of each other's actions in collaborative scenarios).

4.4. Training

TI enables learning experiences where tactile feedback plays a crucial role. This may substantially improve both the learning as well as the teaching experiences in remote learning scenarios. The teacher will be able to experience (see, hear, feel) actions performed by the learner and correct any errors as if they are in a real physical (face-to-face) learning environment. Such applications include, remote military and sports training [na2020simulation] which requires problem solving by collaborating with remote team members, while incorporating feedback provided by the remote trainer in real-time. Furthermore, Internet of Skills [InternetofSkills] application aims at training people in remote and diverse locations to improve their skills and capabilities. It combines advances in motor training and Tactile Internet with Human-in-the-loop to achieve the goal of transferring high quality skills to populations that otherwise do not have access to such training. Moreover, the goal of Surgical Assistance and training [SurgicalTraining] application is to develop a system that provides assistance to an expert surgeon during a surgery or to provide surgery training to students. Such a system is envisaged to be continuously learning and acquiring expert knowledge. To do this, the system interprets sensor data as it observes an expert surgeon performing their procedure.

5. User Equipment Capabilities

Various sensors, actuators, display devices are used to provide a realistic haptic and multimodal interaction with the remote devices over a uni-directional or bi-directional communication. The sensor components capture the tele-manipulation instructions (e.g., kinaesthetic), and the resulting changes (e.g., haptic feedback). Actuators execute the user's tele-manipulation instructions. The number of independent coordinates used for providing the end user experiences (using Human System Interfaces), and for controlling the velocity, position, and the orientation of the controlled devices is defined by their degree of freedom (DoF).

Capabilities of UEs in collecting biometrics can enhance security solutions (such as user identification and authentication). While existing authentication mechanisms relay on SIM (subscriber identity module or subscriber identification module) cards in mobile devices, unique biometrics collected from the users can be used to enhance the security. Considering a scenario where the SIM card token is stolen, an alternative/complementary method of ensuring network connectivity

for the genuine user would involve the use of biometrics as these cannot easily be stolen. Biometrics offers a solution to the weaknesses of knowledge and token-based systems. Examples of continuous biometrics are face, iris, keystroke dynamics, touchscreen gestures, behavioural profiling (e.g. Bluetooth/Wifi/GPS), gait, mood and one-shot biometrics are face, iris, and fingerprint that can be collected by the new UE.

6. TI Service Requirements

As a result of the research and developments in TI, this section presents service requirements to be addressed by the networking community.

6.1. Haptic Media Type

Unlike audio and video, there has not been any haptic media types in standards, until a very recent development in standards to register haptics as a top-level media type. A proposal to introduce haptics as a first-order media type in ISO Base Media File Format (ISOBMFF) was accepted by MPEG Systems File Format sub-group. This standardization process is expected to conclude in October 2021, making haptics a part of the ISO/IEC 14496-12 (ISOBMFF) standard. Providing this recent development, the authors [I-D.muthusamy-dispatch-haptics] make a case for haptics to be added to the list of top-level media types recognised by the IETF. The authors further argue that 'application' top-level type not suitable for haptics as, like audio/video haptics is related to a separate sensory system. Moreover, 'application' is historically used for application code, and haptics is not code but a property of a media stream (like audio and video). Therefore, we believe that the adoption of a top-level haptics media type in IETF is an important step towards further development of haptic communication.

6.2. Ultra-Low Latency

Most Haptic applications demands stringent latency requirements from the underlying communication. Specifically, ultra-low latency, 1ms for haptic interaction [ITU-T2014], is demanded for providing timely delivery of messages between communicating devices by TI applications. The timely delivery of control messages is crucial for critical TI applications such as TI remote surgery. Moreover, timely delivery of messages also assists in playback of multi modal [TactileMultimodal-3GPP] streams (audio, video, haptic) in a synchronous manner, providing a consistent experience that is devoid of cybersickness.

6.3. Ultra-High Reliability

Ultra-high reliability is required by several TI applications. For example, it is not acceptable for communication reliability to be hindered during critical TI applications such as alert transmission for connected PPE (described Section 4.1). Thus, it is crucial that ultra-reliable communication is a key enabler of TI applications.

6.4. Synchronization

The tactile applications often consist of several streams, e.g., audio, video, haptic, each stream with varying service requirements (bitrates, latency, level of reliability). Moreover, depending on the use case and the deployment scenario, streams of an application may be distributed among multiple tactile/terminal devices, e.g., video stream to display, audio stream to sound system, haptic stream to haptic suit. However, all such streams must be played back to the user in a synchronous manner when providing multi-sensory immersive experiences.

Especially, in scenarios where a user uses multiple UEs/terminals for consuming the same user experience, media streams (haptic, audio, video) must be delivered and played to the user in a synchronous manner (e.g., avoiding Cybersickness [Promwongsa]). Due to network conditions and the insufficient support/assistance for synchronization, related streams may arrive at different UEs/terminals out of synchronization (e.g., the lack of information related to inter-dependency among network flows [ITU-NET2030]). Therefore, mechanisms for the coordination (see section 6.6 for detailed discussion) and synchronization of multiple flows, for both the same destination/UE, and for multiple destinations/UEs must be introduced.

6.5. Application-Network Interaction

Emerging TI applications are highly diverse in terms of their use case requirements and constraints. For example, a TI application may comprise multiple streams (e.g., due to multi-modal [TactileMultimodal-3GPP] nature), each of which may be required to be treated differently by the network based on their use case requirements and constraints; some streams may need high bandwidth and ultra-low latency while some others may require ultra-high reliability. The conventional interaction model between applications (end-hosts) and networks are insufficient to deliver the traffic of these emerging TI applications. In other words, applications should not consider the network as a black-box anymore and in turn they should not entirely rely on the end-to-end measurements for adapting

their behaviour as the underlying network condition changes rapidly, mainly because the end-to-end measurements are implicit and thus coarse-grained.

To this end, a new collaborative paradigm between applications and networks need to be realized. This way, applications and networks can express their desired use case requirements and constraints to one another, permitting applications in particular to adapt themselves to network constraints and the networks to orchestrate their resource distribution according to the applications' requirements if desired. This is particularly essential for TI terminals which have to run highly diverse applications/services often with conflicting requirements.

6.6. Multi-Modal Coordinated Parallel Transmission

Applications in TI typically follow a multi-modal communication [TactileMultimodal-3GPP] pattern in which the end-to-end communication between tactile devices (TDs) includes several modes of communication at the same time (e.g., video, audio and haptic). This results in generation of multiple coordinated streams in parallel which ultimately need to be presented to an end user in harmony. Otherwise, the quality of experience (QoE) of the user may not be satisfactory due to lack of precise synchronization across these parallel streams. For example, one stream may get delayed while others are delivered on time. Apart from the synchronization challenges (see also Section 5.4 for more detailed discussion), the instability of the underlying network condition of a stream may also impact the performance of the other coordinated parallel streams of the same TI application, which may ultimately reduce the overall QoE of users. Therefore, it is crucial to have mechanisms particularly tailored for coordination (e.g., data packet scheduling across multiple terminals and/or access networks) so that varying network condition across multiple networks can be intelligently handled. The key goal here is to distribute data packets without creating network congestion and/or increasing end-to-end delay. These type of communications can also significantly benefit when there is a feedback loop mechanism between TI applications (terminals) and networks (see Section Section 6.5 for more details).

6.7. Personalised Multi-Modal Experiences

The TI use cases are highly dynamic in nature. Especially, in multi-user scenarios where user profiles, their dynamic relations and interactions are taken into consideration, e.g., virtual simulation environments used for training, where multiple users act upon the same virtual objects, the information received by individual 'trainee users' may differ due to 1) user preferences (e.g., with haptic

feedback vs without) 2) specific user's perception (e.g., audio, video haptic) of objects and actions/events within the virtual environment (e.g., based on viewpoint, distance to objects/events, and properties of virtual objects). Moreover, the trainer (human or virtual) may choose to provide corresponding feedback (using audio-visual or audio-visual-haptic mediums) to an individual or a group/subset of trainees in real-time. Different users may receive different haptic feedback depending on the type of actions performed and therefore the experiences may differ for each user. When providing such experiences the resulting dynamicity must be considered. Therefore, the multi-modal information provided to each user, through data streams, may be personalised (e.g., based on distinct user perception and user profile).

7. IANA Considerations

This document requests no IANA actions.

8. Security Considerations

Security and trust as well as communication latency are key challenges for delivering tele-surgery. Conventional internet security protocols (namely, AES, WEP, WPA) are used to make the data transfer prone to attack.

Security and reliability of the haptic data locally/remotely are key to Tactile Internet use-cases such as telesurgery use-case. Further work is required on security/privacy aware haptic data/feedback encoding techniques to improve the reliability and security of the TI use-cases. Furthermore, continuous monitoring demands low-power and reliable operation to avoid any interruption in data collection from power restricted devices and therefore the service delivery [monaICC2020].

9. Conclusion

This draft presents the emerging area of Tactile Internet, its key use cases and service requirements. The introduction of haptic communication, a new mode of communication, not only improves existing immersive experiences (e.g., AR/VR) while also facilitates new emerging Tactile immersive experiences (e.g., tele-surgery). Moreover, the resulting communication over the Tactile Internet demands for stringent service requirements on the underlying communication networks, e.g., ultra-high reliability, ultra-low latency transmission, security consideration and synchronization of multi-modal data (including haptic). Therefore, We believe IETF is a key forum for addressing some of the potential challenges described,

for realizing the envisioned Tactile Internet, and for standardizing relevant aspects such as protocols.

10. Acknowledgments

The authors would like to thank Renan Krishna for reviewing and providing useful comments.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

11.2. Informative References

- [Holland] Holland, O. and et. al., "The IEEE 1918.1 "Tactile Internet" Standards Working Group and its Standards", Proceedings of IEEE , 2019, <<https://ieeexplore.ieee.org/document/8605315>>.
- [I-D.muthusamy-dispatch-haptics] Muthusamy, Y. K. and C. Ullrich, "The 'haptics' Top-level Media Type", draft-muthusamy-dispatch-haptics-01 (work in progress), November 2020.
- [IEEE19181] ITU Network 2030 Technical Report, "Network 2030 - Gap analysis of Network 2030 new services, capabilities and use cases", 2020, <<https://www.itu.int/pub/T-FG-NET2030-2020-1>>.
- [Independent] Independent News Article, "SURGEON PERFORMS WORLD'S FIRST REMOTE OPERATION USING '5G SURGERY' ON ANIMAL IN CHINA", 2019, <<https://www.independent.co.uk/life-style/gadgets-and-tech/news/5g-surgery-china-robotic-operation-a8732861.html>>.
- [IndustryRobot] ABmann, U. and et. al., "Human-robot cohabitation in industry", In Tactile Internet, Academic Press pp. 41-73, 2021.

[IndustryTI]

Aijaz, A. and et. al., "The Tactile Internet for Industries: A Review", In Proceedings of the IEEE, 2019.

[InternetofSkills]

Oppici, L. and et. al., "Internet of Skills", In Tactile Internet, Academic Press pp. 75-99, 2021.

[ITU-NET2030]

ITU Network 2030 Technical Report, "Network 2030 - Gap analysis of Network 2030 new services, capabilities and use cases", 2020,
<<https://www.itu.int/pub/T-FG-NET2030-2020-1>>.

[ITU-T2014]

ITU-T Technology Watch Report, "The Tactile Internet", 2014, <https://www.itu.int/dms_pub/itu-t/oth/23/01/T23010000230001PDFE.pdf>.

[monaICC2020]

Ghassemian, M. and et. al., "Secure Non-Public Health Enterprise Networks", In 2020 IEEE International Conference on Communications Workshops (ICC Workshops), 2020.

[na2020simulation]

Na, W. and et. al., "Simulation and measurement: Feasibility study of Tactile Internet applications for mmWave virtual reality", In ETRI Journal 42.2 (2020): 163-174, 2020.

[Promwongsa]

Promwongsa, N. and el. al., "A Comprehensive Survey of the Tactile Internet: State-of-the-Art and Research Directions", IEEE Communications Surveys and Tutorials IEEE, 2021,
<<https://ieeexplore.ieee.org/document/8542940>>.

[SurgicalTraining]

Spiedel, S. and et. al., "Surgical Assistance and Training", In Tactile Internet, Academic Press pp. 23-39, 2021.

[TactileMultimodal-3GPP]

3GPP TR 22.847, "Study on supporting tactile and multi-modality communication services", 2021,
<<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3848>>.

[TIAijaz2019]

Aijaz, A. and et. al., "The Tactile Internet for Industries: A Review", In Proceedings of the IEEE, 2019.

[URLLC-3GPP]

3GPP TR 23.725, "Study on enhancement of Ultra-Reliable Low-Latency Communication (URLLC) support in the 5G Core network (5GC)", 2019,
<<https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3453>>.

Authors' Addresses

Chathura Sarathchandra
InterDigital Europe, Ltd.
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: chathura.sarathchandra@interdigital.com

Morteza Kheirkhah
InterDigital Europe, Ltd.
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: morteza.kheirkhah@interdigital.com

Mona Ghassemian
InterDigital Europe, Ltd.
64 Great Eastern Street, 1st Floor
London EC2A 3QR
United Kingdom

Email: mona.ghassemian@interdigital.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 27 October 2022

F. L. Templin, Ed.
The Boeing Company
25 April 2022

Transmission of IP Packets over Overlay Multilink Network (OMNI)
Interfaces
draft-templin-6man-omni-61

Abstract

Mobile nodes (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, space systems, enterprise wireless devices, pedestrians with cell phones, etc.) communicate with networked correspondents over multiple access network data links and configure mobile routers to connect end user networks. A multilink virtual interface specification is presented that enables mobile nodes to coordinate with a network-based mobility service and/or with other mobile node peers. The virtual interface provides an adaptation layer service that also applies for more static deployments such as enterprise and home networks. This document specifies the transmission of IP packets over Overlay Multilink Network (OMNI) Interfaces.

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 <https://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 27 October 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	4
2. Terminology	7
3. Requirements	15
4. Overlay Multilink Network (OMNI) Interface Model	15
5. OMNI Interface Maximum Transmission Unit (MTU)	22
5.1. Jumbograms	23
5.2. IPv6 Parcels	24
6. The OMNI Adaptation Layer (OAL)	24
6.1. OAL Source Encapsulation and Fragmentation	25
6.2. OAL L2 Encapsulation and Re-Encapsulation	30
6.3. OAL L2 Decapsulation and Reassembly	33
6.4. OAL Header Compression	34
6.5. OAL-in-OAL Encapsulation	38
6.6. OAL Identification Window Maintenance	40
6.7. OAL Fragment Retransmission	45
6.8. OAL MTU Feedback Messaging	46
6.9. OAL Super-Packets	48
6.10. OAL Bubbles	49
6.11. OAL Requirements	50
6.12. OAL Fragmentation Security Implications	51
6.13. OMNI Hosts	52
6.14. IP Parcels	55
7. Frame Format	58
8. Link-Local Addresses (LLAs)	59
9. Unique-Local Addresses (ULAs)	60
10. Global Unicast Addresses (GUAs)	63
11. Node Identification	64
12. Address Mapping - Unicast	65
12.1. The OMNI Option	66
12.2. OMNI Sub-Options	66
12.2.1. Pad1	69
12.2.2. PadN	69
12.2.3. Neighbor Coordination	70
12.2.4. Interface Attributes	72
12.2.5. Multilink Forwarding Parameters	75
12.2.6. Traffic Selector	80
12.2.7. Geo Coordinates	81

12.2.8.	Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Message	82
12.2.9.	Host Identity Protocol (HIP) Message	83
12.2.10.	PIM-SM Message	85
12.2.11.	Fragmentation Report (FRAGREP)	86
12.2.12.	Node Identification	87
12.2.13.	ICMPv6 Error	89
12.2.14.	QUIC-TLS Message	90
12.2.15.	Proxy/Server Departure	90
12.2.16.	Sub-Type Extension	91
13.	Address Mapping - Multicast	94
14.	Multilink Conceptual Sending Algorithm	95
14.1.	Multiple OMNI Interfaces	95
14.2.	Client-Proxy/Server Loop Prevention	96
15.	Router Discovery and Prefix Registration	96
15.1.	Window Synchronization	105
15.2.	Router Discovery in IP Multihop and IPv4-Only Networks	106
15.3.	DHCPv6-based Prefix Registration	108
15.4.	OMNI Link Extension	110
16.	Secure Redirection	110
17.	Proxy/Server Resilience	111
18.	Detecting and Responding to Proxy/Server Failures	111
19.	Transition Considerations	112
20.	OMNI Interfaces on Open Internetworks	113
21.	Time-Varying MNPs	115
22.	(H)HITs and Temporary ULA (TLA)s	116
23.	Address Selection	117
24.	Error Messages	118
25.	IANA Considerations	118
25.1.	"Protocol Numbers" Registry	118
25.2.	"IEEE 802 Numbers" Registry	118
25.3.	"IPv4 Special-Purpose Address" Registry	118
25.4.	"IPv6 Neighbor Discovery Option Formats" Registry	119
25.5.	"Ethernet Numbers" Registry	119
25.6.	"ICMPv6 Code Fields: Type 2 - Packet Too Big" Registry	119
25.7.	"OMNI Option Sub-Type Values" (New Registry)	119
25.8.	"OMNI Geo Coordinates Type Values" (New Registry)	120
25.9.	"OMNI Node Identification ID-Type Values" (New Registry)	120
25.10.	"OMNI Option Sub-Type Extension Values" (New Registry)	121
25.11.	"OMNI RFC4380 UDP/IP Header Option" (New Registry)	121
25.12.	"OMNI RFC6081 UDP/IP Trailer Option" (New Registry)	122
25.13.	Additional Considerations	122
26.	Security Considerations	123
27.	Implementation Status	124

28. Document Updates	124
29. Acknowledgements	124
30. References	126
30.1. Normative References	126
30.2. Informative References	128
Appendix A. OAL Checksum Algorithm	137
Appendix B. IPv6 ND Message Authentication and Integrity	137
Appendix C. VDL Mode 2 Considerations	138
Appendix D. Client-Proxy/Server Isolation Through Link-Layer Address Mapping	139
Appendix E. Change Log	140
Author's Address	140

1. Introduction

Mobile nodes (e.g., aircraft of various configurations, terrestrial vehicles, seagoing vessels, space systems, enterprise wireless devices, pedestrians with cellphones, etc.) configure mobile routers with multiple interface connections to wireless and/or wired-line data links. These data links may have diverse performance, cost and availability properties that can change dynamically according to mobility patterns, flight phases, proximity to infrastructure, etc. The mobile router acts as a Client of a network-based Mobility Service (MS) by configuring a virtual interface over its underlay interface data link connections to support the "6M's of modern Internetworking" (see below).

Each Client configures a virtual interface (termed the "Overlay Multilink Network Interface (OMNI)") as a thin layer over its underlay network interfaces (which may themselves connect to virtual or physical links). The OMNI interface is therefore the only interface abstraction exposed to the IP layer and behaves according to the Non-Broadcast, Multiple Access (NBMA) interface principle, while underlay interfaces appear as link layer communication channels in the architecture. The OMNI interface internally employs the "OMNI Adaptation Layer (OAL)" to ensure that original IP packets are adapted to diverse underlay interfaces with heterogeneous properties. The OMNI interface connects to a virtual overlay known as the "OMNI link". The OMNI link multinet service spans one or more Internetworks that may include private-use infrastructures (e.g., enterprise networks) and/or the global public Internet itself.

Client OMNI interfaces interact with the MS and/or other OMNI nodes through IPv6 Neighbor Discovery (ND) control message exchanges [RFC4861]. The MS consists of a distributed set of service nodes (including Proxy/Servers and other infrastructure elements) that also configure OMNI interfaces. Automatic Extended Route Optimization (AERO) in particular provides a companion MS compatible with the OMNI

architecture [I-D.templin-6man-aero]. AERO discusses details of ND message based route optimization, mobility management, and multinet traversal while the fundamental aspects of OMNI link operation are discussed in this document.

Each OMNI interface provides a multilink nexus for exchanging inbound and outbound traffic via selected underlay interface(s). The IP layer sees the OMNI interface as a point of connection to the OMNI link. Each OMNI link has one or more associated Mobility Service Prefixes (MSPs), which are typically IP Global Unicast Address (GUA) prefixes assigned to the link and from which Mobile Network Prefixes (MNPs) are derived. If there are multiple OMNI links, the IP layer will see multiple OMNI interfaces.

Each Client receives an MNP through IPv6 ND control message exchanges with Proxy/Servers over Access Networks (ANETs) and/or open Internetworks (INETs). The Client sub-delegates the MNP to downstream-attached End-user Networks (ENETs) independently of the underlay interfaces selected for data transport. The Client acts as a fixed or mobile router on behalf of peers on its ENETs, and uses OMNI interface control messaging to coordinate with Hosts, Proxy/Servers and/or other Clients. The Client iterates its control messaging over each of the OMNI interface's ANET/INET underlay interfaces in order to register each interface with the MS (see Section 15). The Client can also provide Proxy/Server-like services for a recursively nested chain of other Clients located in downstream-attached ENETs.

Clients may connect to multiple distinct OMNI links within the same OMNI domain by configuring multiple OMNI interfaces, e.g., omni0, omni1, omni2, etc. Each OMNI interface is configured over a set of underlay interfaces and provides a nexus for Safety-Based Multilink (SBM) operation. The IP layer applies SBM routing to select a specific OMNI interface, then the selected OMNI interface applies Performance-Based Multilink (PBM) internally to select appropriate underlay interfaces. Applications select SBM topologies based on IP layer Segment Routing [RFC8402], while each OMNI interface orchestrates PBM internally based on OMNI layer Segment Routing.

OMNI provides a link model suitable for a wide range of use cases. For example, the International Civil Aviation Organization (ICAO) Working Group-I Mobility Subgroup is developing a future Aeronautical Telecommunications Network with Internet Protocol Services (ATN/IPS) and has issued a liaison statement requesting IETF adoption [ATN] in support of ICAO Document 9896 [ATN-IPS]. The IETF IP Wireless Access in Vehicular Environments (ipwave) working group has further included problem statement and use case analysis for OMNI in a document now in AD evaluation for RFC publication

[I-D.ietf-ipwave-vehicular-networking]. Still other communities of interest include AEEC, RTCA Special Committee 228 (SC-228) and NASA programs that examine commercial aviation, Urban Air Mobility (UAM) and Unmanned Air Systems (UAS). Pedestrians with handheld devices represent another large class of potential OMNI users.

OMNI supports the "6M's of modern Internetworking" including:

1. Multilink - a Client's ability to coordinate multiple diverse underlay interfaces as a single logical unit (i.e., the OMNI interface) to achieve the required communications performance and reliability objectives.
2. Multinet - the ability to span the OMNI link over a segment routing topology with multiple diverse administrative domain network segments while maintaining seamless end-to-end communications between mobile Clients and correspondents such as air traffic controllers, fleet administrators, etc.
3. Mobility - a Client's ability to change network points of attachment (e.g., moving between wireless base stations) which may result in an underlay interface address change, but without disruptions to ongoing communication sessions with peers over the OMNI link.
4. Multicast - the ability to send a single network transmission that reaches multiple Clients belonging to the same interest group, but without disturbing other Clients not subscribed to the interest group.
5. Multihop - a mobile Client vehicle-to-vehicle relaying capability useful when multiple forwarding hops between vehicles may be necessary to "reach back" to an infrastructure access point connection to the OMNI link.
6. MTU assurance - the ability to deliver packets of various robust sizes between peers without loss due to a link size restriction, and to dynamically adjust packets sizes to achieve the optimal performance for each independent traffic flow.

This document specifies the transmission of IP packets and control messages over OMNI interfaces. The operation of both IP protocol versions (i.e., IPv4 [RFC0791] and IPv6 [RFC8200]) is specified as the network layer data plane, while OMNI interfaces use IPv6 ND messaging in the control plane independently of the data plane protocol(s). OMNI interfaces also provide an OAL based on encapsulation and fragmentation over heterogeneous underlay interfaces as an adaptation sublayer between L3 and L2. Both OMNI and the OAL are specified in detail throughout the remainder of this document.

2. Terminology

The terminology in the normative references applies; especially, the terms "link" and "interface" are the same as defined in the IPv6 [RFC8200] and IPv6 Neighbor Discovery (ND) [RFC4861] specifications. Additionally, this document assumes the following IPv6 ND message types: Router Solicitation (RS), Router Advertisement (RA), Neighbor Solicitation (NS), Neighbor Advertisement (NA) and Redirect. Hosts, Clients and Proxy/Servers that implement IPv6 ND maintain per-neighbor state in Neighbor Cache Entries (NCEs). Each NCE is indexed by the neighbor's network layer address(es) while the neighbor's OAL encapsulation address provides context for Identification verification.

The Protocol Constants defined in Section 10 of [RFC4861] are used in their same format and meaning in this document. The terms "All-Routers multicast", "All-Nodes multicast" and "Subnet-Router anycast" are the same as defined in [RFC4291] (with Link-Local scope assumed).

The term "IP" is used to refer collectively to either Internet Protocol version (i.e., IPv4 [RFC0791] or IPv6 [RFC8200]) when a specification at the layer in question applies equally to either version.

The following terms are defined within the scope of this document:

L2

The Data Link layer in the OSI network model. Also known as "layer-2", "link-layer", "sub-IP layer", etc.

L3

The Network layer in the OSI network model. Also known as "layer-3", "IP layer", etc.

Adaptation layer

A mid-layer that adapts L3 to a diverse collection of L2 underlay interfaces and their encapsulations. (No layer number is

assigned, since numbering was an artifact of the legacy reference model that need not carry forward in the modern architecture.) The adaptation layer sees the upper layer as "L3" and sees all lower layer encapsulations as "L2 encapsulations", which may include UDP, IP and true link-layer (e.g., Ethernet, etc.) headers.

Access Network (ANET)

a connected network region (e.g., an aviation radio access network, satellite service provider network, cellular operator network, WiFi network, etc.) that connects Clients to the Mobility Service. Physical and/or data link level security is assumed, and sometimes referred to as "protected spectrum". Private enterprise networks and ground domain aviation service networks may provide multiple secured IP hops between the Client's point of connection and the nearest Proxy/Server.

Internetwork (INET)

a connected network region with a coherent IP addressing plan that provides transit forwarding services between ANETs and/or OMNI nodes that coordinate with the Mobility Service over unprotected media. Since physical and/or data link level security cannot always be assumed, security must be applied by upper layers if necessary. The global public Internet itself is an example.

End-user Network (ENET)

a simple or complex "downstream" network that travels with the Client as a single logical unit. The ENET could be as simple as a single link connecting a single Host, or as complex as a large network with many links, routers, bridges and Hosts. The ENET could also provide an "upstream" link in a recursively-descending chain of additional Clients and ENETs. In this way, an ENET of an upstream Client is seen as the ANET of a downstream Client.

{A,I,E}NET interface

a Client's attachment to a link in an {A,I,E}NET.

*NET

a "wildcard" term used when a given specification applies equally to both ANET/INET cases. From the Client's perspective, *NET interfaces are "upstream" interfaces that connect the Client to the Mobility Service, while ENET interfaces are "downstream" interfaces that the Client uses to connect downstream ENETs, Hosts and/or other Clients.

underlay interface

an ANET/INET/ENET interface over which an OMNI interface is configured. The OMNI interface is seen as a L3 interface by the

IP layer, and each underlay interface is seen as a L2 interface by the OMNI interface. The underlay interface either connects directly to the physical communications media or coordinates with another node where the physical media is hosted.

OMNI link

a Non-Broadcast, Multiple Access (NBMA) virtual overlay configured over one or more INETs and their connected ANETs/ENETs. An OMNI link may comprise multiple distinct "segments" joined by L2 forwarding devices the same as for any link; the addressing plans in each segment may be mutually exclusive and managed by different administrative entities. Proxy/Servers and other infrastructure elements extend the link to support communications between Clients as single-hop neighbors.

OMNI interface

a node's attachment to an OMNI link, and configured over one or more underlay interfaces. If there are multiple OMNI links in an OMNI domain, a separate OMNI interface is configured for each link. The OMNI interface configures a Maximum Transmission Unit (MTU) and a Maximum Reassembly Unit (MRU) the same as any interface.

OMNI Adaptation Layer (OAL)

an OMNI interface sublayer service that encapsulates original IP packets admitted into the interface in an IPv6 header and/or subjects them to fragmentation and reassembly. The OAL is also responsible for generating MTU-related control messages as necessary, and for providing addressing context for OMNI link SRT traversal. The OAL presents a new layer in the Internet architecture known simply as the "adaptation layer".

Host

an end user device that extends the OMNI link over an ENET interface serviced by a Client. (As an implementation matter, the Host either assigns the same IP address from the ENET (underlay) interface to an (overlay) OMNI interface, or configures an OMNI-like function as a virtual sublayer of the ENET interface itself.) The IP addresses assigned to each Host ENET interface remain stable even if the Client's upstream *NET interface connections change.

Client

a network platform/device mobile router that configures one or more OMNI interfaces over distinct sets of underlay interfaces grouped as logical OMNI link units. The Client coordinates with the Mobility Service via upstream networks over *NET interfaces, and provides Proxy/Server services for Hosts and other Clients on

ENET interface downstream networks. The Client's *NET interface addresses and performance characteristics may change over time (e.g., due to node mobility, link quality, etc.) while downstream-attached Hosts and other Clients see the ENET as a stable ANET.

Proxy/Server

a segment routing topology edge node that configures an OMNI interface and connects Clients to the Mobility Service. As a server, the Proxy/Server responds directly to some Client IPv6 ND messages. As a proxy, the Proxy/Server forwards other Client IPv6 ND messages to other Proxy/Servers and Clients. As a router, the Proxy/Server provides a forwarding service for ordinary data packets that may be essential in some environments and a last resort in others. Proxy/Servers at ANET boundaries configure both an ANET downstream interface and *NET upstream interface, while INET-based Proxy/Servers configure only an INET interface.

First-Hop Segment (FHS) Proxy/Server

a Proxy/Server connected to the source Client's *NET that forwards packets sent by the source into the segment routing topology. FHS Proxy/Servers also act as intermediate forwarding nodes to facilitate RS/RA exchanges between Clients and Hub Proxy/Servers.

Last-Hop Segment (LHS) Proxy/Server

a Proxy/Server connected to the target Client's *NET that forwards packets received from the segment routing topology to the target.

Hub Proxy/Server

a single Proxy/Server selected by the Client that provides a designated router service for all of the Client's *NET underlay networks. Since all Proxy/Servers provide equivalent services, Clients normally select the first FHS Proxy/Server they coordinate with to serve as the Hub. However, the Hub can also be any available Proxy/Server for the OMNI link, i.e., and not necessarily one of the Client's FHS Proxy/Servers.

Segment Routing Topology (SRT)

a multinet forwarding region configured over one or more INETs between the FHS Proxy/Server and LHS Proxy/Server. The SRT spans the OMNI link on behalf of source/target Client pairs using segment routing in a manner outside the scope of this document (see: [I-D.templin-6man-aero]).

Mobility Service (MS)

a mobile routing service that tracks Client movements and ensures that Clients remain continuously reachable even across mobility events. The MS consists of the set of all Proxy/Servers and any other OMNI link supporting infrastructure nodes. Specific MS details are out of scope for this document, with an example found in [I-D.templin-6man-aero].

Mobility Service Prefix (MSP)

an aggregated IP Global Unicast Address (GUA) prefix (e.g., 2001:db8::/32, 192.0.2.0/24, etc.) assigned to the OMNI link and from which more-specific Mobile Network Prefixes (MNPs) are delegated. OMNI link administrators typically obtain MSPs from an Internet address registry, however private-use prefixes can also be used subject to certain limitations (see: Section 10). OMNI links that connect to the global Internet advertise their MSPs to their interdomain routing peers.

Mobile Network Prefix (MNP)

a longer IP prefix delegated from an MSP (e.g., 2001:db8:1000:2000::/56, 192.0.2.8/30, etc.) and assigned to a Client. Clients receive MNPs from Proxy/Servers and sub-delegate them to routers, Hosts and other Clients located in ENETs.

original IP packet

a whole IP packet or fragment admitted into the OMNI interface by the network layer prior to OAL encapsulation and fragmentation, or an IP packet delivered to the network layer by the OMNI interface following OAL decapsulation and reassembly.

OAL packet

an original IP packet encapsulated in an IPv6 header (i.e., the OAL header) then submitted for OAL fragmentation and reassembly.

OAL fragment

a portion of an OAL packet following fragmentation but prior to encapsulation, or following encapsulation but prior to OAL reassembly.

(OAL) atomic fragment

an OAL packet that does not require fragmentation is always encapsulated as an "atomic fragment" with a Fragment Header with Fragment Offset and More Fragments both set to 0, but with a valid Identification value.

(OAL) carrier packet

an encapsulated OAL fragment following L2 encapsulation or prior to L2 decapsulation. OAL sources and destinations exchange

carrier packets over underlay interfaces, and may be separated by one or more OAL intermediate nodes. OAL intermediate nodes may perform re-encapsulation on carrier packets by removing the L2 headers of the first hop network and replacing them with new L2 headers for the next hop network. (The term "carrier" honors agents of the service postulated by [RFC1149] and [RFC6214].)

OAL source

an OMNI interface acts as an OAL source when it encapsulates original IP packets to form OAL packets, then performs OAL fragmentation and encapsulation to create carrier packets.

OAL destination

an OMNI interface acts as an OAL destination when it decapsulates carrier packets, then performs OAL reassembly and decapsulation to derive the original IP packet.

OAL intermediate node

an OMNI interface acts as an OAL intermediate node when it removes the L2 encapsulation headers of carrier packets received on a first segment, then re-encapsulates the carrier packets in new L2 encapsulation headers and forwards them into the next segment.

OMNI Option

an IPv6 Neighbor Discovery option providing multilink parameters for the OMNI interface as specified in Section 12.

Interface Identifier (IID)

the least significant 64 bits of an IPv6 address, as specified in the IPv6 addressing architecture [RFC4291].

Link Local Address (LLA)

an IPv6 address beginning with fe80::/64 per the IPv6 addressing architecture [RFC4291] and with either a 64-bit MNP (LLA-MNP) or a 56-bit random value (LLA-RND) encoded in the IID as specified in Section 8.

Unique Local Address (ULA)

an IPv6 address beginning with fd00::/8 followed by a 40-bit Global ID followed by a 16-bit Subnet ID per [RFC4193] and with either a 64-bit MNP (ULA-MNP) or a 56-bit random value (ULA-RND) encoded in the IID as specified in Section 9. (Note that [RFC4193] specifies a second form of ULAs based on the prefix fc00::/8, which are referred to as "ULA-C" throughout this document to distinguish them from the ULAs defined here.)

Temporary Local Address (TLA)

a ULA beginning with fd00::/16 followed by a 48-bit randomly-initialized value followed by an MNP-based (TLA-MNP) or random (TLA-RND) IID as specified in Section 9. Clients use TLAs to bootstrap autoconfiguration in the presence of OMNI link infrastructure or for sustained communications in the absence of infrastructure. (Note that in some environments Clients can instead use a (Hierarchical) Host Identity Tag ((H)HIT) instead of a TLA - see: Section 22.)

eXtended Local Address (XLA)

a TLA beginning with fd00::/64 followed by an MNP-based (XLA-MNP) or random (XLA-RND) IID as specified in Section 9. An XLA is simply a TLA with an all-0 48-bit value following fd00::/16, and can be used to supply a "wildcard match" for IPv6 ND cache entries, a routing table entry for the OMNI link routing system, etc. (Note that XLAs can also be statelessly formed from LLAs (and vice-versa) simply by inverting prefix bits 7 and 8.)

Multilink

a Client OMNI interface's manner of managing multiple diverse *NET underlay interfaces as a single logical unit. The OMNI interface provides a single unified interface to upper layers, while underlay interface selections are performed on a per-packet basis considering traffic selectors such as DSCP, flow label, application policy, signal quality, cost, etc. Multilink selections are coordinated in both the outbound and inbound directions based on source/target underlay interface pairs.

Multinet

an intermediate node's manner of spanning multiple diverse IP Internetwork and/or private enterprise network "segments" at the OAL layer below IP. Through intermediate node concatenation of SRT network segments, multiple diverse Internetworks (such as the global public IPv4 and IPv6 Internets) can serve as transit segments in an end-to-end L2 forwarding path. This OAL concatenation capability provides benefits such as supporting IPv4/IPv6 transition and coexistence, joining multiple diverse operator networks into a cooperative single service network, etc. See: [I-D.templin-6man-aero] for further information.

Multihop

an iterative relaying of IP packets between Client's over an OMNI underlay interface technology (such as omnidirectional wireless) without support of fixed infrastructure. Multihop services entail Client-to-Client relaying within a Mobile/Vehicular Ad-hoc Network (MANET/VANET) for Vehicle-to-Vehicle (V2V) communications and/or for Vehicle-to-Infrastructure (V2I) "range extension" where Clients within range of communications infrastructure elements provide forwarding services for other Clients.

Mobility

any action that results in a change to a Client underlay interface address. The change could be due to, e.g., a handover to a new wireless base station, loss of link due to signal fading, an actual physical node movement, etc.

Safety-Based Multilink (SBM)

A means for ensuring fault tolerance through redundancy by connecting multiple OMNI interfaces within the same domain to independent routing topologies (i.e., multiple independent OMNI links).

Performance Based Multilink (PBM)

A means for selecting one or more underlay interface(s) for packet transmission and reception within a single OMNI interface.

OMNI Domain

The set of all SBM/PBM OMNI links that collectively provides services for a common set of MSPs. All OMNI links within the same domain configure, advertise and respond to the same OMNI IPv6 Anycast address(es).

Multilink Forwarding Information Base (MFIB)

A forwarding table on each OMNI source, destination and intermediate node that includes Multilink Forwarding Vectors (MFV) with both next hop forwarding instructions and context for reconstructing compressed headers for specific underlay interface pairs used to communicate with peers. See: [I-D.templin-6man-aero] for further discussion.

Multilink Forwarding Vector (MFV)

An MFIB entry that includes soft state for each underlay interface pairwise communication session between peers. MFVs are identified by both a next-hop and previous-hop MFV Index (MFVI), with the next-hop established based on an IPv6 ND solicitation and the previous hop established based on the solicited IPv6 ND advertisement response. See: [I-D.templin-6man-aero] for further discussion.

Multilink Forwarding Vector Index (MFVI)

A 4 octet value selected by an OMNI node when it creates an MFV, then advertised to either a next-hop or previous-hop. OMNI intermediate nodes assign two distinct MFVIs for each MFV and advertise one to the next-hop and the other to the previous-hop. OMNI end systems assign and advertise a single MFVI. See: [I-D.templin-6man-aero] for further discussion.

IP Jumbogram

an IPv4 or IPv6 packet with a Jumbo Payload option that includes a 32-bit length field to be used instead of the 16-bit {Total, Payload} Length field (see: Section 5.1). For IPv4, the Total Length field must be set to the length of the IPv4 header only. For IPv6, the Payload Length must be set to 0.

IP Parcel

a special form of an IP Jumbogram with a segment length value included in the {Total, Payload} Length field and also with a Jumbo Payload option (see: Section 5.2).

INADDR

the IP address (and also the UDP port number when UDP is used) that appears in (L2) encapsulation headers in the data plane and in IPv6 ND OMNI option sub-options in the control plane.

3. Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

An implementation is not required to internally use the architectural constructs described here so long as its external behavior is consistent with that described in this document.

4. Overlay Multilink Network (OMNI) Interface Model

An OMNI interface is a virtual interface configured over one or more underlay interfaces, which may be physical (e.g., an aeronautical radio link, etc.) or virtual (e.g., an Internet or higher-layer "tunnel"). The OMNI interface architectural layering model is the same as in [RFC5558][RFC7847], and augmented as shown in Figure 1. The IP layer therefore sees the OMNI interface as a single L3 interface nexus for multiple underlay interfaces that appear as L2 communication channels in the architecture.

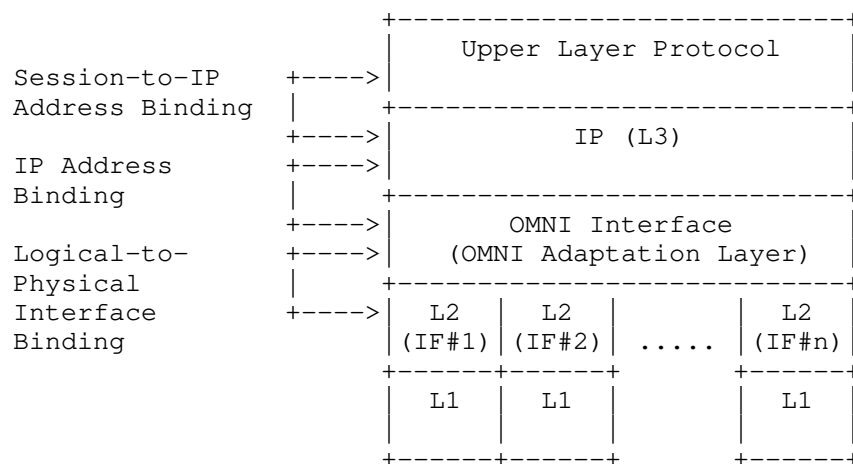


Figure 1: OMNI Interface Architectural Layering Model

Each underlay interface provides an L2/L1 abstraction according to one of the following models:

- * ANET interfaces connect to a protected and secured ANET that is separated from the open INET by Proxy/Servers. The ANET interface may be either on the same L2 link segment as a Proxy/Server, or separated from a Proxy/Server by multiple IP hops. (Note that NATs may appear internally within an ANET or on the Proxy/Server itself and may require NAT traversal the same as for the INET case.)
- * INET interfaces connect to an INET either natively or through one or several IPv4 Network Address Translators (NATs). Native INET interfaces have global IP addresses that are reachable from any INET correspondent. NATed INET interfaces typically configure private IP addresses and connect to a private network behind one or more NATs with the outermost NAT providing INET access.
- * ENET interfaces connect a Client's downstream-attached networks, where the Client provides forwarding services for ENET Host and Client communications to remote peers. An ENET may be as simple as a small stub network that travels with a mobile Client (e.g., an Internet-of-Things) to as complex as a large private enterprise network that the Client connects to a larger ANET or INET. Downstream-attached Hosts and Clients see the ENET as an ANET and see the (upstream) Client as a Proxy/Server.

- * VPNed interfaces use security encapsulation over an underlay network to a Client or Proxy/Server acting as a Virtual Private Network (VPN) gateway. Other than the link-layer encapsulation format, VPNed interfaces behave the same as for Direct interfaces.
- * Direct (aka "point-to-point") interfaces connect directly to a Client or Proxy/Server without crossing any networked paths. An example is a line-of-sight link between a remote pilot and an unmanned aircraft.

The OMNI interface forwards original IP packets from the network layer (L3) using the OMNI Adaptation Layer (OAL) (see: Section 5) as an encapsulation and fragmentation sublayer service. This "OAL source" then further encapsulates the resulting OAL packets/fragments in underlay network headers (e.g., UDP/IP, IP-only, Ethernet-only, etc.) to create L2-encapsulated "carrier packets" for transmission over underlay interfaces. The target OMNI interface receives the carrier packets from underlay interfaces and discards the L2 encapsulation headers. If the resulting OAL packets/fragments are addressed to itself, the OMNI interface acts as an "OAL destination" and performs reassembly if necessary, discards the OAL encapsulation, and delivers the original IP packet to the network layer. If the OAL fragments are addressed to another node, the OMNI interface instead acts as an "OAL intermediate node" by re-encapsulating the carrier packets in new underlay network L2 headers and forwarding them over an underlay interface without reassembling or discarding the OAL encapsulation. The OAL source and OAL destination are seen as "neighbors" on the OMNI link, while OAL intermediate nodes provide a virtual bridging service that joins the segments of a (multinet) Segment Routing Topology (SRT).

The OMNI interface can forward original IP packets over underlay interfaces while including/omitting various lower layer encapsulations including OAL, UDP, IP and Ethernet (ETH) or other link-layer header. The network layer can also access the underlay interfaces directly while bypassing the OMNI interface entirely when necessary. This architectural flexibility may be beneficial for underlay interfaces (e.g., some aviation data links) for which encapsulation overhead may be a primary consideration. OMNI interfaces that send original IP packets directly over underlay interfaces without invoking the OAL can only reach peers located on the same OMNI link segment. Source Clients can instead use the OAL to coordinate with target Clients in the same or different OMNI link segments by sending initial carrier packets to a First-Hop Segment (FHS) Proxy/Server. The FHS Proxy/Server then forwards the packets into the SRT spanning tree, which transports them to a Last-Hop Segment (LHS) Proxy/Server for the target Client.

Original IP packets sent directly over underlay interfaces are subject to the same path MTU related issues as for any Internetworking path, and do not include per-packet identifications that can be used for data origin verification and/or link-layer retransmissions. Original IP packets presented directly to an underlay interface that exceed the underlay network path MTU are dropped with an ordinary ICMPv6 Packet Too Big (PTB) message returned. These PTB messages are subject to loss [RFC2923] the same as for any non-OMNI IP interface.

The OMNI interface encapsulation/decapsulation layering possibilities are shown in Figure 2 below. Imaginary vertical lines drawn between the Network Layer and Underlay interfaces in the figure denote the encapsulation/decapsulation layering combinations possible. Common combinations include IP-only (i.e., direct access to underlay interfaces with or without using the OMNI interface), IP/IP, IP/UDP/IP, IP/UDP/IP/ETH(ERNET), IP/OAL/UDP/IP, IP/OAL/UDP/ETH, etc.

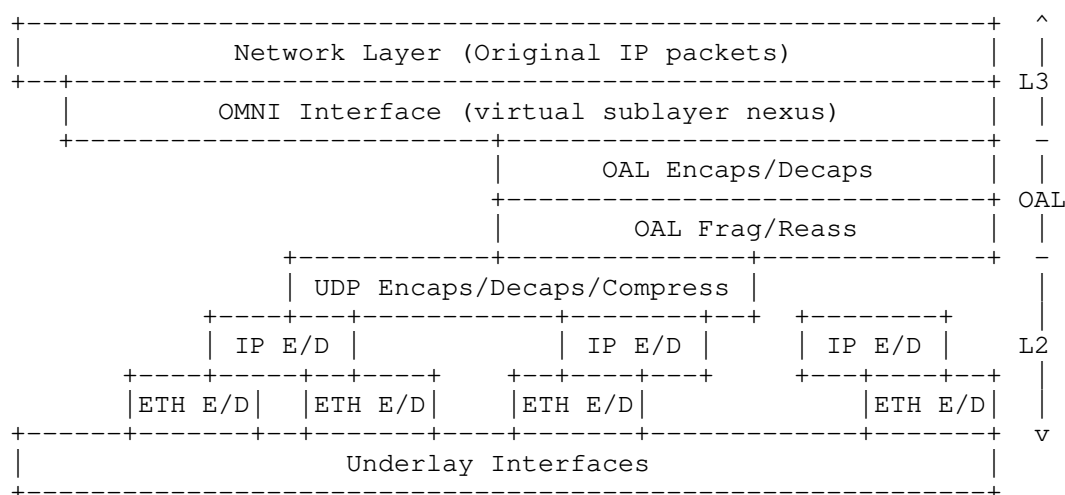


Figure 2: OMNI Interface Layering

The OMNI/OAL model gives rise to a number of opportunities:

- * Clients receive MNPs from the MS, and coordinate with the MS through IPv6 ND message exchanges with Proxy/Servers. Clients use the MNP to construct a unique Link-Local Address (LLA-MNP) through the algorithmic derivation specified in Section 8 and assign the LLA to the OMNI interface. Since LLA-MNPs are uniquely derived from an MNP, no Duplicate Address Detection (DAD) or Multicast Listener Discovery (MLD) messaging is necessary.

- * since Temporary ULAs with random IIDs (TLA-RNDs) are statistically unique, they can be used without DAD until an MNP is obtained.
- * underlay interfaces on the same L2 link segment as a Proxy/Server do not require any L3 addresses (i.e., not even link-local) in environments where communications are coordinated entirely over the OMNI interface.
- * as underlay interface properties change (e.g., link quality, cost, availability, etc.), any active interface can be used to update the profiles of multiple additional interfaces in a single message. This allows for timely adaptation and service continuity under dynamically changing conditions.
- * coordinating underlay interfaces in this way allows them to be represented in a unified MS profile with provisions for mobility and multilink operations.
- * exposing a single virtual interface abstraction to the IPv6 layer allows for multilink operation (including QoS based link selection, packet replication, load balancing, etc.) at L2 while still permitting L3 traffic shaping based on, e.g., DSCP, flow label, etc.
- * the OMNI interface allows multinet traversal over the SRT when communications across different administrative domain network segments are necessary. This mode of operation would not be possible via direct communications over the underlay interfaces themselves.
- * the OAL supports lossless and adaptive path MTU mitigations not available for communications directly over the underlay interfaces themselves. The OAL supports "packing" of multiple IP payload packets within a single OAL "super-packet" and also supports transmission of IP packets and parcels of all sizes up to and including Jumbograms.
- * the OAL applies per-packet identification values that allow for link-layer reliability and data origin authentication.
- * L3 sees the OMNI interface as a point of connection to the OMNI link; if there are multiple OMNI links, L3 will see multiple OMNI interfaces.
- * Multiple independent OMNI interfaces can be used for increased fault tolerance through Safety-Based Multilink (SBM), with Performance-Based Multilink (PBM) applied within each interface.

- * Multiple independent OMNI links can be joined together into a single link without requiring renumbering of infrastructure elements, since the ULAs assigned to the different links will be mutually exclusive.
- * the OMNI/OAL model supports transmission of a new form of IP packets known as "IP Parcels" that improve performance and efficiency for both upper layer protocols and networked paths.

Note that even when the OMNI virtual interface is present, applications can still access underlay interfaces either through the network protocol stack using an Internet socket or directly using a raw socket. This allows for intra-network (or point-to-point) communications without invoking the OMNI interface and/or OAL. For example, when an OMNI interface is configured over an underlay IP interface, applications can still invoke intra-network IP communications directly over the underlay interface as long as the communicating endpoints are not subject to mobility dynamics.

Figure 3 depicts the architectural model for a source Client with an attached ENET connecting to the OMNI link via multiple independent ANETs/INETs (i.e., *NETs). The Client's OMNI interface sends IPv6 ND solicitation messages over available *NET underlay interfaces using any necessary L2 encapsulations. The IPv6 ND messages traverse the *NETs until they reach an FHS Proxy/Server (FHS#1, FHS#2, ..., FHS#n), which returns an IPv6 ND advertisement message and/or forwards a proxied version of the message over the SRT to an LHS Proxy/Server near the target Client (LHS#1, LHS#2, ..., LHS#m). The Hop Limit in IPv6 ND messages is not decremented due to encapsulation; hence, the source and target Client OMNI interfaces appear to be attached to a common link.

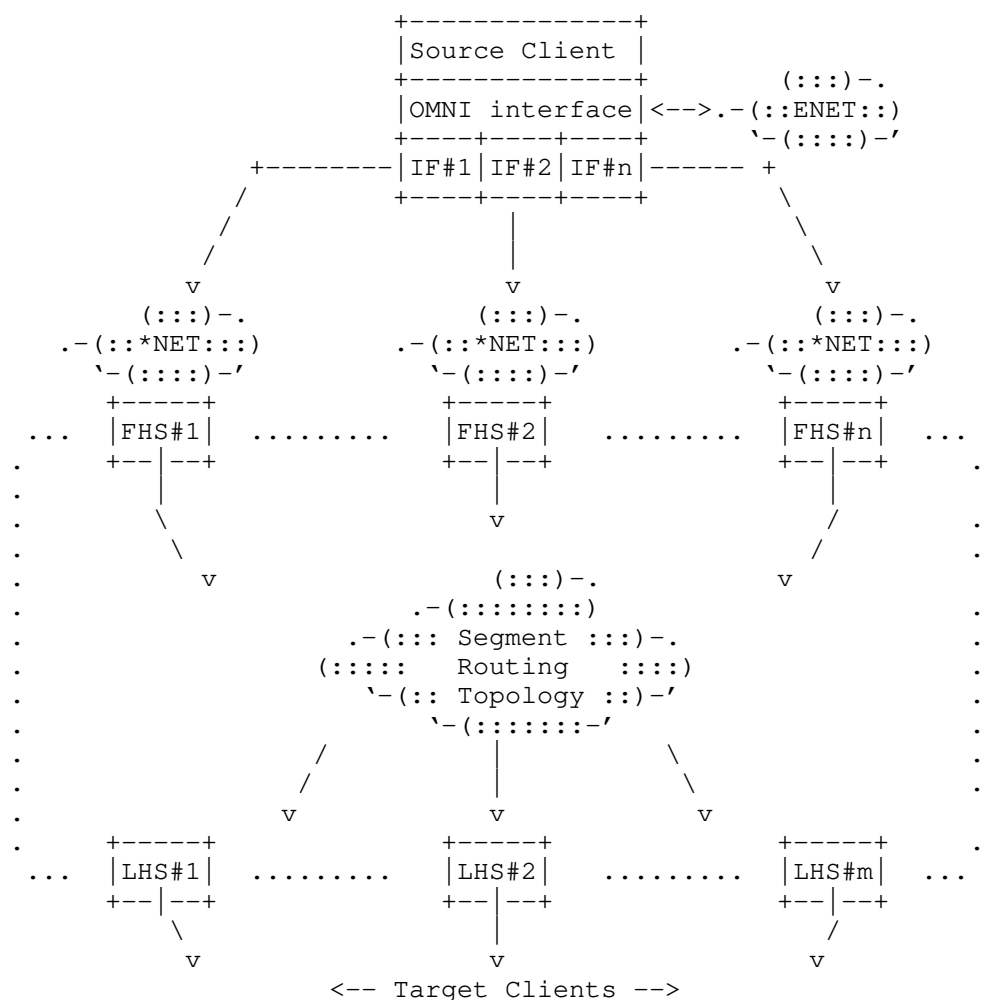


Figure 3: Source/Target Client Coordination over the OMNI Link

After the initial IPv6 ND message exchange, the source Client (as well as any nodes on its attached ENETs) can send packets to the target Client over the OMNI interface. OMNI interface multilink services will forward the packets via FHS Proxy/Servers for the correct underlay *NETs. The FHS Proxy/Server then forwards the packets over the SRT which delivers them to an LHS Proxy/Server, and the LHS Proxy/Server in turn forwards them to the target Client. (Note that when the source and target Client are on the same SRT segment, the FHS and LHS Proxy/Servers may be one and the same.)

Clients select a Hub Proxy/Server (not shown in the figure), which will often be one of their FHS Proxy/Servers but could also be any Proxy/Server on the OMNI link. Clients then register all of their *NET underlay interfaces with the Hub Proxy/Server via the FHS Proxy/Server in a pure proxy role. The Hub Proxy/Server then provides a designated router service for the Client, and the Client can quickly migrate to a new Hub Proxy/Server if the first becomes unresponsive.

Clients therefore use Proxy/Servers as gateways into the SRT to reach OMNI link correspondents via a spanning tree established in a manner outside the scope of this document. Proxy/Servers forward critical MS control messages via the secured spanning tree and forward other messages via the unsecured spanning tree (see Security Considerations). When route optimization is applied as discussed in [I-D.templin-6man-aero], Clients can instead forward directly to SRT intermediate nodes (or directly to correspondents in the same SRT segment) to reduce Proxy/Server load.

Note: while not shown in the figure, a Client's ENET may connect many additional Hosts and even other Clients in a recursive extension of the OMNI link. This OMNI virtual link extension will be discussed more fully throughout the document.

5. OMNI Interface Maximum Transmission Unit (MTU)

The OMNI interface observes the link nature of tunnels, including the Maximum Transmission Unit (MTU), Maximum Reassembly Unit (MRU) and the role of fragmentation and reassembly [I-D.ietf-intarea-tunnels]. The OMNI interface is configured over one or more underlay interfaces as discussed in Section 4, where the interfaces (and their associated underlay network paths) may have diverse MTUs. OMNI interface considerations for accommodating original IP packets of various sizes are discussed in the following sections.

IPv6 underlay interfaces are REQUIRED to configure a minimum MTU of 1280 octets and a minimum MRU of 1500 octets [RFC8200]. Therefore, the minimum IPv6 path MTU is 1280 octets since routers on the path are not permitted to perform network fragmentation even though the destination is required to reassemble more. The network therefore MUST forward original IP packets of at least 1280 octets without generating an IPv6 Path MTU Discovery (PMTUD) Packet Too Big (PTB) message [RFC8201]. (While the source can apply "source fragmentation" for locally-generated IPv6 packets up to 1500 octets and larger still if it knows the destination configures a larger MRU, this does not affect the minimum IPv6 path MTU.)

IPv4 underlay interfaces are REQUIRED to configure a minimum MTU of 68 octets [RFC0791] and a minimum MRU of 576 octets [RFC0791][RFC1122]. Therefore, when the Don't Fragment (DF) bit in the IPv4 header is set to 0 the minimum IPv4 path MTU is 576 octets since routers on the path support network fragmentation and the destination is required to reassemble at least that much. The OMNI interface therefore MUST set DF to 0 in the IPv4 encapsulation headers of carrier packets that are no larger than 576 octets, and SHOULD set DF to 1 in larger carrier packets unless it has a way to determine the encapsulation destination MRU and has carefully considered the issues discussed in Section 6.12.

When the network layer admits an original IP packet into the OMNI interface the OAL prepends an IPv6 encapsulation header (see: Section 6) where the 16-bit Payload Length field limits the maximum-sized original IP packet to $(2^{16} - 1) = 65535$ octets; this is also the maximum size that the OAL can accommodate with IPv6 fragmentation. The OMNI interface therefore sets an MTU and MRU of 65535 octets to support assured delivery of original packets no larger than this size even if IPv6 fragmentation is required. (The OMNI interface MAY set a larger MTU to support best-effort delivery for larger packets; see below.) The OMNI interface then employs the OAL as an encapsulation sublayer service to transform original IP packets into OAL packets/fragments, and the OAL in turn uses underlay network encapsulation to forward carrier packets over underlay interfaces (see: Section 6).

5.1. Jumbograms

While the maximum-sized original IP packet that the OAL can accommodate using IPv6 fragmentation is 65535 octets, OMNI interfaces can forward still larger IPv6 packets as OAL "atomic fragments" through the application of IPv6 Jumbograms [RFC2675]. For such larger packets, the OMNI interface performs OAL encapsulation by appending an IPv6 header followed by an 8-octet Hop-By-Hop header with Jumbo Payload option followed by a Routing Header of no more than 40-octets (if necessary) and finally followed by an 8-octet Fragment Header.

Since the Jumbo Payload option includes a 32-bit length field, OMNI interfaces can therefore configure a larger IP MTU up to a maximum of $((2^{32} - 1) - 8 - 40 - 8) = 4294967239$ octets. In that case, the OAL will still provide original IP packets no larger than 65535 with an IPv6 fragmentation-based assured delivery service while larger IP packets will receive a best-effort delivery service as atomic fragments (note that the OAL destination is permitted to accept atomic fragments that exceed the OMNI interface MRU).

The OAL source forwards jumbo atomic fragments under the assumption that upper and lower layers will employ sufficient integrity assurance, noting that commonly-used 32-bit CRCs may be inadequate for these larger sizes [CRC]. If the packet is dropped along the path to the OAL destination, the OAL source must arrange to return a PTB "hard error" to the original source Section 6.8.

This document notes that a Jumbogram service for IPv4 is also specified in [I-D.templin-intarea-parcels], where all OMNI link aspects of the service are conducted in a similar fashion as for IPv6 above.

5.2. IPv6 Parcels

As specified in [I-D.templin-intarea-parcels], an IP Parcel is a variation of the IP Jumbogram construction beginning with an IP header with the length of the first upper layer protocol segment in the {Total, Payload} Length field, but with a Jumbo Payload option with a length that may be the same as or larger than the length in the IP header. The differences in these lengths determines the size and number of upper layer protocol segments within the parcel.

The IP Parcel format and transmission/reception procedures for OMNI interfaces are specified in Section 6.14. End systems that implement either the full OMNI interface (i.e., Clients) or enough of the OAL to process parcels (i.e., Hosts) are permitted to exchange parcels with consenting peers.

6. The OMNI Adaptation Layer (OAL)

When an OMNI interface forwards an original IP packet from the network layer for transmission over one or more underlay interfaces, the OMNI Adaptation Layer (OAL) acting as the OAL source applies encapsulation to form OAL packets subject to fragmentation producing OAL fragments suitable for L2 encapsulation and transmission as carrier packets over underlay interfaces as described in Section 6.1.

These carrier packets travel over one or more underlay networks spanned by OAL intermediate nodes in the SRT, which re-encapsulate by removing the L2 headers of the first underlay network and appending L2 headers appropriate for the next underlay network in succession. (This process supports the multinet concatenation capability needed for joining multiple diverse networks.) After re-encapsulation by zero or more OAL intermediate nodes, the carrier packets arrive at the OAL destination.

When the OAL destination receives the carrier packets, it discards the L2 headers and reassembles the resulting OAL fragments (if necessary) into an OAL packet as described in Section 6.3. The OAL destination next decapsulates the OAL packet to obtain the original IP packet then delivers the original IP packet to the network layer. The OAL source may be either the source Client or its FHS Proxy/Server, while the OAL destination may be either the LHS Proxy/Server or the target Client. Proxy/Servers (and SRT Gateways as discussed in [I-D.templin-6man-aero]) may also serve as OAL intermediate nodes.

The OAL presents an OMNI sublayer abstraction similar to ATM Adaptation Layer 5 (AAL5). Unlike AAL5 which performs segmentation and reassembly with fixed-length 53 octet cells over ATM networks, however, the OAL uses IPv6 encapsulation, fragmentation and reassembly with larger variable-length cells over heterogeneous underlay networks. Detailed operations of the OAL are specified in the following sections.

6.1. OAL Source Encapsulation and Fragmentation

When the network layer forwards an original IP packet into the OMNI interface, the OAL source creates an "OAL packet" by prepending an IPv6 OAL encapsulation header per [RFC2473] but does not decrement the Hop Limit/TTL of the original IP packet since encapsulation occurs at a layer below IP forwarding. The OAL source copies the "Type of Service/Traffic Class" [RFC2983] and "Explicit Congestion Notification (ECN)" [RFC3168] values in the original packet's IP header into the corresponding fields in the OAL header, then sets the OAL header "Flow Label" as specified in [RFC6438]. The OAL source finally sets the OAL header IPv6 Payload Length to the length of the original IP packet and sets Hop Limit to a value that MUST NOT be larger than 63 yet is still sufficiently large to enable loop-free forwarding over multiple concatenated OMNI link intermediate hops.

The OAL next selects OAL packet source and destination addresses. Client OMNI interfaces set the OAL source address to a Unique Local Address (ULA) based on the Mobile Network Prefix (ULA-MNP). When a Client OMNI interface does not (yet) have a ULA prefix and/or an MNP suffix, it can instead use a Temporary ULA (TLA) (or a (Hierarchical) Host Identity Tag ((H)HIT - see: Section 22) as an OAL address. Finally, when the Client needs to express its MNP outside the context of a specific ULA prefix, it can use an eXtended ULA (XLA). Proxy/Server OMNI interfaces instead set the source address to a Random ULA (ULA-RND) (see: Section 9), but also process packets with anycast and/or multicast OAL addresses that they are configured to recognize.)

The OAL source next selects a 32-bit OAL packet Identification value as specified in Section 6.6. The OAL then calculates a 2-octet OAL checksum using the algorithm specified in Appendix A. The OAL source calculates the checksum over the OAL packet beginning with a pseudo-header of the OAL header similar to that found in Section 8.1 of [RFC8200], then extending over the entire length of the original IP packet. The OAL pseudo-header is formed as shown in Figure 4:

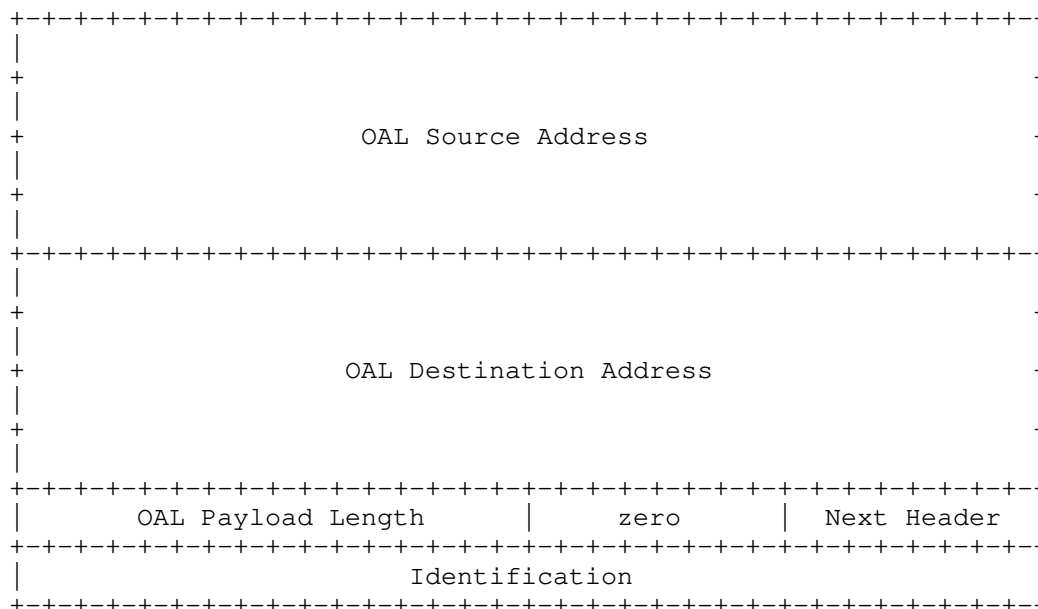


Figure 4: OAL Pseudo-Header

After calculating the checksum, the OAL source next fragments the OAL packet if necessary while assuming the IPv4 minimum path MTU (i.e., 576 octets) as the worst case for OAL fragmentation regardless of the underlay interface IP protocol version since IPv6/IPv4 protocol translation and/or IPv6-in-IPv4 encapsulation may occur in any underlay network path. By initially assuming the IPv4 minimum even for IPv6 underlay interfaces, the OAL source may produce smaller fragments with additional encapsulation overhead but avoids loss due to presenting an underlay interface with a carrier packet that exceeds its MRU. Additionally, the OAL path could traverse multiple SRT segments with intermediate OAL forwarding nodes performing re-encapsulation where the L2 encapsulation of the previous segment is replaced by the L2 encapsulation of the next segment which may be based on a different IP protocol version and/or encapsulation sizes.

The OAL source therefore assumes a default minimum path MTU of 576 octets at each SRT segment for the purpose of generating OAL fragments for L2 encapsulation and transmission as carrier packets. Each successive SRT intermediate node may include either a 20 octet IPv4 or 40 octet IPv6 header, an 8 octet UDP header and in some cases an IP security encapsulation (40 octets maximum assumed) during re-encapsulation. Intermediate nodes at any SRT segment may also insert or modify the Routing Header (40 octets maximum) following the 40 octet OAL IPv6 header and preceding the 8 octet Fragment Header. Therefore, assuming a worst case of $(40 + 40 + 8) = 88$ octets for L2 encapsulations plus $(40 + 40 + 8) = 88$ octets for OAL encapsulation leaves no less than $(576 - 88 - 88) = 400$ octets remaining to accommodate a portion of the original IP packet/fragment. The OAL source therefore sets a minimum Maximum Payload Size (MPS) of 400 octets as the basis for the minimum-sized OAL fragment that can be assured of traversing all SRT segments without loss due to an MTU/MRU restriction. The Maximum Fragment Size (MFS) for OAL fragmentation is therefore determined by the MPS plus the size of the OAL encapsulation headers.

The OAL source SHOULD maintain "path MPS" values for individual OAL destinations initialized to the minimum MPS and increased to larger values if better information is known or discovered. For example, when peers share a common underlay network link or a fixed path with a known larger MTU, the OAL source can set path MPS to a larger size (i.e., greater than 400 octets) as long as the peer reassembles before re-encapsulating and forwarding (while re-fragmenting if necessary). Also, if the OAL source has a way of knowing the maximum L2 encapsulation size for all SRT segments along the path it may be able to increase path MPS to reserve additional room for payload data. Even when OAL header compression is used, the OAL source must include the uncompressed OAL header size in its path MPS calculation since it may need to include a full header at any time.

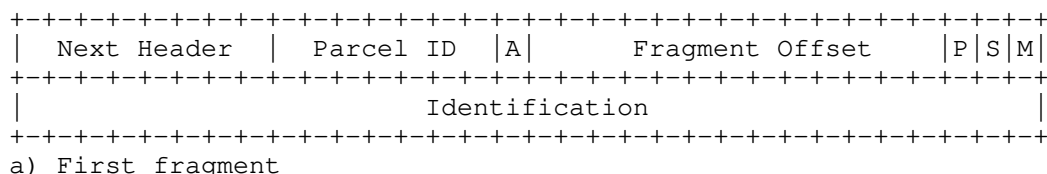
The OAL source can also optimistically set a larger path MPS and/or actively probe individual OAL destinations to discover larger sizes using packetization layer probes in a similar fashion as [RFC4821][RFC8899], but care must be taken to avoid setting static values for dynamically changing paths leading to black holes. The probe involves sending an OAL packet larger than the current path MPS and receiving a small acknowledgement response (with the possible receipt of link-layer error message when a probe is lost). For this purpose, the OAL source can send an NS message with one or more OMNI options with large PadN sub-options (see: Section 12) and/or with a trailing large NULL packet in a super-packet (see: Section 6.9) in order to receive a small NA response from the OAL destination. While observing the minimum MPS will always result in robust and secure behavior, the OAL source should optimize path MPS values when more

efficient utilization may result in better performance (e.g. for wireless aviation data links). The OAL source should maintain separate path MPS values for each (source, target) underlay interface pair for the same OAL destination, since different underlay interface pairs may support differing path MPS values.

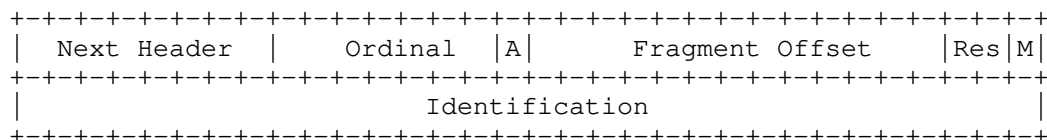
When the OAL source performs fragmentation, it SHOULD produce the minimum number of non-overlapping fragments under current MPS constraints, where each non-final fragment MUST be at least as large as the minimum MPS, while the final fragment MAY be smaller. The OAL source also converts all original IP packets no larger than the current MPS (or larger than 65535 octets) into atomic fragments by including a Fragment Header with Fragment Offset and More Fragments both set to 0. The OAL source then inserts a Routing Header (if necessary) following the IPv6 encapsulation header and before the Fragment Header. If the original IP packet is larger than 65535, the OAL source also inserts a Hop-By-Hop header with Jumbo Payload option immediately following the IPv6 encapsulation header and before the Routing Header (if necessary), then includes an (atomic) Fragment Header. The header extension order for each fragment therefore appears as the OAL IPv6 header followed by Hop-By-Hop header followed by Routing Header followed by Fragment Header.

The OAL source next appends the OAL checksum as the final two octets of the final fragment while increasing its (Jumbo) Payload Length by 2. If appending the checksum would cause the final fragment to exceed the current MPS, the OAL source instead reduces this "former" final fragment's Payload Length (PL) by $(N*8 + (PL \bmod 8))$ octets, where N is an integer that would result in a non-zero reduction but without causing the former final fragment to become smaller than the minimum MPS. The OAL source then creates a "new" final fragment by copying the OAL IPv6 header and extension headers from the former final fragment, then copying the $(N*8 + (PL \bmod 8))$ octets from the end of the former final fragment immediately following the new final fragment extension headers. The OAL source then sets the former final fragment's More Fragments flag to 1, increments the new final fragment's fragment offset by the former final fragment's new $(PL / 8)$ and finally appends the checksum the same as discussed above.

Next, the OAL source replaces the IPv6 Fragment Header 1-octet "Reserved" field (and for first fragments also the 2-bit "Reserved Flags" field) with OMNI-specific encodings as shown in:



a) First fragment

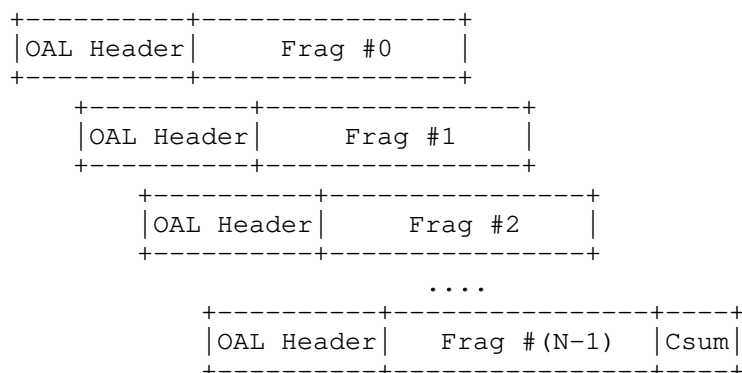


a) Non-first fragment

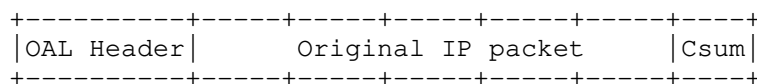
Figure 5: IPv6 Fragment Header Reserved Fields Redefined

For the first fragment, the OAL source sets the "(A)RQ" flag then sets "Parcel ID", "(P)arcel" and "(S)ub-Parcels" as specified in Section 6.14. For each non-first fragment, the OAL source instead sets the "(A)RQ" flag and writes a monotonically-increasing "Ordinal" value between 1 and 127. Specifically, the OAL source writes the ordinal number '1' for the first non-first fragment, '2' for the second, '3' for the third, etc. up to the final fragment or the ordinal value '127', whichever comes first. (For any additional non-first fragments beyond ordinal '127', the OAL source instead writes the value '0' in the Ordinal field and clears the "(A)RQ" flag. The first fragment is implicitly always considered ordinal number '0' even though the header does not include an explicit Ordinal field.)

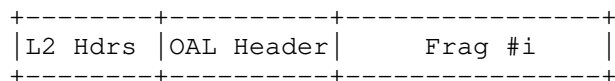
The OAL source finally encapsulates the fragments in L2 headers to form carrier packets and forwards them over an underlay interface, while retaining the fragments and their ordinal numbers (i.e., #0, #1, #2, etc. up to #127) for a brief period to support link-layer retransmissions (see: Section 6.7). OAL fragment and carrier packet formats are shown in Figure 6.



a) OAL fragmentation (Csum in final fragment)



b) An OAL atomic fragment



c) OAL carrier packet after L2 encapsulation

Figure 6: OAL Fragments and Carrier Packets

Note: the minimum MPS assumes that any middleboxes (e.g. IPv4 NATs) that connect private networks with path MTUs smaller than 576 octets must reassemble any fragmented (outbound) IPv4 carrier packets sent by OAL sources before forwarding them to external Internetworks since middleboxes that connect OAL destinations often unconditionally drop (inbound) IPv4 fragments. However, when the path MTU in the destination private network is small, the OAL destination itself will be able to reassemble any IPv4 fragmentation that occurs in the inbound path.

6.2. OAL L2 Encapsulation and Re-Encapsulation

The OAL source or intermediate node next encapsulates each OAL fragment (with either full or compressed headers) in L2 encapsulation headers to create a carrier packet. The OAL source or intermediate node (i.e., the L2 source) includes a UDP header as the innermost sublayer if NAT traversal and/or packet filtering middlebox traversal are required; otherwise, the L2 source includes either a full or compressed IP header and/or an actual link-layer header (e.g., such

as for Ethernet-compatible links). The L2 source then appends any additional encapsulation sublayer headers necessary and presents the resulting carrier packet to an underlay interface, where the underlay network conveys it to a next-hop OAL intermediate node or destination (i.e., the L2 destination).

The L2 source encapsulates the OAL information immediately following the innermost L2 sublayer header. If the first four bits of the encapsulated OAL information following the innermost sublayer header encode the value '6', the information must include an uncompressed IPv6 header (plus extensions) followed by upper layer protocol headers and data. If the first four bits encode the value '4', an uncompressed IPv4 header (plus extensions) followed by upper layer protocol headers and data follows. Otherwise, the first four bits include a "Type" value, and the OAL information appears in an alternate format as specified in Section 6.4 (Types '0' and '1' are currently specified while all other values are reserved for future use). Carrier packets that contain an unrecognized Type value are unconditionally dropped.

The OAL node prepares the innermost L2 encapsulation header for OAL packets as follows:

- * For UDP encapsulation, the L2 source sets the UDP source port to 8060 (i.e., the port number reserved for AERO/OMNI). When the L2 destination is a Proxy/Server or Gateway, the L2 source sets the UDP destination port to 8060; otherwise, the L2 source sets the UDP destination port to its cached port number value for the peer. The L2 source finally sets the UDP Length the same as specified in [RFC0768]. (If the OAL packet includes an IP Jumbogram, the L2 source instead sets the UDP length to 0 and includes a Jumbo Payload option in the L2 IP header.)
- * For IP encapsulation, the L2 source sets the IP {Protocol, Next-Header} to TBD1 (see: IANA Considerations) and sets the {Total, Payload} Length the same as specified in [RFC0791] or [RFC8200]. (If the OAL packet includes a true Jumbogram, the L2 source includes a Jumbo Payload option and sets {Total, Payload} Length plus the Jumbo Payload length according to the OAL length information.)

- * For direct encapsulations over Ethernet-compatible links, the EtherType is set to TBD2 (see: IANA Considerations). Since the Ethernet header does not include a length field, for the OMNI EtherType the Ethernet header is followed by a four-octet Payload Length field followed immediately by the encapsulated OAL information. The Payload Length field encodes the length in octets (in network byte order) of the OAL information exclusive of the lengths of the Ethernet header and trailer.

When an L2 source includes a UDP header, it SHOULD calculate and include a UDP checksum in carrier packets with full OAL headers to prevent mis-delivery, and MAY disable UDP checksums in carrier packets with compressed OAL headers (see: Section 6.4). If the L2 source discovers that a path is dropping carrier packets with UDP checksums disabled, it should enable UDP checksums in future carrier packets sent to the same L2 destination. If the L2 source discovers that a path is dropping carrier packets that do not include a UDP header, it should include a UDP header in future carrier packets.

When an L2 source sends carrier packets with compressed OAL headers and with UDP checksums disabled, mis-delivery due to corruption of the 4-octet Multilink Forwarding Vector Index (MFVI) is possible but unlikely since the corrupted index would somehow have to match valid state in the (sparsely-populated) Multilink Forwarding Information Based (MFIB). In the unlikely event that a match occurs, an OAL destination may receive a mis-delivered carrier packet but can immediately reject packets with an incorrect Identification. If the Identification value is somehow accepted, the OAL destination may submit the mis-delivered carrier packet to the reassembly cache where it will most likely be rejected due to incorrect reassembly parameters. If a reassembly that includes the mis-delivered carrier packets somehow succeeds (or, for atomic fragments) the OAL destination will verify the OAL checksum to detect corruption. Finally, any spurious data that somehow eludes all prior checks will be detected and rejected by end-to-end upper layer integrity checks. See: [RFC6935][RFC6936] for further discussion.

For L2 encapsulations over IP, when the L2 source is also the OAL source it next copies the "Type of Service/Traffic Class" [RFC2983] and "Explicit Congestion Notification (ECN)" [RFC3168] values in the OAL header into the corresponding fields in the L2 IP header, then (for IPv6) set the L2 IPv6 header "Flow Label" as specified in [RFC6438]. The L2 source then sets the L2 IP TTL/Hop Limit the same as for any host (i.e., it does not copy the Hop Limit value from the OAL header) and finally sets the source and destination IP addresses to direct the carrier packet to the next hop. For carrier packets undergoing re-encapsulation, the OAL intermediate node L2 source decrements the OAL header Hop Limit and discards the carrier packet

if the value reaches 0. The L2 source then copies the "Type of Service/Traffic Class" and "Explicit Congestion Notification (ECN)" values from the previous hop L2 encapsulation header into the OAL header (if present), then finally sets the source and destination IP addresses the same as above.

Following L2 encapsulation/re-encapsulation, the L2 source forwards the resulting carrier packets over one or more underlay interfaces. The underlay interfaces often connect directly to physical media on the local platform (e.g., a laptop computer with WiFi, etc.), but in some configurations the physical media may be hosted on a separate Local Area Network (LAN) node. In that case, the OMNI interface can establish a Layer-2 VLAN or a point-to-point tunnel (at a layer below the underlay interface) to the node hosting the physical media. The OMNI interface may also apply encapsulation at the underlay interface layer (e.g., as for a tunnel virtual interface) such that carrier packets would appear "double-encapsulated" on the LAN; the node hosting the physical media in turn removes the LAN encapsulation prior to transmission or inserts it following reception. Finally, the underlay interface must monitor the node hosting the physical media (e.g., through periodic keepalives) so that it can convey up/down/status information to the OMNI interface.

6.3. OAL L2 Decapsulation and Reassembly

When an OMNI interface receives a carrier packet from an underlay interface, it copies the ECN value from the L2 encapsulation headers into the OAL header if the carrier packet contains a first-fragment. The OMNI interface next discards the L2 encapsulation headers and examines the OAL header of the enclosed OAL fragment. If the OAL fragment is addressed to a different node, the OMNI interface (acting as an OAL intermediate node) re-encapsulates and forwards while decrementing the OAL Hop Limit as discussed in Section 6.2. If the OAL fragment is addressed to itself, the OMNI interface (acting as an OAL destination) accepts or drops the fragment based on the (Source, Destination, Identification)-tuple and/or integrity checks.

The OAL destination next drops all non-final OAL fragments smaller than the minimum MPS and all fragments that would overlap or leave "holes" smaller than the minimum MPS with respect to other fragments already received. The OAL destination updates a checklist of accepted fragments of the same OAL packet that include an Ordinal number (i.e., Ordinals 0 through 127), but admits all accepted fragments into the reassembly cache after first removing any extension headers except for the fragment header itself. When the OAL destination receives the final fragment (i.e., the one with More Fragments set to 0), it caches the trailing checksum and reduces the Payload Length by 2. When reassembly is complete, the OAL

destination verifies the OAL packet checksum and discards the packet if the checksum is incorrect. If the OAL packet was accepted, the OAL destination finally removes the OAL headers and delivers the original IP packet to the network layer.

Carrier packets often travel over paths where all links in the path include CRC-32 integrity checks for effective hop-by-hop error detection for payload sizes up to 9180 octets [CRC], but other paths may traverse links (such as tunnels over IPv4) that do not include adequate integrity protection. The OAL checksum therefore allows OAL destinations to detect reassembly misassociation splicing errors and/or carrier packet corruption caused by unprotected links [CKSUM].

The OAL checksum also provides algorithmic diversity with respect to both lower layer CRCs and upper layer Internet checksums as part of a complimentary multi-layer integrity assurance architecture. Any corruption not detected by lower layer integrity checks is therefore very likely to be detected by upper layer integrity checks that use diverse algorithms.

6.4. OAL Header Compression

OAL sources that send carrier packets with full OAL headers include a CRH-32 extension for segment-by-segment forwarding based on a Multilink Forwarding Information Base (MFIB) in each OAL intermediate node. OAL source, intermediate and destination nodes can instead establish header compression state through IPv6 ND NS/NA message exchanges. After an initial NS/NA exchange, OAL nodes can apply OAL Header Compression to significantly reduce encapsulation overhead.

Each OAL node establishes MFIB soft state entries known as Multilink Forwarding Vectors (MVF's) which support both carrier packet forwarding and OAL header compression/decompression. For OAL sources, each MFV is referenced by a single Multilink Forwarding Vector Index (MFVI) that provides compression/decompression and forwarding context for the next hop. For OAL destinations, the MFV is referenced by a single MFVI that provides context for the previous hop. For OAL intermediate nodes, the MFV is referenced by two MFVIs - one for the previous hop and one for the next hop.

When an OAL node forwards carrier packets to a next hop, it can include a full OAL header with a CRH-32 extension containing one or more MFVIs. Whenever possible, however, the OAL node should instead omit significant portions of the OAL header (including the CRH-32) while applying OAL header compression. The full or compressed OAL header follows immediately after the innermost L2 encapsulation (i.e., UDP, IP or L2) as discussed in Section 6.2. Two OAL compressed header types (Types '0' and '1') are currently specified below (note that the (A)RQ flag is always considered set and therefore omitted from the compressed headers themselves).

For OAL first-fragments (including atomic fragments), the OAL node uses OMNI Compressed Header - Type 0 (OCH-0) format as shown in Figure 7:

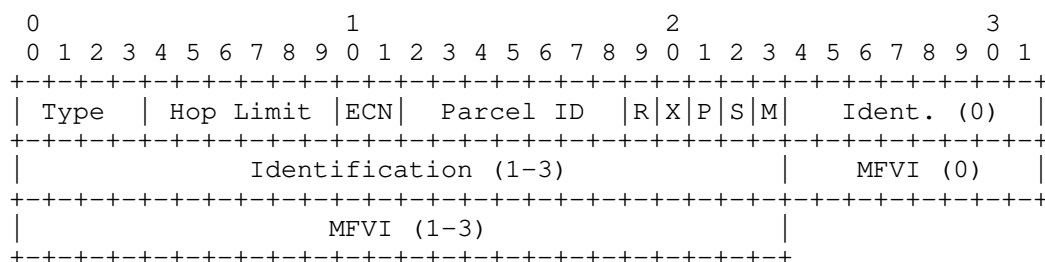


Figure 7: OMNI Compressed Header - Type 0 (OCH-0)

The format begins with a 4-bit Type, a 6-bit Hop Limit, a 2-bit Explicit Congestion Notification (ECN) field, a 7-bit Parcel ID and 5 flag bits. The format concludes with a 4-octet Identification field followed (optionally) by a 4-octet MFVI field. The OAL node sets Type to the value 0, sets Hop Limit to the minimum of the uncompressed OAL header Hop Limit and 63, sets ECN the same as for an uncompressed OAL header, and sets (P)arcel, (S)ub-parcels, (M)ore Fragments, Identification) the same as for an uncompressed fragment header. The OAL node finally sets Inde(X) and includes an MFVI if necessary; otherwise, it clears Inde(X) and omits the MFVI. (The (R)eserved flag is set to 0 on transmission and ignored on reception.)

The OAL first fragment (beginning with the original IP header) is then included immediately following the OCH-0 header, and the L2 header length field is reduced by the difference in length between the compressed headers and full-length OAL IPv6 and Fragment headers. The OAL destination can therefore determine the Payload Length by examining the L2 header length field and/or the length field(s) in the original IP header. The OCH-0 format applies for first fragments only, which are always regarded as ordinal fragment 0 even though no explicit Ordinal field is included. The (A)RQ flag is always implicitly set, and therefore omitted from the OCH-0 header.

For OAL non-first fragments (i.e., those with non-zero Fragment Offsets), the OAL uses OMNI Compressed Header - Type 1 (OCH-1) format as shown in Figure 8:

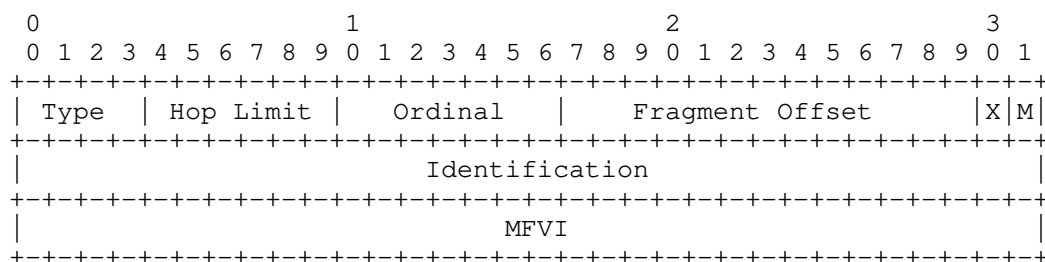


Figure 8: OMNI Compressed Header - Type 1 (OCH-1)

The format begins with a 4-bit Type, a 6-bit Hop Limit, a 7-bit Ordinal, a 13-bit Fragment Offset and 2 flag bits. The format concludes with a 4-octet Identification field followed (optionally) by a 4-octet MFVI field. The OAL node sets Type to the value 1, sets Hop Limit to the minimum of the uncompressed OAL header Hop Limit and 63, and sets (Ordinal, Fragment Offset, (M)ore Fragments, Identification) the same as for an uncompressed fragment header. If an MFVI is needed, the OAL node finally sets Inde(X) and includes an MFVI; otherwise, the node clears Inde(X) and omits the MFVI.

The OAL non-first fragment body is then included immediately following the OCH-1 header, and the L2 header length field is reduced by the difference in length between the compressed headers and full-length OAL IPv6 and Fragment headers. The OAL destination will then be able to determine the Payload Length by examining the L2 header length field. The OCH-1 format applies for non-first fragments only; therefore, the OAL source sets Ordinal to a monotonically increasing value beginning with 1 for the first non-first fragment, 2 for the second non-first fragment, etc., up to and including the final fragment. If more than 127 non-first fragments are included, these additional fragments instead set Ordinal to 0. The (A)RQ flag is always implicitly set, and therefore omitted from the OCH-1 header.

When an OAL destination or intermediate node receives a carrier packet, it determines the length of the encapsulated OAL information by examining the length field of the innermost L2 header, verifies that the innermost next header field indicates OMNI (see: Section 6.2), then examines the first four bits immediately following the innermost header. If the bits contain the value 4 or 6, the OAL node processes the remainder as an uncompressed OAL/IP header. If the bits contain a value 0 or 1, the OAL node instead processes the remainder of the header as an OCH-0/1 as specified above.

For carrier packets with OCH or full OAL headers addressed to itself and with CRH-32 extensions, the OAL node then uses the MFVI to locate the cached MFV which determines the next hop. During forwarding, the OAL node changes the MFVI to the cached value for the MVE next hop. If the OAL node is the destination, it instead reconstructs the full OAL headers then adds the resulting OAL fragment to the reassembly cache if the Identification is acceptable. (Note that for carrier packets that include an OCH-0 with both the X and M flags set to 0, the OAL node can instead locate forwarding state by examining the original IP packet header information that appears immediately after the OCH-0 header.)

Note: OAL header compression does not interfere with checksum calculation and verification, which must be applied according to the full OAL pseudo-header per Section 6.1 even when compression is used.

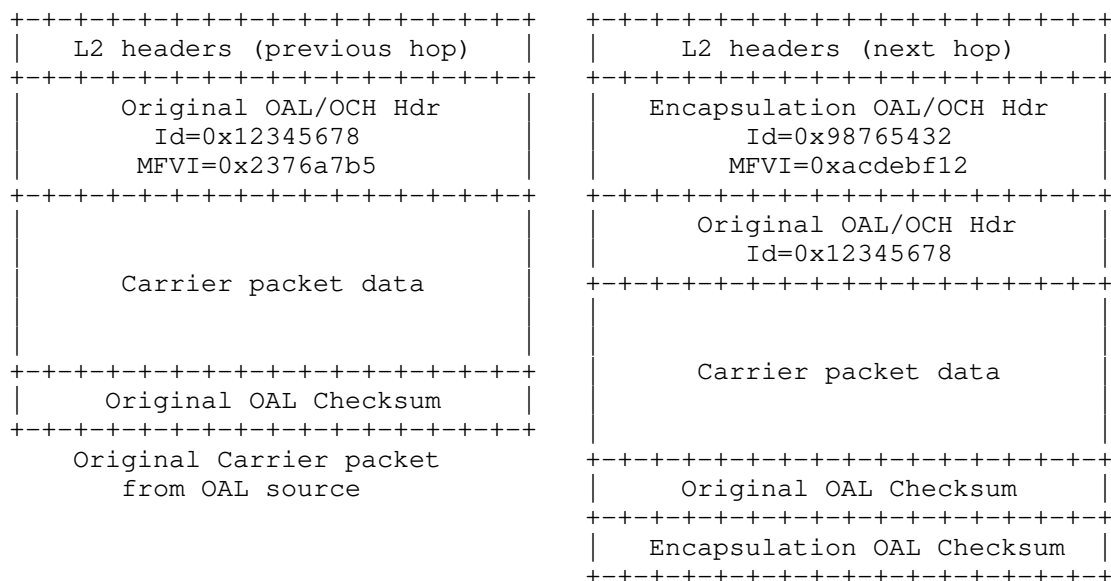
Note: The OCH-0/1 formats do not include the Traffic Class and Flow Label information that appears in uncompressed OAL IPv6 headers. Therefore, when OAL header compression state is initialized the Traffic Class and Flow Label are considered fixed for as long as the flow uses OCH-0/1 headers. If the flow requires frequent changes to Traffic Class and/or Flow Label information, it can include uncompressed OAL headers either continuously or periodically to update header compression state.

6.5. OAL-in-OAL Encapsulation

When an OAL source is unable to forward carrier packets directly to an OAL destination without "tunneling" through a pair of OAL intermediate nodes, the OAL source must regard the intermediate nodes as ingress and egress tunnel endpoints. This will result in nested OAL-in-OAL encapsulation in which the OAL source performs fragmentation on the inner OAL packet then forwards the fragments to the ingress tunnel endpoint which encapsulates each resulting OAL fragment in an additional OAL header before performing fragmentation following encapsulation.

For example, if the OAL source has an NCE for the OAL destination with MFVI 0x2376a7b5 and Identification 0x12345678 and the OAL ingress tunnel endpoint has an NCE for the OAL egress tunnel endpoint with MFVI 0xacdeb12 and Identification 0x98765432, the OAL source prepares the carrier packets using compressed/uncompressed OAL headers that include the MFVI and Identification corresponding to the OAL destination and with L2 header information addressed to the next hop toward the ingress tunnel endpoint. When the ingress tunnel endpoint receives the carrier packet, it recognizes the current MFVI included by the OAL source and determines the correct next hop MFVI.

The ingress tunnel endpoint then discards the L2 headers from the previous hop and encapsulates the original compressed/uncompressed OAL header within a second compressed/uncompressed OAL header while including the next-hop MFVI in the outer OAL encapsulation header and omitting the MFVI in the inner header. The ingress tunnel endpoint then includes L2 encapsulation headers with destinations appropriate for the next hop on the path to the egress tunnel endpoint. The encapsulation appears as shown in Figure 9:



Carrier packet following OAL ingress
(re)encapsulation before fragmentation

Figure 9: Carrier Packet in Carrier Packet Encapsulation

Note that only a single OAL-in-OAL encapsulation layer is supported, and that MFVIs appear only in the outer OAL header (i.e., either within a CRH-32 routing header when a full OAL header is used or within an OCH header with X set to 0). The inner OAL header should omit the CRH-32 header or use an OCH header with X set to 1, respectively.

Note that OAL/OCH encapsulation may cause the payloads of OAL packets produced by the ingress tunnel endpoint to exceed the minimum MPS by a small amount. If the ingress has assurance that the path to the egress will include only links capable of transiting the resulting (slightly larger) carrier packets it should forward without further fragmentation. Otherwise, the ingress must perform fragmentation following encapsulation to produce two fragments such that the size of the first fragment matches the size of the original OAL packet, and with the remainder in a second fragment. The egress tunnel endpoint must then reassemble then decapsulate to arrive at the original OAL packet which is then subject to further forwarding.

6.6. OAL Identification Window Maintenance

The OAL encapsulates each original IP packet as an OAL packet then performs fragmentation to produce one or more carrier packets with the same 32-bit Identification value. In environments where spoofing is not considered a threat, OMNI interfaces send OAL packets with Identifications beginning with an unpredictable Initial Send Sequence (ISS) value [RFC7739] monotonically incremented (modulo 2^{32}) for each successive OAL packet sent to either a specific neighbor or to any neighbor. (The OMNI interface may later change to a new unpredictable ISS value as long as the Identifications are assured unique within a timeframe that would prevent the fragments of a first OAL packet from becoming associated with the reassembly of a second OAL packet.) In other environments, OMNI interfaces should maintain explicit per-neighbor send and receive windows to detect and exclude spurious carrier packets that might clutter the reassembly cache as discussed below.

OMNI interface neighbors use TCP-like synchronization to maintain windows with unpredictable ISS values incremented (modulo 2^{32}) for each successive OAL packet and re-negotiate windows often enough to maintain an unpredictable profile. OMNI interface neighbors exchange IPv6 ND messages with OMNI options that include TCP-like information fields to manage streams of OAL packets instead of streams of octets. As a link-layer service, the OAL provides low-persistence best-effort retransmission with no mitigations for duplication, reordering or deterministic delivery. Since the service model is best-effort and only control message sequence numbers are acknowledged, OAL nodes can select unpredictable new initial sequence numbers outside of the current window without delaying for the Maximum Segment Lifetime (MSL).

OMNI interface neighbors maintain current and previous window state in IPv6 ND neighbor cache entries (NCEs) to support dynamic rollover to a new window while still sending OAL packets and accepting carrier packets from the previous windows. Each NCE is indexed by the neighbor's ULA, while the OAL encapsulation ULA (which may be different) provides context for Identification verification. OMNI interface neighbors synchronize windows through asymmetric and/or symmetric IPv6 ND message exchanges. When a node receives an IPv6 ND message with new window information, it resets the previous window state based on the current window then resets the current window based on new and/or pending information.

The IPv6 ND message OMNI option header extension sub-option includes TCP-like information fields including Sequence Number, Acknowledgement Number, Window and flags (see: Section 12). OMNI interface neighbors maintain the following TCP-like state variables in the NCE:

Send Sequence Variables (current, previous and pending)

- SND.NXT - send next
- SND.WND - send window
- ISS - initial send sequence number

Receive Sequence Variables (current and previous)

- RCV.NXT - receive next
- RCV.WND - receive window
- IRS - initial receive sequence number

OMNI interface neighbors "OAL A" and "OAL B" exchange IPv6 ND messages per [RFC4861] with OMNI options that include TCP-like information fields. When OAL A synchronizes with OAL B, it maintains both a current and previous SND.WND beginning with a new unpredictable ISS and monotonically increments SND.NXT for each successive OAL packet transmission. OAL A initiates synchronization by including the new ISS in the Sequence Number of an authentic IPv6 ND message with the SYN flag set and with Window set to M (up to 2^{24}) as a tentative receive window size while creating a NCE in the INCOMPLETE state if necessary. OAL A caches the new ISS as pending, uses the new ISS as the Identification for OAL encapsulation, then sends the resulting OAL packet to OAL B and waits up to RetransTimer milliseconds to receive an IPv6 ND message response with the ACK flag set (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).

When OAL B receives the SYN, it creates a NCE in the STALE state if necessary, resets its RCV variables, caches the tentative (send) window size M, and selects a (receive) window size N (up to 2^{24}) to indicate the number of OAL packets it is willing to accept under the current RCV.WND. (The RCV.WND should be large enough to minimize control message overhead yet small enough to provide an effective filter for spurious carrier packets.) OAL B then prepares an IPv6 ND message with the ACK flag set, with the Acknowledgement Number set to OAL A's next sequence number, and with Window set to N. Since OAL B does not assert an ISS of its own, it uses the IRS it has cached for OAL A as the Identification for OAL encapsulation then sends the ACK to OAL A.

When OAL A receives the ACK, it notes that the Identification in the OAL header matches its pending ISS. OAL A then sets the NCE state to REACHABLE and resets its SND variables based on the Window size and Acknowledgement Number (which must include the sequence number following the pending ISS). OAL A can then begin sending OAL packets to OAL B with Identification values within the (new) current SND.WND for up to ReachableTime milliseconds or until the NCE is updated by a new IPv6 ND message exchange. This implies that OAL A must send a new SYN before sending more than N OAL packets within the current SND.WND, i.e., even if ReachableTime is not nearing expiration. After OAL B returns the ACK, it accepts carrier packets received from OAL A within either the current or previous RCV.WND as well as any new authentic NS/RS SYN messages received from OAL A even if outside the windows.

OMNI interface neighbors can employ asymmetric window synchronization as described above using two independent (SYN -> ACK) exchanges (i.e., a four-message exchange), or they can employ symmetric window synchronization using a modified version of the TCP three-way handshake as follows:

- * OAL A prepares a SYN with an unpredictable ISS not within the current SND.WND and with Window set to M as a tentative receive window size. OAL A caches the new ISS and Window size as pending information, uses the pending ISS as the Identification for OAL encapsulation, then sends the resulting OAL packet to OAL B and waits up to RetransTimer milliseconds to receive an ACK response (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).
- * OAL B receives the SYN, then resets its RCV variables based on the Sequence Number while caching OAL A's tentative receive Window size M and a new unpredictable ISS outside of its current window as pending information. OAL B then prepares a response with Sequence Number set to the pending ISS and Acknowledgement Number set to OAL A's next sequence number. OAL B then sets both the SYN and ACK flags, sets Window to N and sets the OPT flag according to whether an explicit concluding ACK is optional or mandatory. OAL B then uses the pending ISS as the Identification for OAL encapsulation, sends the resulting OAL packet to OAL A and waits up to RetransTimer milliseconds to receive an acknowledgement (retransmitting up to MAX_UNICAST_SOLICIT times if necessary).
- * OAL A receives the SYN/ACK, then resets its SND variables based on the Acknowledgement Number (which must include the sequence number following the pending ISS) and OAL B's advertised Window N. OAL A then resets its RCV variables based on the Sequence Number and marks the NCE as REACHABLE. If the OPT flag is clear, OAL A next prepares an immediate solicited NA message with the ACK flag set,

the Acknowledgement Number set to OAL B's next sequence number, with Window set a value that may be the same as or different than M, and with the OAL encapsulation Identification to SND.NXT, then sends the resulting OAL packet to OAL B. If the OPT flag is set and OAL A has OAL packets queued to send to OAL B, it can optionally begin sending their carrier packets under the (new) current SND.WND as implicit acknowledgements instead of returning an explicit ACK. In that case, the tentative Window size M becomes the current receive window size.

- * OAL B receives the implicit/explicit acknowledgement(s) then resets its SND state based on the pending/advertised values and marks the NCE as REACHABLE. If OAL B receives an explicit acknowledgement, it uses the advertised Window size and abandons the tentative size. (Note that OAL B sets the OPT flag in the SYN/ACK to assert that it will interpret timely receipt of carrier packets within the (new) current window as an implicit acknowledgement. Potential benefits include reduced delays and control message overhead, but use case analysis is outside the scope of this specification.)

Following synchronization, OAL A and OAL B hold updated NCEs and can exchange OAL packets with Identifications set to SND.NXT while the state remains REACHABLE and there is available window capacity. Either neighbor may at any time send a new SYN to assert a new ISS. For example, if OAL A's current SND.WND for OAL B is nearing exhaustion and/or ReachableTime is nearing expiration, OAL A continues to send OAL packets under the current SND.WND while also sending a SYN with a new unpredictable ISS. When OAL B receives the SYN, it resets its RCV variables and may optionally return either an asymmetric ACK or a symmetric SYN/ACK to also assert a new ISS. While sending SYNs, both neighbors continue to send OAL packets with Identifications set to the current SND.NXT then reset the SND variables after an acknowledgement is received.

While the optimal symmetric exchange is efficient, anomalous conditions such as receipt of old duplicate SYNs can cause confusion for the algorithm as discussed in Section 3.4 of [RFC0793]. For this reason, the OMNI option header includes an RST flag which OAL nodes set in solicited NA responses to ACKs received with incorrect acknowledgement numbers. The RST procedures (and subsequent synchronization recovery) are conducted exactly as specified in [RFC0793].

OMNI interfaces may set the PNG ("ping") flag when a reachability confirmation outside the context of the IPv6 ND protocol is needed (OMNI interfaces therefore most often set the PNG flag in advertisement messages and ignore it in solicitation messages). When

an OMNI interface receives a PNG, it returns an unsolicited NA (uNA) ACK with the PNG message Identification in the Acknowledgment, but without updating RCV state variables. OMNI interfaces return unicast uNA ACKs even for multicast PNG destination addresses, since OMNI link multicast is based on unicast emulation.

OMNI interfaces that employ the window synchronization procedures described above observe the following requirements:

- * OMNI interfaces MUST select new unpredictable ISS values that are at least a full window outside of the current SND.WND.
- * OMNI interfaces MUST set the initial SYN message Window field to a tentative value to be used only if no concluding NA ACK is sent.
- * OMNI interfaces that receive advertisements with the PNG and/or SYN flag set MUST NOT set the PNG and/or SYN flag in uNA responses.
- * OMNI interfaces that send advertisements with the PNG and/or SYN flag set MUST ignore uNA responses with the PNG and/or SYN flag set.
- * OMNI interfaces MUST send IPv6 ND messages used for window synchronization securely while using unpredictable initial Identification values until synchronization is complete.

Note: Although OMNI interfaces employ TCP-like window synchronization and support uNA ACK responses to SYNs and PNGs, all other aspects of the IPv6 ND protocol (e.g., control message exchanges, NCE state management, timers, retransmission limits, etc.) are honored exactly per [RFC4861].

Note: Recipients of OAL-encapsulated IPv6 ND messages index the NCE based on the message source address, which also determines the carrier packet Identification window. However, IPv6 ND messages may contain a message source address that does not match the OMNI encapsulation source address when the recipient acts as a proxy.

Note: OMNI interface neighbors apply the same send and receive windows for all of their (multilink) underlay interface pairs that exchange carrier packets. Each interface pair represents a distinct underlay network path, and the set of paths traversed may be highly diverse when multiple interface pairs are used. OMNI intermediate nodes therefore SHOULD NOT cache window synchronization parameters in IPv6 ND messages they forward since there is no way to ensure network-wide middlebox state consistency.

6.7. OAL Fragment Retransmission

When the OAL source sends carrier packets to an OAL destination, it should cache recently sent packets in case timely best-effort selective retransmission is requested. The OAL destination in turn maintains a checklist for the (Source, Destination, Identification)-tuple of recently received carrier packets and notes the ordinal numbers of OAL packet fragments already received (i.e., as Frag #0, Frag #1, Frag #2, etc.). The timeframe for maintaining the OAL source and destination caches determines the link persistence (see: [RFC3366]).

If the OAL destination notices some fragments missing after most other fragments within the same link persistence timeframe have already arrived, it may issue an Automatic Repeat Request (ARQ) with Selective Repeat (SR) by sending a uNA message to the OAL source. The OAL destination creates a uNA message with an OMNI option with one or more Fragmentation Report (FRAGREP) sub-options that include a list of (Identification, Bitmap)-tuples for fragments received and missing from this OAL source (see: Section 12 and [I-D.templin-6man-fragrep]). The OAL destination includes an authentication signature if necessary, performs OAL encapsulation (with the its own address as the OAL source and the source address of the message that prompted the uNA as the OAL destination) and sends the message to the OAL source.

When the OAL source receives the uNA message, it authenticates the message then examines the FRAGREP. For each (Source, Destination, Identification)-tuple, the OAL source determines whether it still holds the corresponding carrier packets in its cache and retransmits any for which the Bitmap indicates a loss event. For example, if the Bitmap indicates that ordinal fragments #3, #7, #10 and #13 from the OAL packet with Identification 0x12345678 are missing the OAL source only retransmits carrier packets containing those fragments. When the OAL destination receives the retransmitted carrier packets, it admits the enclosed fragments into the reassembly cache and updates its checklist. If some fragments are still missing, the OAL destination may send a small number of additional uNA ARQ/SRs within the link persistence timeframe.

The OAL therefore provides a link-layer low-to-medium persistence ARQ/SR service consistent with [RFC3366] and Section 8.1 of [RFC3819]. The service provides the benefit of timely best-effort link-layer retransmissions which may reduce packet loss and avoid some unnecessary end-to-end delays. This best-effort network-based service therefore compliments higher layer end-to-end protocols responsible for true reliability.

6.8. OAL MTU Feedback Messaging

When the OMNI interface forwards original IP packets from the network layer, it invokes the OAL and returns internally-generated ICMPv4 Fragmentation Needed [RFC1191] or ICMPv6 Path MTU Discovery (PMTUD) Packet Too Big (PTB) [RFC8201] messages as necessary. This document refers to both of these ICMPv4/ICMPv6 message types simply as "PTBs", and introduces a distinction between PTB "hard" and "soft" errors as discussed below and also in [I-D.templin-6man-fragrep].

Ordinary PTB messages with ICMPv4 header "unused" field or ICMPv6 header Code field value 0 are hard errors that always indicate that a packet has been dropped due to a real MTU restriction. However, the OMNI interface can also forward large original IP packets via OAL encapsulation and fragmentation while at the same time returning PTB soft error messages (subject to rate limiting) if it deems the original IP packet too large according to factors such as link performance characteristics, number of fragments needed, reassembly congestion, etc. This ensures that the path MTU is adaptive and reflects the current path used for a given data flow. The OMNI interface can therefore continuously forward packets without loss while returning PTB soft error messages recommending a smaller size if necessary. Original sources that receive the soft errors in turn reduce the size of the packets they send (i.e., the same as for hard errors), but can soon resume sending larger packets if the soft errors subside.

An OAL source sends PTB soft error messages by setting the ICMPv4 header "unused" field or ICMPv6 header Code field to the value 1 if the packet was dropped or 2 if the packet was forwarded successfully. The OAL source sets the PTB destination address to the original IP packet source, and sets the source address to one of its OMNI interface addresses that is routable from the perspective of the original source. The OAL source then sets the MTU field to a value smaller than the original packet size but no smaller than 576 for ICMPv4 or 1280 for ICMPv6, writes the leading portion of the original IP packet first fragment into the "packet in error" field, and returns the PTB soft error to the original source. When the original source receives the PTB soft error, it temporarily reduces the size of the packets it sends the same as for hard errors but may seek to increase future packet sizes dynamically while no further soft errors are arriving. (If the original source does not recognize the soft error code, it regards the PTB the same as a hard error but should heed the retransmission advice given in [RFC8201] suggesting retransmission based on normal packetization layer retransmission timers.)

An OAL destination may experience reassembly cache congestion, and can return uNA messages to the OAL source that originated the fragments (subject to rate limiting) that include OMNI encapsulated PTB messages with code 1 or 2. The OAL destination creates a uNA message with an OMNI option containing an authentication message sub-option if necessary followed optionally by a ICMPv6 Error sub-option that encodes a PTB message with a reduced value and with the leading portion an OAL first fragment containing the header of an original IP packet whose source must be notified (see: Section 12). The OAL destination encapsulates the leading portion of the OAL first fragment (beginning with the OAL header) in the PTB "packet in error" field, signs the message if an authentication sub-option is included, performs OAL encapsulation (with the its own address as the OAL source and the source address of the message that prompted the uNA as the OAL destination) and sends the message to the OAL source.

When the OAL source receives the uNA message, it sends a corresponding network layer PTB soft error to the original source to recommend a smaller size. The OAL source crafts the PTB by extracting the leading portion of the original IP packet from the OMNI encapsulated PTB message (i.e., not including the OAL header) and writes it in the "packet in error" field of a network layer PTB with destination set to the original IP packet source and source set to one of its OMNI interface addresses that is routable from the perspective of the original source.

Original sources that receive PTB soft errors can dynamically tune the size of the original IP packets they to send to produce the best possible throughput and latency, with the understanding that these parameters may change over time due to factors such as congestion, mobility, network path changes, etc. The receipt or absence of soft errors should be seen as hints of when increasing or decreasing packet sizes may be beneficial. The OMNI interface supports continuous transmission and reception of packets of various sizes in the face of dynamically changing network conditions. Moreover, since PTB soft errors do not indicate a hard limit, original sources that receive soft errors can resume sending larger packets without waiting for the recommended 10 minutes specified for PTB hard errors [RFC1191][RFC8201]. The OMNI interface therefore provides an adaptive service that accommodates MTU diversity especially well-suited for dynamic multilink environments.

6.9. OAL Super-Packets

By default, the OAL source includes a 40-octet IPv6 encapsulation header for each original IP packet during OAL encapsulation. The OAL source also calculates then performs fragmentation such that a copy of the 40-octet IPv6 header plus an 8-octet IPv6 Fragment Header is included in each OAL fragment (when a Routing Header is added, the OAL encapsulation headers become larger still). However, these encapsulations may represent excessive overhead in some environments. OAL header compression can dramatically reduce the amount of encapsulation overhead, however a complimentary technique known as "packing" (see: [I-D.ietf-intarea-tunnels]) supports encapsulation of multiple original IP packets and/or control messages within a single OAL "super-packet".

When the OAL source has multiple original IP packets to send to the same OAL destination with total length no larger than the OAL destination MRU, it can concatenate them into a super-packet encapsulated in a single OAL header. Within the OAL super-packet, the IP header of the first original IP packet (iHa) followed by its data (iDa) is concatenated immediately following the OAL header, then the IP header of the next original packet (iHb) followed by its data (iDb) is concatenated immediately following the first original packet, etc. with a trailing checksum field included in the final fragment. The OAL super-packet format is transposed from [I-D.ietf-intarea-tunnels] and shown in Figure 10:

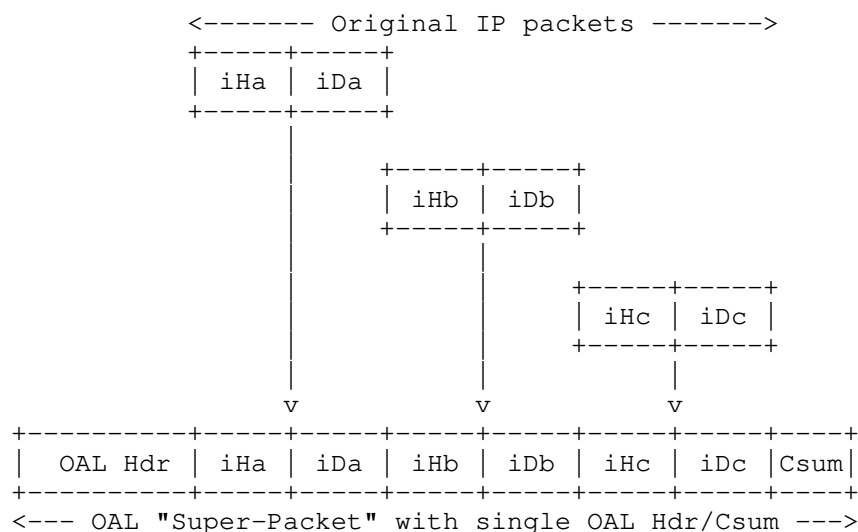


Figure 10: OAL Super-Packet Format

When the OAL source prepares a super-packet, it applies OAL fragmentation, includes a trailing checksum in the final fragment, applies L2 encapsulation to each fragment then sends the resulting carrier packets to the OAL destination. When the OAL destination receives the super-packet it sets aside the trailing checksum, reassembles if necessary, then verifies the checksum while regarding the remaining OAL header Payload Length as the sum of the lengths of all payload packets. The OAL destination then selectively extracts each original IP packet (e.g., by setting pointers into the super-packet buffer and maintaining a reference count, by copying each packet into a separate buffer, etc.) and forwards each packet to the network layer. During extraction, the OAL determines the IP protocol version of each successive original IP packet 'j' by examining the four most-significant bits of iH(j), and determines the length of the packet by examining the rest of iH(j) according to the IP protocol version.

When an OAL source prepares a super-packet that includes an IPv6 ND message with an authentication signature or ICMPv6 message checksum as the first original IP packet (i.e., iHa/iDa), it calculates the authentication signature or checksum over the remainder of super-packet. Security and integrity for forwarding initial protocol data packets in conjunction with IPv6 ND messages used to establish NCE state are therefore supported. (A common use case entails a path MPS probe beginning with a signed IPv6 ND message followed by a NULL IPv6 packet with a suitably large (Jumbo) Payload Length but with Next Header set to 59 for "No Next Header".)

6.10. OAL Bubbles

OAL sources may send NULL OAL packets known as "bubbles" for the purpose of establishing Network Address Translator (NAT) state on the path to the OAL destination. The OAL source prepares a bubble by crafting an OAL header with appropriate IPv6 source and destination ULAs, with the IPv6 Next Header field set to the value 59 ("No Next Header" - see [RFC8200]) and with only the trailing OAL Checksum field (i.e., and no protocol data) immediately following the IPv6 header.

The OAL source includes a random Identification value then encapsulates the OAL packet in L2 headers destined to either the mapped address of the OAL destination's first-hop ingress NAT or the L2 address of the OAL destination itself. When the OAL source sends the resulting carrier packet, any egress NATs in the path toward the L2 destination will establish state based on the activity but the bubble will be harmlessly discarded by either an ingress NAT on the path to the OAL destination or by the OAL destination itself.

The bubble concept for establishing NAT state originated in [RFC4380] and was later updated by [RFC6081]. OAL bubbles may be employed by mobility services such as [I-D.templin-6man-aero].

6.11. OAL Requirements

In light of the above, OAL sources, destinations and intermediate nodes observe the following normative requirements:

- * OAL sources MUST forward original IP packets either larger than the OMNI interface MRU or smaller than the minimum MPS minus the trailing checksum size as atomic fragments (i.e., and not as multiple fragments).
- * OAL sources MUST produce non-final fragments with payloads no smaller than the minimum MPS during fragmentation.
- * OAL intermediate nodes SHOULD and OAL destinations MUST unconditionally drop any non-final OAL fragments with payloads smaller than the minimum MPS.
- * OAL destinations MUST drop any new OAL fragments with offset and length that would overlap with other fragments and/or leave holes smaller than the minimum MPS between fragments that have already been received.

Note: Under the minimum MPS, ordinary 1500 octet original IP packets would require at most 4 OAL fragments, with each non-final fragment containing 400 payload octets and the final fragment containing 302 payload octets (i.e., the final 300 octets of the original IP packet plus the 2 octet trailing checksum). For all packet sizes, the likelihood of successful reassembly may improve when the OMNI interface sends all fragments of the same fragmented OAL packet consecutively over the same underlay interface pair instead of spread across multiple underlay interface pairs. Finally, an assured minimum/path MPS allows continuous operation over all paths including those that traverse bridged L2 media with dissimilar MTUs.

Note: Certain legacy network hardware of the past millennium was unable to accept packet "bursts" resulting from an IP fragmentation event - even to the point that the hardware would reset itself when presented with a burst. This does not seem to be a common problem in the modern era, where fragmentation and reassembly can be readily demonstrated at line rate (e.g., using tools such as 'iperf3') even over fast links on ordinary hardware platforms. Even so, while the OAL destination is reporting reassembly congestion (see: Section 6.8) the OAL source could impose "pacing" by inserting an inter-fragment delay and increasing or decreasing the delay according to congestion indications.

6.12. OAL Fragmentation Security Implications

As discussed in Section 3.7 of [RFC8900], there are four basic threats concerning IPv6 fragmentation; each of which is addressed by effective mitigations as follows:

1. Overlapping fragment attacks - reassembly of overlapping fragments is forbidden by [RFC8200]; therefore, this threat does not apply to the OAL.
2. Resource exhaustion attacks - this threat is mitigated by providing a sufficiently large OAL reassembly cache and instituting "fast discard" of incomplete reassemblies that may be part of a buffer exhaustion attack. The reassembly cache should be sufficiently large so that a sustained attack does not cause excessive loss of good reassemblies but not so large that (timer-based) data structure management becomes computationally expensive. The cache should also be indexed based on the arrival underlay interface such that congestion experienced over a first underlay interface does not cause discard of incomplete reassemblies for uncongested underlay interfaces.
3. Attacks based on predictable fragment identification values - in environments where spoofing is possible, this threat is mitigated through the use of Identification windows beginning with unpredictable values per Section 6.6. By maintaining windows of acceptable Identifications, OAL neighbors can quickly discard spurious carrier packets that might otherwise clutter the reassembly cache. The OAL additionally provides an integrity check to detect corruption that may be caused by spurious fragments received with in-window Identification values.
4. Evasion of Network Intrusion Detection Systems (NIDS) - since the OAL source employs a robust MPS, network-based firewalls can inspect and drop OAL fragments containing malicious data thereby disabling reassembly by the OAL destination. However, since OAL

fragments may take different paths through the network (some of which may not employ a firewall) each OAL destination must also employ a firewall.

IPv4 includes a 16-bit Identification (IP ID) field with only 65535 unique values such that at high data rates the field could wrap and apply to new carrier packets while the fragments of old packets using the same IP ID are still alive in the network [RFC4963]. Since carrier packets sent via an IPv4 path with DF=0 are normally no larger than 576 octets, IPv4 fragmentation is possible only at small-MTU links in the path which should support data rates low enough for safe reassembly [RFC3819]. (IPv4 carrier packets larger than 576 octets with DF=0 may incur high data rate reassembly errors in the path, but the OAL checksum provides OAL destination integrity assurance.) Since IPv6 provides a 32-bit Identification value, IP ID wraparound at high data rates is not a concern for IPv6 fragmentation.

Fragmentation security concerns for large IPv6 ND messages are documented in [RFC6980]. These concerns are addressed when the OMNI interface employs the OAL instead of directly fragmenting the IPv6 ND message itself. For this reason, OMNI interfaces MUST NOT send IPv6 ND messages larger than the OMNI interface MTU, and MUST employ OAL encapsulation and fragmentation for IPv6 ND messages larger than the minimum/path MPS for this OAL destination.

Unless the path is secured at the network-layer or below (i.e., in environments where spoofing is possible), OMNI interfaces MUST NOT send ordinary carrier packets with Identification values outside the current window and MUST secure IPv6 ND messages used for address resolution or window state synchronization. OAL destinations SHOULD therefore discard without reassembling any out-of-window OAL fragments received over an unsecured path.

6.13. OMNI Hosts

OMNI Hosts are end systems that extend the OMNI link over ENET underlay interfaces (i.e., either as an OMNI interface or as a sublayer of the ENET interface itself). Each ENET is connected to the rest of the OMNI link by a Client that receives an MNP delegation. Clients delegate MNP addresses and/or sub-prefixes to ENET nodes (i.e., Hosts, other Clients, routers and non-OMNI hosts) using standard mechanisms such as DHCP [RFC8415][RFC2131] and IPv6 Stateless Address AutoConfiguration (SLAAC) [RFC4862]. Clients forward packets between their ENET Hosts and peers on external networks acting as routers and/or OAL intermediate nodes.

OMNI Hosts coordinate with Clients and/or other Hosts connected to the same ENET using IP-encapsulated IPv6 ND messages. The IP encapsulation headers and ND messages both use the MNP-based addresses assigned to ENET underlay interfaces as source and destination addresses (i.e., instead of ULAs). For IPv4 MNPs, the ND messages use IPv4-Compatible IPv6 addresses [RFC4291] in place of the IPv4 addresses. (Note that IPv4-Compatible IPv6 addresses are deprecated for all other uses by the aforementioned standard.)

Hosts discover Clients by sending encapsulated RS messages using an OMNI link IP anycast address (or the unicast address of the Client) as the RS L2 encapsulation destination as specified in Section 15. The Client configures the IPv4 and/or IPv6 anycast addresses for the OMNI link on its ENET interface and advertises the address(es) into the ENET routing system. The Client then responds to the encapsulated RS messages by sending an encapsulated RA message that uses its ENET unicast address as the source. (To differentiate itself from an INET border Proxy/Server, the Client sets the RA message OMNI Interface Attributes sub-option LHS field to 0 for the Host's interface index. When the RS message includes an L2 anycast destination address, the Client also includes an Interface Attributes sub-option for interface index 0 to inform the Host of its L2 unicast address - see: Section 15 for full details on the RS and RA message contents.)

Hosts coordinate with peer Hosts on the same ENET by sending encapsulated NS messages to receive an NA reply. (Hosts determine whether a peer is on the same ENET by matching the peer's IP address with the MNP (sub)-prefix for the ENET advertised in the Client's RA message [RFC8028].) Each ENET peer then creates a NCE and synchronizes Identification windows the same as for OMNI link neighbors, and the Host can then engage in OMNI link transactions with the Client and/or other ENET Hosts. By coordinating with the Client in this way, the Host treats the Client as if it were an ANET Proxy/Server, and the Client provides the same services that a Proxy/Server would provide. By coordinating with other Hosts, the peer hosts can exchange large IP packets or parcels over the ENET using IPv6 fragmentation if necessary.

When a Host prepares an IP packet or parcel, it uses the IP address of its native ENET interface as the source and the IP address of the (remote) peer as the destination. The Host next performs parcel segmentation if necessary (see: Section 6.14) then encapsulates the packet/parcel in an IP header of the version supported by the ENET while setting the source to the same address and destination to either the same address if the peer is on the local ENET, or to the IP address of the Client otherwise. The Host can then proceed to exchange packets/parcels with the destination, either directly or via the Client as an intermediate node.

The encapsulation procedures are coordinated per Section 6.1, except that the IP encapsulation header version matches the native ENET IP protocol version and uses IPv6 GUA or public/private IPv4 addresses instead of ULAs. The Host sets the encapsulation IP header {Protocol, Next-Header} field to TBD1 to indicate that this is an OAL encapsulation and not an ordinary IP-in-IP encapsulation. When the inner header is IPv4-based, the Host next translates the encapsulation header into an IPv6 header with IPv4-Compatible addresses while setting the [IPv6 Traffic Class, Payload Length, Next Header, Hop Limit] fields according to the IPv4 {Type of Service, Total Length, Protocol, TTL} fields, respectively, while setting Flow Label to 0. The Host then calculates an OAL checksum, writes the value as the final two octets of the encapsulated packet then applies IPv6 fragmentation to the encapsulated packet to produce IPv6 fragments no smaller than the MPS the same as described in Section 6.1. If the original encapsulation IP header was IPv4, the Host next translates the IPv6 encapsulation headers back to IPv4 headers with Protocol value set to 44 since the immediately next header is the IPv6 Fragment Header. The Host finally sends the IP encapsulated fragments to the ENET peer.

When the ENET peer receives IP encapsulated fragments, for IPv4 it first translates the encapsulation headers back to IPv6 headers with IPv4-Compatible addresses the same as above. The peer then reassembles and verifies the OAL checksum. If the checksum is correct, the peer next removes the encapsulation headers and applies parcel reassembly if necessary. The peer then either delivers the encapsulated packet/parcel to upper layers if the peer is the destination or forwards the packet/parcel toward the final destination if the peer is a Client acting as an intermediate node.

Hosts and Clients that initiate OMNI-based packet/parcel transactions should first test the path toward the final destination using the parcel path qualification procedure specified in [I-D.templin-intarea-parcels]. An OMNI Host that sends and receives parcels need not implement the full OMNI interface abstraction but MUST implement enough of the OAL to be capable of fragmenting and reassembling maximum-length encapsulated IP packets/parcels and sub-parcels as discussed above and in the following section.

6.14. IP Parcels

IP parcels are specified in [I-D.templin-intarea-parcels], while details for their application over OMNI interfaces is specified here. IP parcels are formed by an OMNI Host or Client upper layer protocol entity (identified by the "5-tuple" source IP address/port number, destination IP address/port number and protocol number) when it produces a protocol data unit containing the concatenation of up to 64 upper layer protocol segments. All non-final segments MUST be equal in length while the final segment MUST NOT be larger but MAY be smaller. Each non-final segment MUST be no larger than 65535 minus the length of the IP header plus extensions, minus the length of the OAL encapsulation header and trailer. The upper layer protocol then presents the buffer and non-final segment size to the IP layer which appends a single IP header (plus any extension headers) before presenting the parcel to the OMNI Interface.

For IPv4, the IP layer prepares the parcel by appending an IPv4 header with a Jumbo Payload option (see: Section 5.1) where "Jumbo Payload Length" is a 32-bit unsigned integer value (in network byte order) set to the lengths of the IPv4 header plus all concatenated segments. The IP layer next sets the IPv4 header DF bit to 1, then sets the IPv4 header Total Length field to the length of the IPv4 header plus the length of the first segment only. (Note: the IP layer can form true IPv4 jumbograms (as opposed to parcels) by instead setting the Total Length field to the length of the IPv4 header only.)

For IPv6, the IP layer forms a parcel by appending an IPv6 header with a Jumbo Payload option the same as for IPv4 above where "Jumbo Payload Length" is set to the lengths of the IPv6 Hop-by-Hop Options header and any other extension headers present plus all concatenated segments. The IP layer next sets the IPv6 header Payload Length field to the lengths of the IPv6 Hop-by-Hop Options header and any other extension headers present plus the length of the first segment only. (Note: the IP layer can form true IPv6 jumbograms (as opposed to parcels) by instead setting the Payload Length field to 0.)

An IP parcel therefore has the following structure:

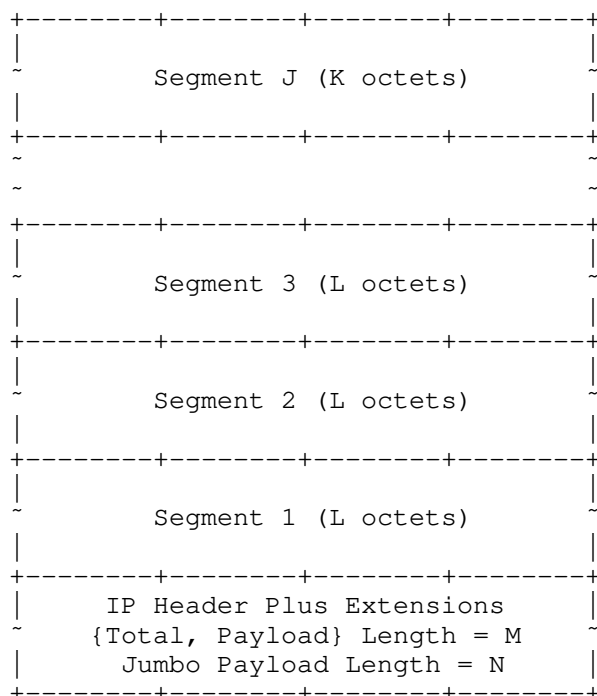


Figure 11: OMNI Interface IP Parcels

where J is the total number of segments (between 1 and 64), L is the length of each non-final segment which MUST NOT be larger than 65535 (minus headers as above) and K is the length of the final segment which MUST NOT be larger than L. The values M and N are then set to the length of the IP header plus extensions for IPv4 or to the length of the extensions only for IPv6, then further calculated as follows:

$$M = M + ((J-1) * L + K)$$

$$N = N + (((J-1) * L) + K)$$

Note: a "singleton" parcel is one that includes only the IP header plus extensions with a single segment of length K, while a "null" parcel is a singleton with K=0, i.e., a parcel consisting of only the IP header plus extensions with no octets beyond.

When the IP layer forwards a parcel, the OMNI interface invokes the OAL which forwards it to either a Client as an intermediate node or the final destination itself. The OAL source first assigns a monotonically-incrementing (modulo 127) "Parcel ID" and subdivides the parcel into sub-parcels no larger than the maximum of the path

MTU to the next hop or 64KB (minus the length of encapsulation headers). The OAL source determines the number of segments of length L that can fit into each sub-parcel under these size constraints, e.g. if the OAL source determines that a sub-parcel can contain 3 segments of length L, it creates sub-parcels with the first containing segments 1-3, the second containing segments 4-6, etc. and with the final containing any remaining segments. The OAL source then appends an identical IP header plus extensions to each sub-parcel while resetting M and N in each according to the above equations with J set to 3 and K set to L for each non-final sub-parcel and with J set to the remaining number of segments for the final sub-parcel.

The OAL source next performs encapsulation on each sub-parcel with destination set to the next hop address. If the next hop is reached via an ANET/INET interface, the OAL source inserts an OAL header the same as discussed in Section 6.1 and sets the destination to the ULA-MNP of the target Client. If the next hop is reached via an ENET interface, the OAL source instead inserts an IP header of the appropriate protocol version for the underlay ENET (i.e., even if the encapsulation header is IPv4) and sets the destination to the ENET IP address of the next hop. The OAL source inserts the encapsulation header even if no actual fragmentation is needed and/or even if the Jumbo Payload option is present.

The OAL source next assigns an Identification number that is monotonically-incremented for each consecutive sub-parcel, calculates and appends the OAL checksum, then performs IPv6 fragmentation over the sub-parcel if necessary to create fragments small enough to traverse the path to the next hop. (If the encapsulation header is IPv4, the OAL source first translates the encapsulation header into an IPv6 header with IPv4-Compatible IPv6 addresses before performing the fragmentation/reassembly operation while inserting an IPv6 Fragment Header.) The OAL source then writes the "Parcel ID" and sets/clears the "(P)arcel" and "(More) (S)ub-Parcels" bits in the Fragment Header of the first fragment (see: Figure 5). (The OAL source sets P to 1 for a parcel or to 0 for a non-parcel. When P is 1, the OAL next sets S to 1 for non-final sub-parcels or to 0 if the sub-parcel contains the final segment.) The OAL source then forwards each IP encapsulated packet/fragment to the next hop (i.e., after first translating the IPv6 encapsulation header back to IPv4 if necessary).

When the next hop receives the encapsulated IP fragments or whole packets, it acts as an OAL destination and reassembles if necessary (i.e., after first translating the IPv4 encapsulation header to IPv6 if necessary). If the P flag in the first fragment is 0, the OAL destination then processes the reassembled entity as an ordinary IP

packet; otherwise it continues processing as a sub-parcel. If the OAL destination is not the final destination, it retains the sub-parcels along with their Parcel ID and Identification values for a brief time in hopes of re-combining with peer sub-parcels of the same original parcel identified by the 4-tuple consisting of the IP encapsulation source and destination, Identification and Parcel ID. The OAL destination re-combines peers by concatenating the segments included in sub-parcels with the same Parcel ID and with Identification values within 64 of one another to create a larger sub-parcel possibly even as large as the entire original parcel. Order of concatenation is not important, with the exception that the final sub-parcel (i.e., the one with S set to 0) must occur as the final concatenation before transmission. The OAL destination then appends a common IP header plus extensions to each re-combined sub-parcel while resetting M and N in each according to the above equations with J, K and L set accordingly.

When the current OAL destination is an intermediate node, it next becomes an OAL source to forward the re-combined (sub-)parcel(s) to the next hop toward the final destination using encapsulation/translation the same as specified above. (Each such intermediate node MUST ensure that the S flag remains set to 0 in the sub-parcel that contains the final segment.) When the parcel or sub-parcels arrive at the final OAL destination, it re-combines them into the largest possible (sub)-parcels while honoring the S flag then delivers them to upper layers which act on the enclosed 5-tuple information supplied by the original source.

Note: while the final destination may be tempted to re-combine the sub-parcels of multiple different parcels with identical upper layer protocol 5-tuples and with non-final segments of identical length, this process could become complicated when the different parcels each have final segments of diverse lengths. Since this could possibly defeat any perceived performance advantages, the decision of whether and how to perform inter-parcel concatenation is an implementation matter.

7. Frame Format

When the OMNI interface forwards original IP packets from the network layer it first invokes the OAL to create OAL packets/fragments if necessary, then includes any L2 encapsulations and finally engages the native frame format of the underlay interface. For example, for Ethernet-compatible interfaces the frame format is specified in [RFC2464], for aeronautical radio interfaces the frame format is specified in standards such as ICAO Doc 9776 (VDL Mode 2 Technical Manual), for various forms of tunnels the frame format is found in the appropriate tunneling specification, etc.

See Figure 2 for a map of the various L2 layering combinations possible. For any layering combination, the final layer (e.g., UDP, IP, Ethernet, etc.) must have an assigned number and frame format representation that is compatible with the selected underlay interface.

8. Link-Local Addresses (LLAs)

[RFC4861] requires that nodes assign Link-Local Addresses (LLAs) to all interfaces, and that routers use their LLAs as the source address for RA and Redirect messages. OMNI interfaces honor the first requirement, but do not honor the second since the OMNI link could consist of the concatenation of multiple links with diverse ULA prefixes (see Section 9) but for which multiple nodes might configure identical interface identifiers (IIDs). OMNI interface LLAs are therefore considered only as context for IID formation as discussed below and have no other operational role.

OMNI interfaces assign IPv6 LLAs through pre-service administrative actions. Clients assign "LLA-MNPs" with IIDs that embed the Client's unique MNP, while Proxy/Servers assign "LLA-RNDs" that include a randomly-generated IIDs generated as specified in [RFC7217]. LLAs are configured as follows:

- * IPv6 LLA-MNPs encode the most-significant 64 bits of an MNP within the least-significant 64 bits of the IPv6 link-local prefix fe80::/64, i.e., in the IID portion of the LLA. The LLA prefix length is determined by adding 64 to the MNP prefix length. e.g., for the MNP 2001:db8:1000:2000::/56 the corresponding LLA-MNP prefix is fe80::2001:db8:1000:2000/120. (The base LLA-MNP for each "/N" prefix sets the final 128-N bits to 0, but all LLA-MNPs that match the prefix are also accepted.) Non-MNP IPv6 prefix-based LLAs are also represented the same as for LLA-MNPs, but include a GUA prefix that is not properly covered by the MSP.
- * IPv4-Compatible LLA-MNPs are constructed as fe80::{IPv4-Prefix}, i.e., the IID consists of 32 '0' bits followed by a 32 bit IPv4 address/prefix, which may be either public or private in correspondence with the network layer addressing plan. The IPv4-Compatible LLA-MNP prefix length is determined by adding 96 to the IPv4 prefix length. For example, the IPv4-Compatible LLA-MNP for 192.0.2.0/24 is fe80::192.0.2.0/120, also written as fe80::c000:0200/120. (The base LLA-MNP for each "/N" prefix sets the final 128-N bits to 0, but all LLA-MNPs that match the prefix are also accepted.) Non-MNP IPv4 prefix-based LLAs are also represented the same as for LLA-MNPs, but include a GUA prefix that is not properly covered by the MSP.

- * LLA-RNDs are randomly-generated and assigned to Proxy/Servers and other SRT infrastructure elements. They may also be assigned by Clients to support the MNP delegation process. The upper 72 bits of the LLA-RND encode the prefix fe80::/72, and the lower 56 bits include a randomly-generated candidate pseudo-random value configured as specified in [RFC7217]; if the most significant 24 bits of the 56 bit candidate encodes the value '0', the node generates a new candidate to obtain one with a different most significant 24 bits to avoid overlap with IPv4-Compatible LLAs.
- * The address fe80::/128 (i.e., the LLA /64 prefix followed by an all-zero IID) is considered the LLA Subnet Router Anycast address

Since the prefix 0000::/8 is "Reserved by the IETF" [RFC4291], no MNPs can be allocated from that block ensuring that there is no possibility for overlap between the different MNP and RND LLA constructs discussed above.

Since LLA-MNPs are based on the distribution of administratively assured unique MNPs, and since LLA-RNDs are assumed unique through pseudo-random assignment, OMNI interfaces set the autoconfiguration variable DupAddrDetectTransmits to 0 [RFC4862].

Note: If future protocol extensions relax the 64-bit boundary in IPv6 addressing, the additional prefix bits of an MNP could be encoded in bits 16 through 63 of the LLA-MNP. (The most-significant 64 bits would therefore still be in bits 64-127, and the remaining bits would appear in bits 16 through 48.) However, this would interfere with the relationship between OMNI LLAs and ULAs (see: Section 9) and render many OMNI functions inoperable. The analysis provided in [RFC7421] furthermore suggests that the 64-bit boundary will remain in the IPv6 architecture for the foreseeable future.

9. Unique-Local Addresses (ULAs)

OMNI links use IPv6 Unique-Local Addresses (ULAs) as the source and destination addresses in both IPv6 ND messages and OAL packet IPv6 encapsulation headers. ULAs are routable only within the scope of an OMNI link, and are derived from the IPv6 Unique Local Address prefix fd00::/8 (i.e., the prefix fc00::/7 followed by the L bit set to 1). When the first 16 bits of the ULA encode the value fd00::/16, the address is considered as either a Temporary ULA (TLA) or an eXtended ULA (XLA) - see below. For all other ULAs, the 56 bits following fd00::/8 encode a 40-bit Global ID followed by a 16-bit Subnet ID as specified in Section 3 of [RFC4193]. All OMNI link ULA types finally include a 64-bit value in the IID portion of the address ULA::/64 as specified below.

When a node configures a ULA for OMNI, it selects a 40-bit Global ID for the OMNI link initialized to a candidate pseudo-random value as specified in Section 3 of [RFC4193]; if the most significant 8 bits of the candidate encodes the value '0', the node selects a new candidate until it obtains one with a different most significant 8 bits. All nodes on the same OMNI link use the same Global ID, and statistical uniqueness of the pseudo-random Global ID provides a unique OMNI link identifier allowing different links to be joined together in the future without requiring renumbering.

Next, for each logical segment of the same OMNI link the node selects a 16-bit Subnet ID value between 0x0000 and 0xffff. Nodes on the same logical segment configure the same Subnet ID, but nodes on different segments of the same OMNI link can still exchange IPv6 ND messages as single-hop neighbors even if they configure different Subnet IDs. When a node moves to a different OMNI link segment, it resets the Global ID and Subnet ID value according to the new segment but need not change the IID.

ULAs and their associated prefix lengths are configured in correspondence with LLAs through stateless prefix translation where "ULA-MNPs" simply copy the IIDs of their corresponding LLA-MNPs and "ULA-RNDs" simply copy the IIDs of their corresponding LLA-RNDs. For example, for the OMNI link ULA prefix `fd{Global}:{Subnet}::/64`:

- * the ULA-MNP corresponding to the LLA-MNP `fe80::2001:db8:1:2` with a 56-bit MNP length is simply `fd{Global}:{Subnet}:2001:db8:1:2/120` (where, the ULA prefix length becomes 64 plus the IPv6 MNP length).
- * the ULA-MNP corresponding to `fe80::192.0.2.0` with a 28-bit MNP length is simply `fd{Global}:{Subnet}::192.0.2.0/124` (where, the ULA prefix length becomes 96 plus the IPv4 MNP length).
- * the ULA-RND corresponding to `fe80::0012:3456:789a:bcde` is simply `fd{Global}:{Subnet}::0012:3456:789a:bcde/128`.
- * the Subnet Router Anycast ULA corresponding to `fe80::/128` is simply `fd{Global}:{Subnet}::/128`.

The ULA presents an IPv6 address format that is routable within the OMNI link routing system and can be used to convey link-scoped (i.e., single-hop) IPv6 ND messages across multiple hops through IPv6 encapsulation [RFC2473]. The OMNI link extends across one or more underlying Internetworks to include all Proxy/Servers and other service nodes. All Clients are also considered to be connected to the OMNI link, however unnecessary encapsulations are omitted whenever possible to conserve bandwidth (see: Section 14).

Clients can configure TLAs when they have no other ULA addresses by setting the ULA prefix to fd00::/16 followed by a 48-bit randomly-generated number followed by a random or MNP-based IID as discussed in Section 8. XLAs are a special-case TLA that use the prefix fd00::/64. (Note that XLAs can also be formed from LLAs simply by inverting bits 7 and 8 of 'fe80' to form 'fd00'.)

OMNI nodes use XLA-MNPs as "default" ULAs for representing MNPs in the OMNI link routing system. Clients use {TLA,XLA}-MNPs when they already know their MNP but need to express it outside the context of a specific ULA prefix, and Proxy/Servers advertise XLA-MNPs into the OMNI link routing system instead of advertising fully-qualified {TLA,ULA}-MNPs and/or non-routable LLA-MNPs.

{TLAs,XLAs} provide initial "bootstrapping" addresses while the Client is in the process of procuring an MNP and/or identifying the ULA prefix for the OMNI link segment; TLAs are not advertised into the OMNI link routing system but can be used for Client-to-Client communications within a single {A,I,E}NET when no OMNI link infrastructure is present. Within each individual {A,I,E}NET, TLAs employ optimistic DAD principles [RFC4429] since they are statistically unique.

Each OMNI link may be subdivided into SRT segments that often correspond to different administrative domains or physical partitions. Each SRT segment is identified by a different Subnet ID within the same ULA ::/48 prefix. Multiple distinct OMNI links with different ULA ::/48 prefixes can also be joined together into a single unified OMNI link through simple interconnection without requiring renumbering. In that case, the (larger) unified OMNI link routing system may carry multiple distinct ULA prefixes.

OMNI nodes can use Segment Routing [RFC8402] to support efficient forwarding to destinations located in other OMNI link segments. A full discussion of Segment Routing over the OMNI link appears in [I-D.templin-6man-aero].

Note: IPv6 ULAs taken from the prefix fc00::/7 followed by the L bit set to 0 (i.e., as fc00::/8) are never used for OMNI OAL addressing, however the range could be used for MSP/MNP addressing under certain limiting conditions (see: Section 10). When used within the context of OMNI, ULAs based on the prefix fc00::/8 are referred to as "ULA-C's".

Note: When they appear in the OMNI link routing table, ULA-RNDs always use prefix lengths between /48 and /64 (or, /128) while XLA-MNPs always use prefix lengths between /65 and /128. {TLA,ULA}-MNPs and {TLA,XLA}-RNDs should never appear in the OMNI link routing table, but may appear in {A,I,E}NET routing tables.

10. Global Unicast Addresses (GUAs)

OMNI domains use IP Global Unicast Address (GUA) prefixes [RFC4291] as Mobility Service Prefixes (MSPs) from which Mobile Network Prefixes (MNP) are delegated to Clients. Fixed correspondent node networks reachable from the OMNI link are represented by non-MNP GUA prefixes that are not derived from the MSP, but are treated in all other ways the same as for MNPs.

For IPv6, GUA MSPs are assigned by IANA [IPV6-GUA] and/or an associated Regional Internet Registry (RIR) such that the OMNI link can be interconnected to the global IPv6 Internet without causing inconsistencies in the routing system. An OMNI link could instead use ULAs with the 'L' bit set to 0 (i.e., from the prefix fc00::/8) [RFC4193], however this would require IPv6 NAT if the domain were ever connected to the global IPv6 Internet.

For IPv4, GUA MSPs are assigned by IANA [IPV4-GUA] and/or an associated RIR such that the OMNI link can be interconnected to the global IPv4 Internet without causing routing inconsistencies. An OMNI ANET/ENET could instead use private IPv4 prefixes (e.g., 10.0.0.0/8, etc.) [RFC3330], however this would require IPv4 NAT at the INET-to-ANET/ENET boundary. OMNI interfaces advertise IPv4 MSPs into IPv6 routing systems as IPv4-Compatible IPv6 prefixes [RFC4291] (e.g., the IPv6 prefix for the IPv4 MSP 192.0.2.0/24 is ::192.0.2.0/120).

OMNI interfaces assign the IPv4 anycast address TBD3 (see: IANA Considerations), and IPv4 routers that configure OMNI interfaces advertise the prefix TBD3/N into the routing system of other networks (see: IANA Considerations). OMNI interfaces also configure global IPv6 anycast addresses formed according to [RFC3056] as:

2002:TBD3{32}:MSP{64}:Link-ID{16}

where TBD3{32} is the 32 bit IPv4 anycast address and MSP{64} encodes an MSP zero-padded to 64 bits (if necessary). For example, the OMNI IPv6 anycast address for MSP 2001:db8::/32 is 2002:TBD3{32}:2001:db8:0:0:{Link-ID}, the OMNI IPv6 anycast address for MSP 192.0.2.0/24 is 2002:TBD3{32}::c000:0200:{Link-ID}, etc.).

The 16-bit Link-ID in the OMNI IPv6 anycast address identifies a specific OMNI link within the domain that services the MSP. The special Link-ID value '0' is a wildcard that matches all links, while all other values identify specific links. Mappings between Link-ID values and the ULA Global IDs assigned to OMNI links are outside the scope of this document.

OMNI interfaces assign OMNI IPv6 anycast addresses, and IPv6 routers that configure OMNI interfaces advertise the corresponding prefixes into the routing systems of other networks. An OMNI IPv6 anycast prefix is formed the same as for any IPv6 prefix; for example, the prefix 2002:TBD3{32}:2001:db8::/80 matches all OMNI IPv6 anycast addresses covered by the prefix. When IPv6 routers advertise OMNI IPv6 anycast prefixes in this way, Clients can locate and associate with either a specific OMNI link or any OMNI link within the domain that services the MSP of interest.

OMNI interfaces use OMNI IPv6 and IPv4 anycast addresses to support Service Discovery in the spirit of [RFC7094], i.e., the addresses are not intended for use in long-term transport protocol sessions. Specific applications for OMNI IPv6 and IPv4 anycast addresses are discussed throughout the document as well as in [I-D.templin-6man-aero].

11. Node Identification

OMNI Clients and Proxy/Servers that connect over open Internet networks include a unique node identification value for themselves in the OMNI options of their IPv6 ND messages (see: Section 12.2.12). An example identification value alternative is the Host Identity Tag (HIT) as specified in [RFC7401], while Hierarchical HITs (HHITs) [I-D.ietf-drip-rid] may be more appropriate for certain domains such as the Unmanned (Air) Traffic Management (UTM) service for Unmanned Air Systems (UAS). Another example is the Universally Unique Identifier (UUID) [RFC4122] which can be self-generated by a node without supporting infrastructure with very low probability of collision.

When a Client is truly outside the context of any infrastructure, it may have no MNP information at all. In that case, the Client can use a TLA or (H)HIT as an IPv6 source/destination address for sustained communications in Vehicle-to-Vehicle (V2V) and (multihop) Vehicle-to-Infrastructure (V2I) scenarios. The Client can also propagate the ULA/(H)HIT into the multihop routing tables of (collective) Mobile/Vehicular Ad-hoc Networks (MANETs/VANETs) using only the vehicles themselves as communications relays.

When a Client connects via a protected-spectrum ANET, an alternate form of node identification (e.g., MAC address, serial number, airframe identification value, VIN, etc.) embedded in a ULA may be sufficient. The Client can then include OMNI "Node Identification" sub-options (see: Section 12.2.12) in IPv6 ND messages should the need to transmit identification information over the network arise.

12. Address Mapping - Unicast

OMNI interfaces maintain a neighbor cache for tracking per-neighbor state and use the link-local address format specified in Section 8. IPv6 Neighbor Discovery (ND) [RFC4861] messages sent over OMNI interfaces without encapsulation observe the native underlay interface Source/Target Link-Layer Address Option (S/TLLAO) format (e.g., for Ethernet the S/TLLAO is specified in [RFC2464]). IPv6 ND messages sent over OMNI interfaces using encapsulation do not include S/TLLAOs, but instead include a new option type that encodes encapsulation addresses, interface attributes and other OMNI link information. Hence, this document does not define an S/TLLAO format but instead defines a new option type termed the "OMNI option" designed for these purposes. (Note that OMNI interface IPv6 ND messages sent without encapsulation may include both OMNI options and S/TLLAOs, but the information conveyed in each is mutually exclusive.)

OMNI interfaces prepare IPv6 ND messages that include one or more OMNI options (and any other IPv6 ND options) then completely populate all option information. If the OMNI interface includes an authentication signature, it sets the IPv6 ND message Checksum field to 0 and calculates the authentication signature over the length of the entire OAL packet or super-packet (beginning with a pseudo-header of the IPv6 ND message IPv6 header) but does not calculate/include the IPv6 ND message checksum itself. Otherwise, the OMNI interface calculates the standard IPv6 ND message checksum over the entire OAL packet or super-packet and writes the value in the Checksum field. OMNI interfaces verify authentication and/or integrity of each IPv6 ND message received according to the specific check(s) included, and process the message further only following verification.

OMNI interface Clients such as aircraft typically have multiple wireless data link types (e.g. satellite-based, cellular, terrestrial, air-to-air directional, etc.) with diverse performance, cost and availability properties. The OMNI interface would therefore appear to have multiple L2 connections, and may include information for multiple underlay interfaces in a single IPv6 ND message exchange. OMNI interfaces manage their dynamically-changing multilink profiles by including OMNI options in IPv6 ND messages as discussed in the following subsections.

12.1. The OMNI Option

OMNI options appear in IPv6 ND messages formatted as shown in Figure 12:

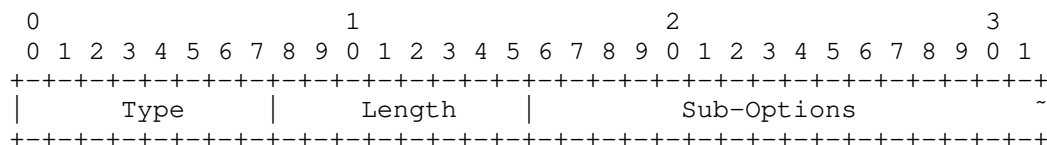


Figure 12: OMNI Option Format

In this format:

- * Type is set to TBD4 (see: IANA Considerations).
- * Length is set to the number of 8 octet blocks in the option. The value 0 is invalid, while the values 1 through 255 (i.e., 8 through 2040 octets, respectively) indicate the total length of the OMNI option. If multiple OMNI option instances appear in the same IPv6 ND message, the union of the contents of all OMNI options is accepted unless otherwise qualified for specific sub-options below.
- * Sub-Options is a Variable-length field padded if necessary such that the complete OMNI Option is an integer multiple of 8 octets long. Sub-Options contains zero or more sub-options as specified in Section 12.2.

The OMNI option is included in all OMNI interface IPv6 ND messages; the option is processed by receiving interfaces that recognize it and otherwise ignored. The OMNI interface processes all OMNI option instances received in the same IPv6 ND message in the consecutive order in which they appear. The OMNI option(s) included in each IPv6 ND message may include full or partial information for the neighbor. The OMNI interface therefore retains the union of the information in the most recently received OMNI options in the corresponding NCE.

12.2. OMNI Sub-Options

Each OMNI option includes a Sub-Options block containing zero or more individual sub-options. Each consecutive sub-option is concatenated immediately following its predecessor. All sub-options except Pad1 (see below) are in an OMNI-specific type-length-value (TLV) format encoded as follows:

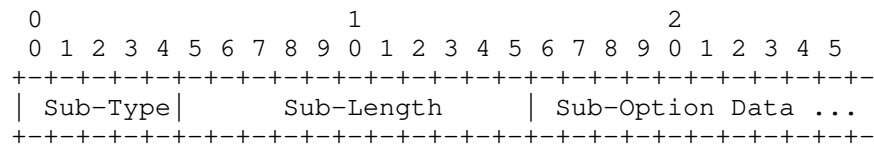


Figure 13: Sub-Option Format

* Sub-Type is a 5-bit field that encodes the sub-option type. Sub-option types defined in this document are:

Sub-Option Name	Sub-Type
Pad1	0
PadN	1
Neighbor Coordination	2
Interface Attributes	3
Multilink Forwarding Params	4
Traffic Selector	5
Geo Coordinates	6
DHCPv6 Message	7
HIP Message	8
PIM-SM Message	9
Fragmentation Report	10
Node Identification	11
ICMPv6 Error	12
QUIC-TLS Message	13
Proxy/Server Departure	14
Sub-Type Extension	30

Figure 14

Sub-Types 15-29 are available for future assignment for major protocol functions, while Sub-Type 30 supports scalable extension to include other functions. Sub-Type 31 is reserved by IANA.

- * Sub-Length is an 11-bit field that encodes the length of the Sub-Option Data in octets.
- * Sub-Option Data is a block of data with format determined by Sub-Type and length determined by Sub-Length. Note that each individual sub-option may end on an arbitrary octet boundary, whereas the OMNI option itself must include padding if necessary for 8-octet alignment.

The OMNI interface codes each sub-option with a 2 octet header that includes Sub-Type in the most significant 5 bits followed by Sub-Length in the next most significant 11 bits. Each sub-option encodes a maximum Sub-Length value of 2038 octets minus the lengths of the

OMNI option header and any preceding sub-options. This allows ample Sub-Option Data space for coding large objects (e.g., ASCII strings, domain names, protocol messages, security codes, etc.), while a single OMNI option is limited to 2040 octets the same as for any IPv6 ND option.

The OMNI interface codes initial sub-options in a first OMNI option instance and subsequent sub-options in additional instances in the same IPv6 ND message in the intended order of processing. The OMNI interface can then code any remaining sub-options in additional IPv6 ND messages if necessary. Implementations must observe these size limits and refrain from sending IPv6 ND messages larger than the OMNI interface MTU.

The OMNI interface processes all OMNI option Sub-Options received in an IPv6 ND message while skipping over and ignoring any unrecognized sub-options. The OMNI interface processes the Sub-Options of all OMNI option instances in the consecutive order in which they appear in the IPv6 ND message, beginning with the first instance and continuing through any additional instances to the end of the message. If an individual sub-option length would cause processing to exceed the OMNI option instance and/or IPv6 ND message lengths, the OMNI interface accepts any sub-options already processed and ignores the remainder of that instance. The interface then processes any remaining OMNI option instances in the same fashion to the end of the IPv6 ND message.

When an OMNI interface includes an authentication sub-option (e.g., see: Section 12.2.9), it MUST appear as the first sub-option of the first OMNI option which must appear immediately following the IPv6 ND message header (all other authentication sub-options are ignored). If the IPv6 ND message is the first packet in a combined OAL super-packet, the OMNI interface calculates the authentication signature over the entire length of the super-packet, i.e., and not just to the end of the IPv6 ND message itself. When the first sub-option is not authentication, the OMNI interface instead calculates the IPv6 ND message checksum over the entire length of the packet/super-packet.

When a Client OMNI interface prepares a secured unicast RS message, it includes an Interface Attributes sub-option specific to the underlay interface that will transmit the RS (see: Section 12.2.4) immediately following the authentication and header extension sub-options if present; otherwise as the first sub-option of the first OMNI option which must appear immediately following the IPv6 ND message header. When a Client OMNI interface prepares a secured unicast NS message, it instead includes a Multilink Forwarding Parameters sub-option specific to the underlay interface that will transmit the NS (see: Section 12.2.5).

Note: large objects that exceed the maximum Sub-Option Data length are not supported under the current specification; if this proves to be limiting in practice, future specifications may define support for fragmenting large sub-options across multiple OMNI options within the same IPv6 ND message (or even across multiple IPv6 ND messages, if necessary).

The following sub-option types and formats are defined in this document:

12.2.1. Pad1

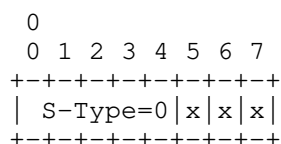


Figure 15: Pad1

- * Sub-Type is set to 0. If multiple instances appear in OMNI options of the same message all are processed.
- * Sub-Type is followed by 3 'x' bits, set to any value on transmission (typically all-zeros) and ignored on reception. Pad1 therefore consists of 1 octet with the most significant 5 bits set to 0, and with no Sub-Length or Sub-Option Data fields following.

If more than one octet of padding is required, the PadN option, described next, should be used, rather than multiple Pad1 options.

12.2.2. PadN

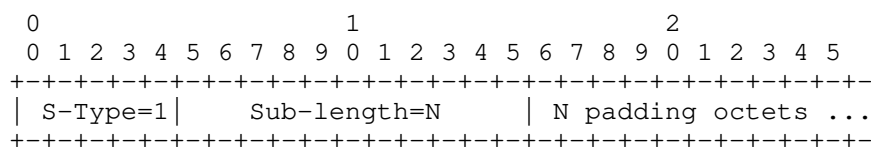


Figure 16: PadN

- * Sub-Type is set to 1. If multiple instances appear in OMNI options of the same message all are processed.
- * Sub-Length is set to N that encodes the number of padding octets that follow.

- * Sub-Option Data consists of N octets, set to any value on transmission (typically all-zeros) and ignored on receipt.

When a proxy forwards an IPv6 ND message with OMNI options, it can employ PadN to cancel any sub-options (other than Pad1) that should not be processed by the next hop by simply writing the value '1' over the Sub-Type. When the proxy alters the IPv6 ND message contents in this way, any included authentication and integrity checks are invalidated. See: Appendix B for a discussion of IPv6 ND message authentication and integrity.

12.2.3. Neighbor Coordination

IPv6 ND messages used for Prefix Length assertion, service coordination and/or Window Synchronization include a Neighbor Coordination sub-option. If a Neighbor Coordination sub-option is included, it must appear immediately after the authentication sub-option if present; otherwise, as the first (non-padding) sub-option of the first OMNI option. If multiple Neighbor Coordination sub-options are included (whether in a single OMNI option or multiple), only the first is processed and all others are ignored.

The Neighbor Coordination sub-option is formatted as follows:

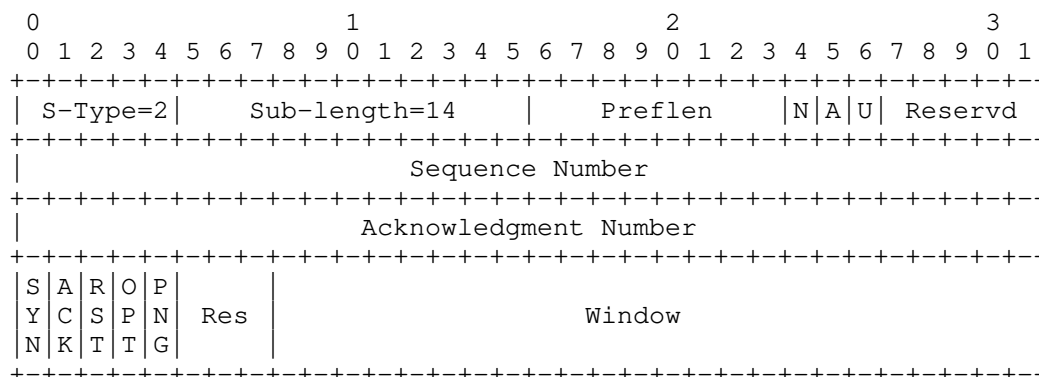


Figure 17: Neighbor Coordination

- * Sub-Type is set to 2.
- * Sub-Length is set to 14.
- * The first two octets of Sub-Option Data contains a 1-octet Prefix Length followed by a 1-octet flags field interpreted as follows:

- Preflen is an 8 bit field that determines the length of prefix associated with a ULA. Values 0 through 128 specify a valid prefix length (if any other value appears the OMNI option must be ignored). For IPv6 ND messages sent from a Client to the MS, Preflen applies to the IPv6 source ULA and provides the length that the Client is requesting from or asserting to the MS. For IPv6 ND messages sent from the MS to the Client, Preflen applies to the IPv6 destination ULA and indicates the length that the MS is granting to the Client. For IPv6 ND messages sent between MS endpoints, Preflen provides the length associated with the source/target Client MNP that is subject of the ND message. When an IPv6 ND RS/RA message sets Preflen to 0, the recipient regards the message as a prefix release indication.
- The N/A/U flags are set or cleared in Client RS messages as directives to FHS and Hub Proxy/Servers and ignored in all other IPv6 ND messages. When an FHS Proxy/Server forwards or processes an RS with the N flag set, it responds directly to NS Neighbor Unreachability Detection (NUD) messages by returning NA(NUD) replies; otherwise, it forwards NS(NUD) messages to the Client. When the Hub Proxy/Server receives an RS with the A flag set, it responds directly to NS Address Resolution (AR) messages by returning NA(AR) replies; otherwise, it forwards NS(AR) messages to the Client. When the Hub Proxy/Server receives an RS with the U flag set, it maintains a Report List of recent NS(AR) message sources for this Client and sends uNA messages to all list members if any aspects of the Client's underlay interfaces change. Proxy/Servers function according to the N/A/U flag settings received in the most recent RS message to support dynamic Client updates. In all IPv6 ND messages, the remaining 5 flag bits are set to 0 on transmission and ignored on reception.
- * The remainder of Sub-Option Data contains a 4-octet Sequence Number, followed by a 4-octet Acknowledgement Number, followed by a 1-octet flags field followed by a 3-octet Window size modeled from the Transmission Control Protocol (TCP) header specified in Section 3.1 of [RFC0793]. The (SYN, ACK, RST) flags are used for TCP-like window synchronization, while the TCP (URG, PSH, FIN) flags are not used and therefore omitted. The (OPT, PNG) flags are OMNI-specific, and the remaining flags are Reserved. Together, these fields support the asymmetric and symmetric OAL window synchronization services specified in Section 6.6.

12.2.4. Interface Attributes

The Interface Attributes sub-option provides neighbors with forwarding information for the multilink conceptual sending algorithm discussed in Section 14. Neighbors use the forwarding information to selecting among potentially multiple candidate underlay interfaces that can be used to forward carrier packets to the neighbor based on factors such as traffic selectors and link quality. Interface Attributes further include link-layer address information to be used for either direct INET encapsulation for targets in the local SRT segment or spanning tree forwarding for targets in remote SRT segments.

OMNI nodes include Interface Attributes for some/all of a target Client's underlay interfaces in NS/NA and uNA messages used to publish Client information (see: [I-D.templin-6man-aero]). At most one Interface Attributes sub-option for each distinct omIndex may be included; if an NS/NA message includes multiple Interface Attributes sub-options for the same omIndex, the first is processed and all others are ignored. OMNI nodes that receive NS/NA messages can use all of the included Interface Attributes and/or Traffic Selectors to formulate a map of the prospective target node as well as to seed the information to be populated in a Multilink Forwarding Parameters sub-option (see: Section 12.2.5).

OMNI Clients and Proxy/Servers also include Interface Attributes sub-options in RS/RA messages used to initialize, discover and populate routing and addressing information. Each RS message MUST contain exactly one Interface Attributes sub-option with an omIndex corresponding to the Client's underlay interface used to transmit the message, and each RA message MUST echo the same Interface Attributes sub-option with any (proxied) information populated by the FHS Proxy/Server to provide operational context.

OMNI Client RS and Proxy/Server RA messages MUST include the Interface Attributes sub-option for the Client underlay interface in the first OMNI option immediately following the Neighbor Coordination and/or authentication sub-option(s) if present; otherwise, immediately following the OMNI header. When an FHS Proxy/Server receives an RS message destined to an anycast L2 address, it MUST include an Interface Attributes sub-option with omIndex '0' that encodes its unicast L2 address relative to the Client's underlay interface immediately after the Interface Attributes sub-option in the solicited RA response. Any additional Interface Attributes sub-options that appear in RS/RA messages are ignored.

The Interface Attributes sub-options are formatted as shown below:

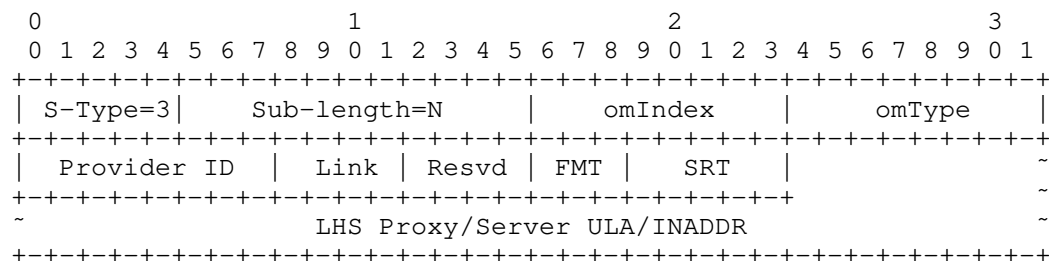


Figure 18: Interface Attributes

- * Sub-Type is set to 3.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow.
- * Sub-Option Data contains an "Interface Attributes" option encoded as follows:
 - omIndex is a 1-octet value corresponding to a specific underlay interface. Client OMNI interfaces MUST number each distinct underlay interface with an omIndex value between '1' and '255' that represents a Client-specific 8-bit mapping for the actual ifIndex value assigned by network management [RFC2863], then set omIndex to either a specific omIndex value or '0' to denote "unspecified".
 - omType is set to an 8-bit integer value corresponding to the underlay interface identified by omIndex. The value represents an OMNI interface-specific 8-bit mapping for the actual IANA ifType value registered in the 'IANAifType-MIB' registry [<http://www.iana.org>].
 - Provider ID is set to an OMNI interface-specific 8-bit ID value for the network service provider associated with this omIndex.
 - Link encodes a 4-bit link metric. The value '0' means the link is DOWN, and the remaining values mean the link is UP with metric ranging from '1' ("lowest") to '15' ("highest").
 - Resvd is a 4-bit Reserved field set to 0 on transmission and ignored on reception.
 - FMT - a 3-bit "Forward/Mode/Type" code interpreted as follows:

- o The most significant two bits (i.e., "FMT-Forward" and "FMT-Mode") are interpreted in conjunction with one another. When FMT-Forward is clear, the LHS Proxy/Server performs OAL reassembly and decapsulation to obtain the original IP packet before forwarding. If the FMT-Mode bit is clear, the LHS Proxy/Server then forwards the original IP packet at layer 3; otherwise, it invokes the OAL to re-encapsulate, re-fragment and forwards the resulting carrier packets to the Client via the selected underlay interface. When FMT-Forward is set, the LHS Proxy/Server forwards unsecured OAL fragments to the Client without reassembling, while reassembling secured OAL fragments before re-fragmenting and forwarding to the Client. If FMT-Mode is clear, all carrier packets destined to the Client must always be forwarded through the LHS Proxy/Server; otherwise the Client is eligible for direct forwarding over the open INET where it may be located behind one or more NATs.
- o The least significant bit (i.e., "FMT-Type") determines the length of the LHS Proxy/Server INADDR field. If FMT-Type is clear, INADDR includes a 4-octet IPv4 address; otherwise, a 16-octet IPv6 address. (Note: the INADDR "short form" minimizes overhead for ND messages that include many Interface Attributes sub-options with IPv4 addresses.)
- SRT - a 5-bit Segment Routing Topology prefix length value between 0 and 16 that (when added to 48) determines the prefix length associated with the LHS ULA Subnet ID. For example, the value 5 corresponds to the prefix ULA::/53.
- LHS Proxy/Server ULA/INADDR - the first 15 octets following the "FMT/SRT" octet includes the 120 least significant bits of the ULA of the LHS Proxy/Server on the path from a source neighbor to the target Client's underlay interface. (Note that the FMT/SRT code is replaced with the value "fd" after processing to form a proper Proxy/Server ULA.) When SRT and ULA are both set to 0, the LHS Proxy/Server is considered unspecified in this IPv6 ND message. FMT, SRT and LHS together provide guidance for the OMNI interface forwarding algorithm. Specifically, if SRT/LHS is located in the local OMNI link segment, then the source can reach the target Client either through its dependent Proxy/Server or through direct encapsulation following NAT traversal according to FMT. Otherwise, the target Client is located on a different SRT segment and the path from the source must employ a combination of route optimization and spanning tree hop traversals. INADDR identifies the LHS Proxy/Server's INET-facing interface not located behind NATs, therefore no UDP port number is included since port number 8060 is used when the

L2 encapsulation includes a UDP header. Instead, INADDR includes only a 4-octet IPv4 or 16-octet IPv6 address with type and length determined by FMT-Type. The IP address is recorded in network byte order in ones-compliment "obfuscated" form per [RFC4380].

12.2.5. Multilink Forwarding Parameters

OMNI nodes include the Multilink Forwarding Parameters sub-option in NS/NA messages used to coordinate with multilink route optimization targets. If an NS message includes the sub-option, the solicited NA response must also include the sub-option. The OMNI node MUST include the sub-option in the first OMNI option immediately following the Neighbor Coordination and/or authentication message sub-option(s) if present. Otherwise, the OMNI node MUST include the sub-option immediately following the OMNI header. Each NS/NA message may contain at most one Multilink Forwarding Parameters sub-option; if an NS/NA message contains additional Multilink Forwarding Parameters sub-options, the first is processed and all others are ignored.

When an NS/NA message includes the sub-option, the FHS Client omIndex MUST correspond to the underlay interface used to transmit the message. When the NS/NA message also includes Interface Attributes sub-options any that include the same FHS/LHS Client omIndex are ignored while all others are processed.

The Multilink Forwarding Parameters sub-option includes the necessary state for establishing Multilink Forwarding Vectors (MFVs) in the Multilink Forwarding Information Bases (MFIBs) of the OAL source, destination and intermediate nodes in the path. The sub-option also records addressing information for FHS/LHS nodes on the path, including "INADDRs" which MUST be unicast IP encapsulation addresses (i.e., and not anycast/multicast). The manner for populating multilink forwarding information is specified in detail in [I-D.templin-6man-aero].

The Multilink Forwarding Parameters sub-option is formatted as shown in Figure 19:

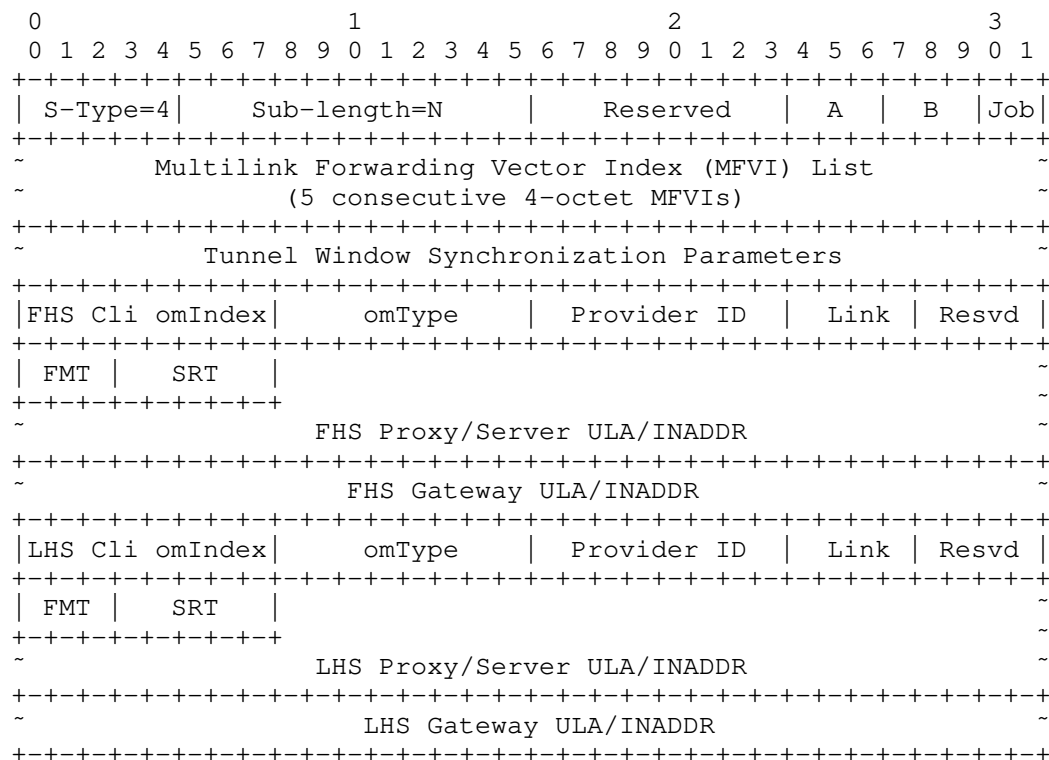


Figure 19: Multilink Forwarding Parameters

- * Sub-Type is set to 4. If multiple instances appear in the same message (i.e., whether in a single OMNI option or multiple) the first instance is processed and all others are ignored.
- * Sub-Length encodes the number of Sub-Option Data octets that follow. The length includes all fields up to and including the Tunnel Window Synchronization Parameters for all Job codes, while including the remaining fields only for Job codes "0" and "1" (see below).
- * Sub-Option Data contains Multilink Forwarding Parameters as follows:
 - Reserved is a 1-octet reserved field set to 0 on transmission and ignored on receipt.

- A/B and Job are fields that determine per-hop processing of the MFVI List, where A is a 3-bit count of the number of "A" MFVI List entries and B is a 3-bit count of the number of "B" MFVI List entries (valid A/B values are 0-5). Job is a 2-bit code interpreted as follows:
 - o '00' - "Initialize; Build B" - the FHS source sets this code in an NS used to initialize MFV state (any other messages that include this code MUST be dropped). The FHS source first sets A/B to 0, and the FHS source and each intermediate node along the path to the LHS destination that processes the message creates a new MFV. Each node that processes the message then assigns a unique 4-octet "B" MFVI to the MVE and also writes the value into list entry B, then increments B. When the message arrives at the LHS destination, B will contain the number of MFVI List "B" entries, with the FHS source entry first, followed by entries for each consecutive intermediate node and ending with an entry for the final intermediate node (i.e., the list is populated in the forward direction).
 - o '01' - "Follow B; Build A" - the LHS source sets this code in a solicited NA response to a solicitation with Job code "0" (any other messages that include this code MUST be dropped). The LHS source first copies the MFVI List and B value from the code "0" solicitation into these fields and sets A to 0. The LHS source and each intermediate node along the path to the FHS destination that processes the message then uses MFVI List entry B to locate the corresponding MFV. Each node that processes the message then assigns a unique 4-octet "A" MFVI to the MVE and also writes the value into list entry B, then increments A and decrements B. When the message arrives at the FHS destination, A will contain the number of MFVI List "A" entries, with the LHS source entry last, preceded by entries for each consecutive intermediate node and beginning with an entry for the final intermediate node (i.e., the list is populated in the reverse direction).
 - o '10' - "Follow A; Record B" - the FHS node that sent the original code "0" solicitation and received the corresponding code "1" advertisement sets this code in any subsequent NS/NA messages sent to the same LHS destination. The FHS source copies the MFVI List and A value from the code "1" advertisement into these fields and sets B to 0. The FHS source and each intermediate node along the path to the LHS destination that processes the message then uses the "A" MFVI found at list entry B to locate the corresponding

MFV. Each node that processes the message then writes the MFV's "B" MFVI into list entry B, then decrements A and increments B. When the message arrives at the LHS destination, B will contain the number of MFVI List "B" entries populated in the forward direction.

- o '11' - "Follow B; Record A" - the LHS node that received the original code "0" solicitation and sent the corresponding code "1" advertisement sets this code in any subsequent NS/NA messages sent to the same FHS destination. The LHS source copies the MFVI List and B values from the code "0" solicitation into these fields and sets A to 0. The LHS source and each intermediate node along the path to the FHS destination that processes the message then uses the "B" MFVI List entry found at list entry B to locate the corresponding MFV. Each node that processes the message then writes the MFV's "A" MFVI into list entry B, then increments A and decrements B. When the message arrives at the FHS destination, A will contain the number of MFVI List "A" entries populated in the reverse direction.

Job and A/B together determine the per-hop behavior at each FHS/LHS source, intermediate node and destination that processes an IPv6 ND message. When a Job code specifies "Initialize", each FHS/LHS node that processes the message creates a new MFV. When a Job code specifies "Build", each node that processes the message assigns a new MFVI. When a Job code specifies "Follow", each node that processes the message uses an A/B MFVI List entry to locate an MFV (if the MFV cannot be located, the node returns a parameter problem and drops the message). Using this algorithm, FHS sources that send code '00' solicitations and receive code '01' advertisements discover only "A" information, while LHS sources that receive code '00' solicitations and return code '01' advertisements discover only "B" information. FHS/LHS intermediate nodes can instead examine A, B and the MFVI List to determine the number of previous hops, the number of remaining hops, and the A/B MFVIs associated with the previous/remaining hops. However, no intermediate nodes will discover inappropriate A/B MFVIs for their location in the multihop forwarding chain. See: [I-D.templin-6man-aero] for further discussion on A/B MFVI processing.

- Multilink Forwarding Vector Index (MFVI) List is a 20-octet block that contains 5 consecutive 4-octet MFVI entries. The FHS/LHS source and each intermediate node on the path to the destination processes the list according to the Job and A/B codes (see above). Note that the reason the MFVI list contains

at most 5 entries is that only the FHS (Client, Proxy/Server, Gateway) and LHS (Client, Proxy/Server, Gateway) nodes are eligible for OMNI link route optimization resulting in at most 5 MFVIs "hops" that must be exposed. All other OMNI link nodes (i.e., downstream Clients that connect via an FHS/LHS Client) must forward through their upstream-dependent OMNI link neighbors without applying OMNI link route optimization.

- Tunnel Window Synchronization Parameters is a 12-octet block that consists of a 4-octet Sequence Number followed by a 4-octet Acknowledgement Number followed by a 1-octet Flags field followed by a 3-octet Window field (i.e., the same as for the OMNI header parameters). Tunnel endpoints use these parameters for simultaneous middlebox window synchronization in a single solicitation/advertisement message exchange.
- For Job codes '00' and '01' only, two trailing state variable blocks are included for First-Hop Segment (FHS) followed by Last-Hop Segment (LHS) network elements. When present, each block encodes the following information:
 - o Client omIndex, omType, Provider ID and Resvd/Link are 1-octet fields (at offset 0 from the beginning of the Sub-Option Data) that include link parameters for the Client underlay interface. These fields are populated based on information discovered in Interface Attributes sub-options included in earlier RS/RA and/or NS/NA exchanges.
 - o FMT/SRT is a 1-octet field with a 5-bit SRT prefix length that applies to all elements in the segment. The FMT-Forward/Mode bits determine the characteristics of the Proxy/Server relationship for this specific Client underlay interface (i.e., the same as described in Section 12.2.4), and the FMT-Type bits determine the IP address version for all INADDR fields relative to this SRT segment. Unlike the case for Interface Attributes, all INADDR fields are always 16 bits in length regardless of the IP protocol version with IPv4 INADDRs encoded as IPv4-Compatible IPv6 addresses [RFC4291]. (Note: the INADDR "long-form" is used exclusively since there may be no a priori knowledge of the IP address version used at each hop.) The IP address is recoded in network byte order, and in ones-compliment "obfuscated" form the same as described in Section 12.2.4.
 - o Proxy/Server ULA/INADDR includes a 15 octet value that encodes the 120 least significant bits of the Proxy/Server ULA followed by a 16 octet INADDR. (Note that the FMT/SRT code is replaced with the value "fd" after processing to

form a proper Proxy/Server ULA.) INADDR identifies an open INET interface not located behind NATs, therefore no UDP port number is included since port number 8060 is used when the L2 encapsulation includes a UDP header.

- o Gateway ULA/INADDR encodes a 16 octet ULA followed by a 16 octet INADDR exactly as for the Proxy/Server ULA/INADDR. (Note that the Gateway ULA simply encodes the value "fd" in the most significant bits, since the FMT/SRT code applies to both the Proxy/Server and Gateway.)

12.2.6. Traffic Selector

When used in conjunction with Interface Attributes and/or Multilink Forwarding Parameters information, the Traffic Selector sub-option provides forwarding information for the multilink conceptual sending algorithm discussed in Section 14.

IPv6 ND messages include Traffic Selectors for some or all of the source/target Client's underlay interfaces. Traffic Selectors for some or all of a target Client's underlay interfaces are also included in uNA messages used to publish Client information changes. See: [I-D.templin-6man-aero] for more information.

Traffic Selectors must be honored by all implementations in the format shown below:

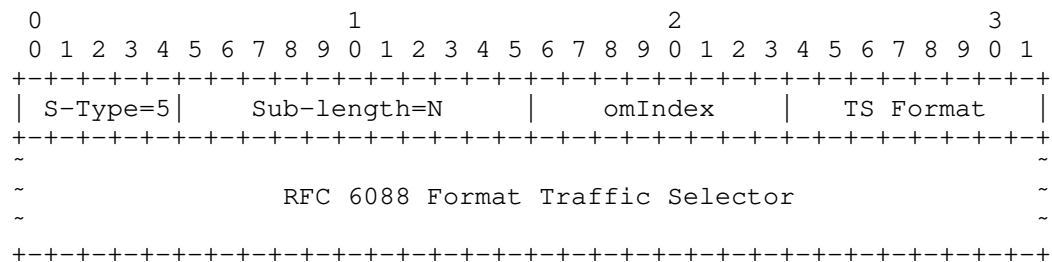


Figure 20: Traffic Selector

- * Sub-Type is set to 5. Each IPv6 ND message may contain zero or more Traffic Selectors for each omIndex; when multiple Traffic Selectors for the same omIndex appear, all are processed and the cumulative information from all is accepted.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow.
- * Sub-Option Data contains a "Traffic Selector" encoded as follows:

- omIndex is a 1-octet value corresponding to a specific underlay interface the same as specified above for Interface Attributes and Multilink Forwarding Parameters above. The OMNI options of a single message may include multiple Traffic Selector sub-options; each with the same or different omIndex values.
- TS Format is a 1-octet field that encodes a Traffic Selector version per [RFC6088]. If TS Format encodes the value 1 or 2, the Traffic Selector includes IPv4 or IPv6 information, respectively. If TS Format encodes any other value, the sub-option is ignored.
- The remainder of the sub-option includes a traffic selector formatted per [RFC6088] beginning with the "Flags (A-N)" field, and with the Traffic Selector IP protocol version coded in the TS Format field. If a single interface identified by omIndex requires Traffic Selectors for multiple IP protocol versions, or if a Traffic Selector block would exceed the available space, the remaining information is coded in additional Traffic Selector sub-options that all encode the same omIndex.

12.2.7. Geo Coordinates

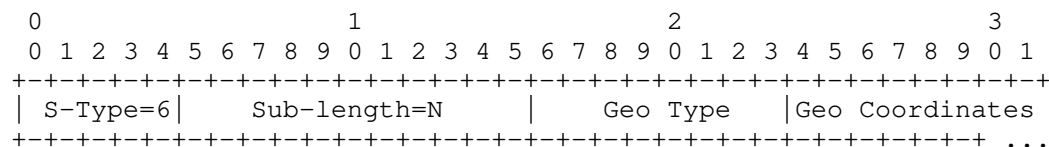


Figure 21: Geo Coordinates Sub-option

- * Sub-Type is set to 6. If multiple instances appear in OMNI options of the same message all are processed.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow.
- * Geo Type is a 1 octet field that encodes a type designator that determines the format and contents of the Geo Coordinates field that follows. The following types are currently defined:
 - 0 - NULL, i.e., the Geo Coordinates field is zero-length.
- * A set of Geo Coordinates of length up to the remaining available space for this OMNI option. New formats to be specified in future documents and may include attributes such as latitude/longitude, altitude, heading, speed, etc.

12.2.8. Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Message

The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) sub-option may be included in the OMNI options of Client RS messages and Proxy/Server RA messages. FHS Proxy/Servers that forward RS/RA messages between a Client and an LHS Proxy/Server also forward DHCPv6 Sub-Options unchanged. Note that DHCPv6 messages do not include a Checksum field since integrity is protected by the IPv6 ND message checksum, authentication signature and/or lower-layer authentication and integrity checks.

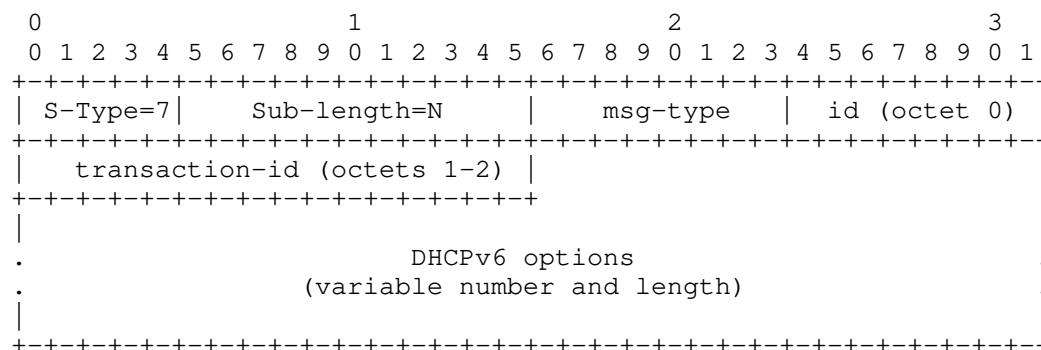


Figure 22: DHCPv6 Message Sub-option

- * Sub-Type is set to 7. If multiple instances appear in OMNI options of the same message the first is processed and all others are ignored.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow. The 'msg-type' and 'transaction-id' fields are always present; hence, the length of the DHCPv6 options is limited by the remaining available space for this OMNI option.
- * 'msg-type' and 'transaction-id' are coded according to Section 8 of [RFC8415].
- * A set of DHCPv6 options coded according to Section 21 of [RFC8415] follows.

12.2.9. Host Identity Protocol (HIP) Message

The Host Identity Protocol (HIP) Message sub-option (when present) provides authentication for IPv6 ND messages exchanged between Clients and FHS Proxy/Servers over an open Internetwork. FHS Proxy/Servers authenticate the HIP authentication signatures in source Client IPv6 ND messages before securely forwarding them to other OMNI nodes. LHS Proxy/Servers that receive secured IPv6 ND messages from other OMNI nodes that do not already include a security sub-option insert HIP authentication signatures before forwarding them to the target Client.

OMNI interfaces MUST include the HIP message (when present) as the first sub-option of the first OMNI option, which MUST appear immediately following the IPv6 ND message header. OMNI interfaces can therefore easily locate the HIP message and verify the authentication signature without applying deep inspection. OMNI interfaces that receive IPv6 ND messages without a HIP (or other authentication) sub-option as the first OMNI sub-option instead verify the IPv6 ND message checksum.

OMNI interfaces include the HIP message sub-option when they forward IPv6 ND messages that require security over INET underlay interfaces, i.e., where authentication and integrity is not already assured by lower layers. The OMNI interface calculates the authentication signature over the entire length of the OAL packet (or super-packet) beginning with a pseudo-header of the IPv6 ND message header and extending over the remainder of the OAL packet. OMNI interfaces that process OAL packets that contain secured IPv6 ND messages verify the signature then either process the rest of the message locally or forward a proxied copy to the next hop.

When a FHS Client inserts a HIP message sub-option in an NS/NA message destined to a target in a remote spanning tree segment, it must ensure that the insertion does not cause the message to exceed the OMNI interface MTU. When the remote segment LHS Proxy/Server forwards the NS/NA message from the spanning tree to the target Client, it inserts a new HIP message sub-option if necessary while overwriting or cancelling the (now defunct) HIP message sub-option supplied by the FHS Client.

If the defunct HIP sub-option size was smaller than the space needed for the LHS Client HIP message (or, if no defunct HIP sub-option is present), the LHS Proxy/Server adjusts the space immediately following the OMNI header by copying the preceding portion of the IPv6 ND message into buffer headroom free space or copying the remainder of the IPv6 ND message into buffer tailroom free space. The LHS Proxy/Server then insets the new HIP sub-option immediately after the OMNI header and immediately before the next sub-option while properly overwriting the defunct sub-option if present.

If the defunct HIP sub-option size was larger than the space needed for the LHS Client HIP message, the LHS Proxy/Server instead overwrites the existing sub-option and writes a single Pad1 or PadN sub-option over the next 1-2 octets to cancel the remainder of the defunct sub-option. If the LHS Proxy/Server cannot create sufficient space through any means without causing the OMNI option to exceed 2040 octets or causing the IPv6 ND message to exceed the OMNI interface MTU, it returns a suitable error (see: Section 12.2.13) and drops the message.

The HIP message sub-option is formatted as shown below:

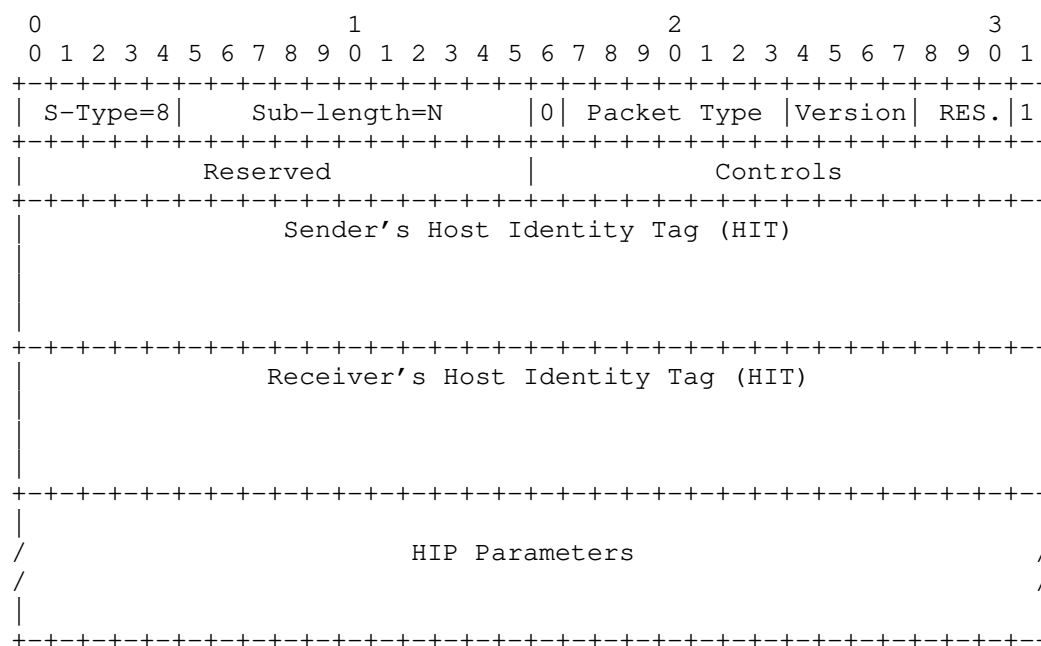


Figure 23: HIP Message Sub-option

- * Sub-Type is set to 8. If multiple instances appear in OMNI options of the same message the first is processed and all others are ignored.
- * Sub-Length is set to N, i.e., the length of the option in octets beginning immediately following the Sub-Length field and extending to the end of the HIP parameters. The length of the entire HIP message is therefore limited by the remaining available space for this OMNI option.
- * The HIP message is coded per Section 5 of [RFC7401], except that the OMNI "Sub-Type" and "Sub-Length" fields replace the first 2 octets of the HIP message header (i.e., the Next Header and Header Length fields). Also, since the IPv6 ND message is already protected by the authentication signature and/or lower-layer authentication and integrity checks, the HIP message Checksum field is replaced by a Reserved field set to 0 on transmission and ignored on reception.

Note: In some environments, maintenance of a Host Identity Tag (HIT) namespace may be unnecessary for securely associating an OMNI node with an IPv6 address-based identity. In that case, IPv6 ULAs can be used instead of HITs in the authentication signature as long as the address can be uniquely associated with the Sender/Receiver.

12.2.10. PIM-SM Message

The Protocol Independent Multicast - Sparse Mode (PIM-SM) Message sub-option may be included in the OMNI options of IPv6 ND messages. PIM-SM messages are formatted as specified in Section 4.9 of [RFC7761], with the exception that the Checksum field is replaced by a Reserved field (set to 0) since the IPv6 ND message is already protected by the IPv6 ND message checksum, authentication signature and/or lower-layer authentication and integrity checks. The PIM-SM message sub-option format is shown in Figure 24:

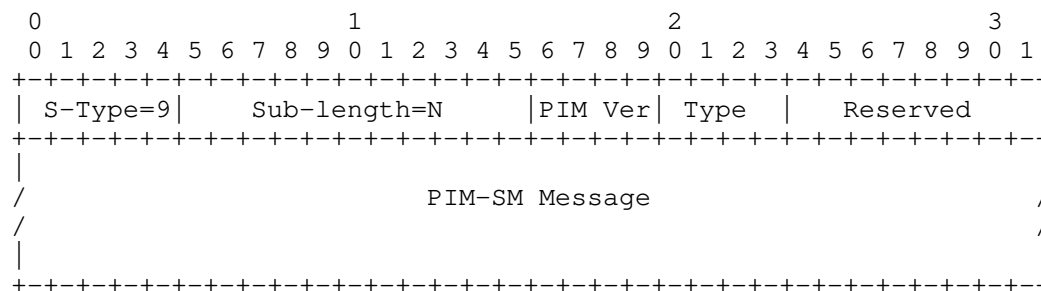


Figure 24: PIM-SM Message Option Format

- * Sub-Type is set to 9. If multiple instances appear in OMNI options of the same message all are processed.
- * Sub-Length is set to N, i.e., the length of the option in octets beginning immediately following the Sub-Length field and extending to the end of the PIM-SM message. The length of the entire PIM-SM message is therefore limited by the remaining available space for this OMNI option.
- * The PIM-SM message is coded exactly as specified in Section 4.9 of [RFC7761], except that the Checksum field is replaced by a Reserved field set to 0 on transmission and ignored on reception. The "PIM Ver" field MUST encode the value 2, and the "Type" field encodes the PIM message type. (See Section 4.9 of [RFC7761] for a list of PIM-SM message types and formats.)

12.2.11. Fragmentation Report (FRAGREP)

Fragmentation Report (FRAGREP) sub-options may be included in the OMNI options of uNA messages sent from an OAL destination to an OAL source. The message consists of $(N / 20)$ -many (Identification, Bitmap)-tuples which include the Identification values of OAL fragments received plus a Bitmap marking the ordinal positions of individual fragments received and fragments missing.

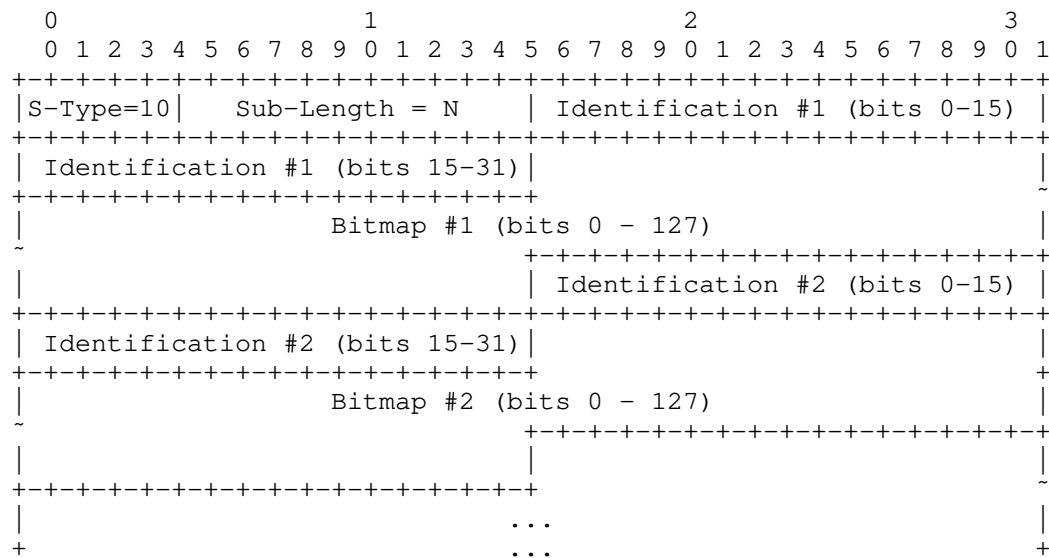


Figure 25: Fragmentation Report (FRAGREP)

- * Sub-Type is set to 10. If multiple instances appear in OMNI options of the same message all are processed.
- * Sub-Length is set to N, i.e., the length of the option in octets beginning immediately following the Sub-Length field and extending to the end of the sub-option. If N is not an integral multiple of 20 octets, the sub-option is ignored. The length of the entire sub-option should not cause the entire IPv6 ND message to exceed the minimum IPv6 MTU.
- * Identification (i) includes the IPv6 Identification value found in the Fragment Header of a received OAL fragment. (Only those Identification values included represent fragments for which loss was unambiguously observed; any Identification values not included correspond to fragments that were either received in their entirety or may still be in transit.)
- * Bitmap (i) includes an ordinal checklist of up to 128 fragments, with each bit set to 1 for a fragment received or 0 for a fragment missing. For example, for a 20-fragment OAL packet with ordinal fragments #3, #10, #13 and #17 missing and all other fragments received, Bitmap (i) encodes the following:

```

      0                               1                               2
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|1|1|1|0|1|1|1|1|1|1|0|1|1|0|1|1|1|0|1|1|0|0|0|...
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 26

(Note that loss of an OAL atomic fragment is indicated by a Bitmap(i) with all bits set to 0.)

12.2.12. Node Identification

```

      0                               1                               2                               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|S-Type=11|      Sub-length=N      |      ID-Type      |      ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
~                               Node Identification Value (N-1 octets)                               ~
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Figure 27: Node Identification

- * Sub-Type is set to 11. If multiple instances appear in OMNI options of the same IPv6 ND message the first instance of a specific ID-Type is processed and all other instances of the same ID-Type are ignored. (It is therefore possible for a single IPv6 ND message to convey multiple distinct Node Identifications - each with a different ID-Type.)
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow. The ID-Type field is always present; hence, the maximum Node Identification Value length is limited by the remaining available space in this OMNI option.
- * ID-Type is a 1 octet field that encodes the type of the Node Identification Value. The following ID-Type values are currently defined:
 - 0 - Universally Unique Identifier (UUID) [RFC4122]. Indicates that Node Identification Value contains a 16 octet UUID.
 - 1 - Host Identity Tag (HIT) [RFC7401]. Indicates that Node Identification Value contains a 16 octet HIT.
 - 2 - Hierarchical HIT (HHIT) [I-D.ietf-drip-rid]. Indicates that Node Identification Value contains a 16 octet HHIT.
 - 3 - Network Access Identifier (NAI) [RFC7542]. Indicates that Node Identification Value contains an N-1 octet NAI.
 - 4 - Fully-Qualified Domain Name (FQDN) [RFC1035]. Indicates that Node Identification Value contains an N-1 octet FQDN.
 - 5 - IPv6 Address. Indicates that Node Identification contains a 16-octet IPv6 address that is not a (H)HIT. The IPv6 address type is determined according to the IPv6 addressing architecture [RFC4291].
 - 6 - 252 - Unassigned.
 - 253-254 - Reserved for experimentation, as recommended in [RFC3692].
 - 255 - reserved by IANA.
- * Node Identification Value is an (N - 1) octet field encoded according to the appropriate the "ID-Type" reference above.

OMNI interfaces code Node Identification Values used for DHCPv6 messaging purposes as a DHCP Unique Identifier (DUID) using the "DUID-EN for OMNI" format with enterprise number 45282 (see: Section 25) as shown in Figure 28:

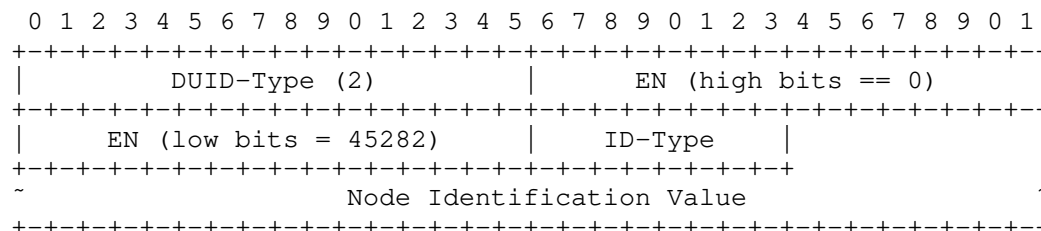


Figure 28: DUID-EN for OMNI Format

In this format, the OMNI interface codes the ID-Type and Node Identification Value fields from the OMNI sub-option following a 6 octet DUID-EN header, then includes the entire "DUID-EN for OMNI" in a DHCPv6 message per [RFC8415].

12.2.13. ICMPv6 Error

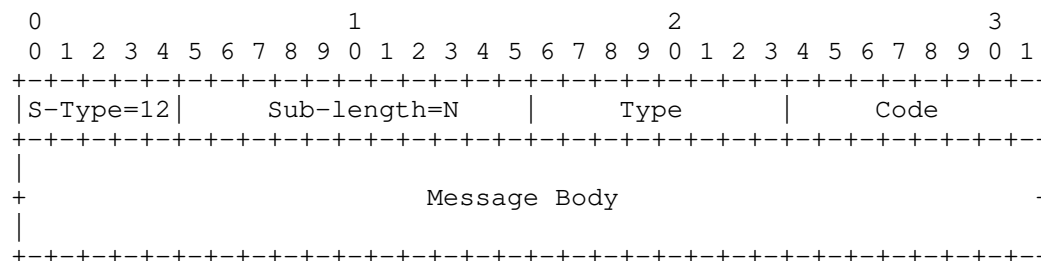


Figure 29: ICMPv6 Error

- * Sub-Type is set to 12. If multiple instances appear in OMNI options of the same IPv6 ND message all are processed.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow.
- * Sub-Option Data includes a one octet Type followed by a one octet Code followed by an (N-2)-octet Message Body encoded exactly as per Section 2.1 of [RFC4443]. OMNI interfaces include as much of the ICMPv6 error message body in the sub-option as possible without causing the entire IPv6 ND message to exceed the minimum IPv6 MTU. While all ICMPv6 error message types are supported, OAL destinations in particular may include ICMPv6 PTB messages in uNA

messages to provide MTU feedback information via the OAL source (see: Section 6.8). Note: ICMPv6 informational messages must not be included and must be ignored if received.

12.2.14. QUIC-TLS Message

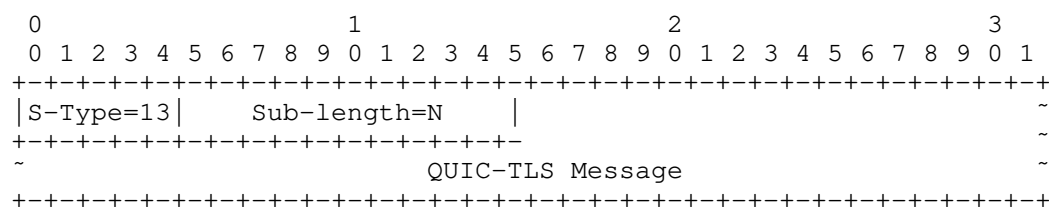


Figure 30: QUIC-TLS Message

- * Sub-Type is set to 13. If multiple instances appear in OMNI options of the same IPv6 ND message, the first is processed and all others are ignored.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow.
- * The QUIC-TLS message [RFC9000][RFC9001][RFC9002] encodes the QUIC and TLS message parameters necessary to support QUIC connection establishment.

When present, the QUIC-TLS Message sub-option MUST appear immediately after the header of the first OMNI option in the IPv6 ND message; if the sub-option appears in any other location it MUST be ignored. IPv6 ND solicitation and advertisement messages serve as couriers to transport the QUIC and TLS parameters necessary to establish a secured QUIC connection.

12.2.15. Proxy/Server Departure

OMNI Clients include a Proxy/Server Departure sub-option in RS messages when they associate with a new FHS and/or Hub Proxy/Server and need to send a departure indication to an old FHS and/or Hub Proxy/Server. The Proxy/Server Departure sub-option is formatted as shown below:

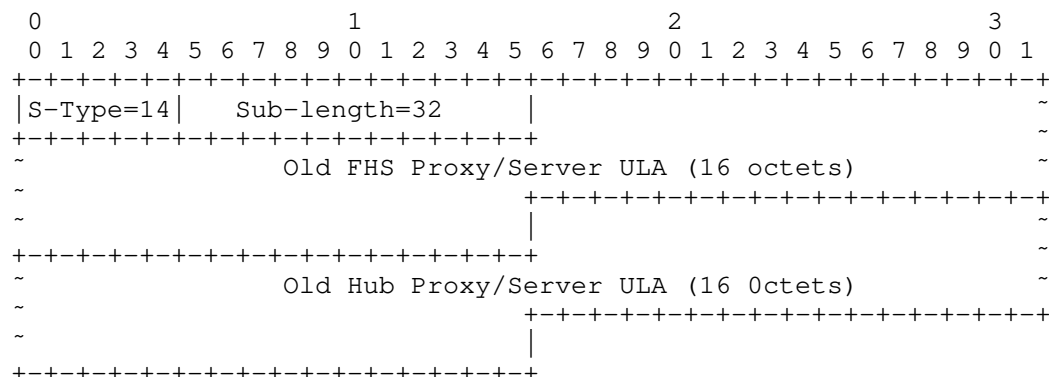


Figure 31: Proxy/Server Departure

- * Sub-Type is set to 14.
- * Sub-Length is set to 32.
- * Sub-Option Data contains the 16 octet ULA for the "Old FHS Proxy/Server" followed by a 16 octet ULA for an "Old Hub Proxy/Server". (If the Old FHS/Hub is unspecified, the corresponding ULA instead includes the value 0.)

12.2.16. Sub-Type Extension

Since the Sub-Type field is only 5 bits in length, future specifications of major protocol functions may exhaust the remaining Sub-Type values available for assignment. This document therefore defines Sub-Type 30 as an "extension", meaning that the actual Sub-Type type is determined by examining a 1 octet "Extension-Type" field immediately following the Sub-Length field. The Sub-Type Extension is formatted as shown in Figure 32:

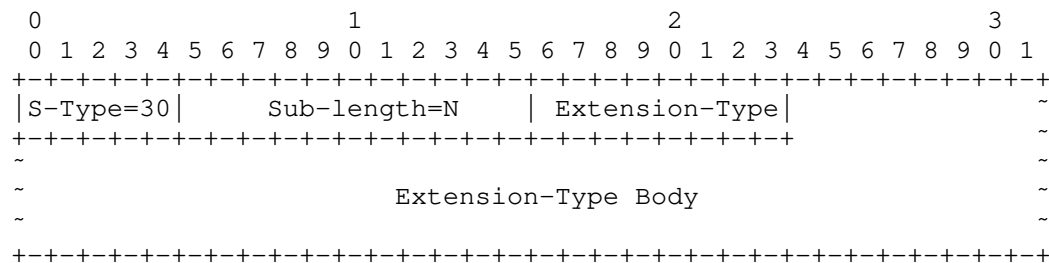


Figure 32: Sub-Type Extension

- * Sub-Type is set to 30. If multiple instances appear in OMNI options of the same message all are processed, where each individual extension defines its own policy for processing multiple of that type.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow. The Extension-Type field is always present, and the maximum Extension-Type Body length is limited by the remaining available space in this OMNI option.
- * Extension-Type contains a 1 octet Sub-Type Extension value between 0 and 255.
- * Extension-Type Body contains an N-1 octet block with format defined by the given extension specification.

Extension-Type values 0 and 1 are defined in the following subsections, while Extension-Type values 2 through 252 are available for assignment by future specifications which must also define the format of the Extension-Type Body and its processing rules. Extension-Type values 253 and 254 are reserved for experimentation, as recommended in [RFC3692], and value 255 is reserved by IANA.

12.2.16.1. RFC4380 Header Extension Option

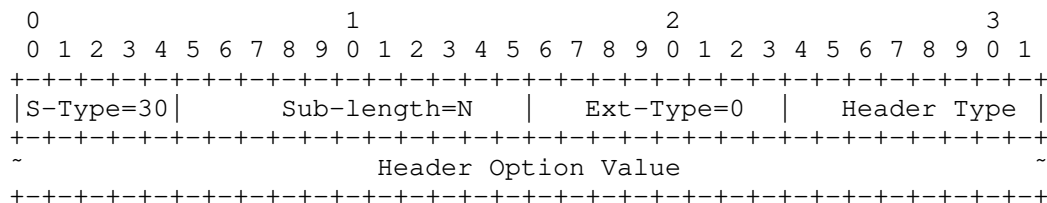


Figure 33: RFC4380 Header Extension Option (Extension-Type 0)

- * Sub-Type is set to 30.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow. The Extension-Type and Header Type fields are always present, and the Header Option Value is limited by the remaining available space in this OMNI option.
- * Extension-Type is set to 0. Each instance encodes exactly one header option per Section 5.1.1 of [RFC4380], with Ext-Type and Header Type representing the first two octets of the option. If multiple instances of the same Header Type appear in OMNI options of the same message the first instance is processed and all others are ignored. If Header Type indicates an Authentication

Encapsulation (see below), the entire sub-option MUST appear as the first sub-option of the first OMNI option, which MUST appear immediately following the IPv6 ND message header.

- * Header Type and Header Option Value are coded exactly as specified in Section 5.1.1 of [RFC4380]; the following types are currently defined:
 - 0 - Origin Indication (IPv4) - value coded as a UDP port number followed by a 4-octet IPv4 address both in "obfuscated" form per Section 5.1.1 of [RFC4380].
 - 1 - Authentication Encapsulation - value coded per Section 5.1.1 of [RFC4380].
 - 2 - Origin Indication (IPv6) - value coded per Section 5.1.1 of [RFC4380], except that the address is a 16-octet IPv6 address instead of a 4-octet IPv4 address.
- * Header Type values 3 through 252 are available for assignment by future specifications, which must also define the format of the Header Option Value and its processing rules. Header Type values 253 and 254 are reserved for experimentation, as recommended in [RFC3692], and value 255 is Reserved by IANA.

12.2.16.2. RFC6081 Trailer Extension Option

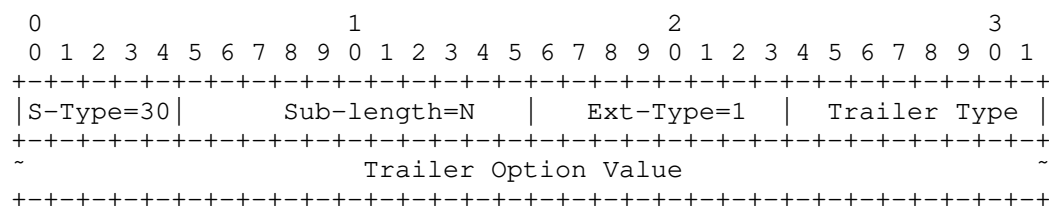


Figure 34: RFC6081 Trailer Extension Option (Extension-Type 1)

- * Sub-Type is set to 30.
- * Sub-Length is set to N that encodes the number of Sub-Option Data octets that follow. The Extension-Type and Trailer Type fields are always present, and the maximum-length Trailer Option Value is limited by the remaining available space in this OMNI option.
- * Extension-Type is set to 1. Each instance encodes exactly one trailer option per Section 4 of [RFC6081]. If multiple instances of the same Trailer Type appear in OMNI options of the same message the first instance is processed and all others ignored.

- * Trailer Type and Trailer Option Value are coded exactly as specified in Section 4 of [RFC6081]; the following Trailer Types are currently defined:
 - 0 - Unassigned
 - 1 - Nonce Trailer - value coded per Section 4.2 of [RFC6081].
 - 2 - Unassigned
 - 3 - Alternate Address Trailer (IPv4) - value coded per Section 4.3 of [RFC6081].
 - 4 - Neighbor Discovery Option Trailer - value coded per Section 4.4 of [RFC6081].
 - 5 - Random Port Trailer - value coded per Section 4.5 of [RFC6081].
 - 6 - Alternate Address Trailer (IPv6) - value coded per Section 4.3 of [RFC6081], except that each address is a 16-octet IPv6 address instead of a 4-octet IPv4 address.
- * Trailer Type values 7 through 252 are available for assignment by future specifications, which must also define the format of the Trailer Option Value and its processing rules. Trailer Type values 253 and 254 are reserved for experimentation, as recommended in [RFC3692], and value 255 is Reserved by IANA.

13. Address Mapping - Multicast

The multicast address mapping of the native underlay interface applies. The Client mobile router also serves as an IGMP/MLD Proxy for its ENETs and/or hosted applications per [RFC4605].

The Client uses Multicast Listener Discovery (MLDv2) [RFC3810] to coordinate with Proxy/Servers, and underlay network elements use MLD snooping [RFC4541]. The Client can also employ multicast routing protocols to coordinate with network-based multicast sources as specified in [I-D.templin-6man-aero].

Since the OMNI link model is NBMA, OMNI links support link-scoped multicast through iterative unicast transmissions to individual multicast group members (i.e., unicast/multicast emulation).

14. Multilink Conceptual Sending Algorithm

The Client's IPv6 layer selects the outbound OMNI interface according to SBM considerations when forwarding original IP packets from local or ENET applications to external correspondents. Each OMNI interface maintains a neighbor cache the same as for any IPv6 interface, but includes additional state for multilink coordination. Each Client OMNI interface maintains default routes via Proxy/Servers discovered as discussed in Section 15, and may configure more-specific routes discovered through means outside the scope of this specification.

For each original IP packet it forwards, the OMNI interface selects one or more source underlay interfaces based on PBM factors (e.g., traffic attributes, cost, performance, message size, etc.) and one or more target underlay interfaces for the neighbor based on Interface Attributes received in IPv6 ND messages (see: Section 12.2.4). Multilink forwarding may also direct packet replication across multiple underlay interface pairs for increased reliability at the expense of duplication. The set of all Interface Attributes and Traffic Selectors received in IPv6 ND messages determines the multilink forwarding profile for selecting target underlay interfaces.

When the OMNI interface sends an original IP packet over a selected source underlay interface, it first employs OAL encapsulation and fragmentation as discussed in Section 5, then performs L2 encapsulation as directed by the appropriate MFV. The OMNI interface also performs L2 encapsulation (following OAL encapsulation) when the nearest Proxy/Server is located multiple hops away as discussed in Section 15.2.

OMNI interface multilink service designers MUST observe the BCP guidance in Section 15 [RFC3819] in terms of implications for reordering when original IP packets from the same flow may be spread across multiple underlay interfaces having diverse properties.

14.1. Multiple OMNI Interfaces

Clients may connect to multiple independent OMNI links within the same or different OMNI domains to support SBM. The Client configures a separate OMNI interface for each link so that multiple interfaces (e.g., omni0, omni1, omni2, etc.) are exposed to the IP layer. Each OMNI interface configures one or more OMNI anycast addresses (see: Section 10), and the Client injects the corresponding anycast prefixes into the ENET routing system. Multiple distinct OMNI links can therefore be used to support fault tolerance, load balancing, reliability, etc.

Applications in ENETs can use Segment Routing to select the desired OMNI interface based on SBM considerations. The application writes an OMNI anycast address into the original IP packet's destination address, and writes the actual destination (along with any additional intermediate hops) into the Segment Routing Header. Standard IP routing directs the packet to the Client's mobile router entity, where the anycast address identifies the correct OMNI interface for next hop forwarding. When the Client receives the packet, it replaces the IP destination address with the next hop found in the Segment Routing Header and forwards the message via the OMNI interface identified by the anycast address.

Note: The Client need not configure its OMNI interface indexes in one-to-one correspondence with the global OMNI Link-IDs configured for OMNI domain administration since the Client's indexes (i.e., `omni0`, `omni1`, `omni2`, etc.) are used only for its own local interface management.

14.2. Client-Proxy/Server Loop Prevention

After a Proxy/Server has registered an MNP for a Client (see: Section 15), the Proxy/Server will forward all packets destined to an address within the MNP to the Client. The Client will under normal circumstances then forward the packet to the correct destination within its connected (downstream) ENETs.

If at some later time the Client loses state (e.g., after a reboot), it may begin returning packets with destinations corresponding to its MNP to the Proxy/Server as its default router. The Proxy/Server therefore drops any original IP packets received from the Client with a destination address that corresponds to the Client's MNP (i.e., whether ULA or GUA), and drops any carrier packets with both source and destination address corresponding to the same Client's MNP regardless of their origin.

15. Router Discovery and Prefix Registration

Clients engage the MS by sending RS messages with OMNI options under the assumption that one or more Proxy/Server will process the message and respond. The RS message is received by a FHS Proxy/Server, which may in turn forward a proxied copy of the RS to a Hub Proxy/Server located on the same or different SRT segment. The Hub Proxy/Server then returns an RA message either directly to the Client or via an FHS Proxy/Server acting as a proxy.

Clients and FHS Proxy/Servers include an authentication signature in their RS/RA exchanges when necessary; otherwise, they calculate and include a valid IPv6 ND message checksum (see: Section 12 and

Appendix B). FHS and Hub Proxy/Server RS/RA message exchanges over the SRT secured spanning tree instead always include the checksum and omit the authentication signature. Clients and Proxy/Servers use the information included in RS/RA messages to establish NCE state and OMNI link autoconfiguration information as discussed in this section.

For each underlay interface, the Client sends RS messages with OMNI options to coordinate with a (potentially) different FHS Proxy/Server for each interface but with a single Hub Proxy/Server. All Proxy/Servers are identified by their ULA-RNDs and accept carrier packets addressed to their anycast/unicast L2 INADDRs; the Hub Proxy/Server may be chosen among any of the Client's FHS Proxy/Servers or may be any other Proxy/Server for the OMNI link. Example ULA/INADDR discovery methods are given in [RFC5214] and include data link login parameters, name service lookups, static configuration, a static "hosts" file, etc. In the absence of other information, the Client can resolve the DNS Fully-Qualified Domain Name (FQDN) "linkupnetworks.[domainname]" where "linkupnetworks" is a constant text string and "[domainname]" is a DNS suffix for the OMNI link (e.g., "example.com"). The name resolution will retain a set of DNS resource records with the addresses of Proxy/Servers for the domain.

Each FHS Proxy/Server configures an ULA-RND based on a /64 ULA prefix for the link/segment with randomly-generated Global ID to assure global uniqueness then administratively assigned to FHS Proxy/Servers for the link to assure global consistency. The Client can then configure ULA-MNPs derived from the 64-bit ULA prefix assigned to a FHS Proxy/Server for each underlay interface. The FHS Proxy/Servers discovered over multiple of the Client's underlay interfaces may configure the same or different ULA prefixes, and the Client's ULA-MNP for each underlay interface will fall within the ULA (multilink) subnet relative to each FHS Proxy/Server.

Clients configure OMNI interfaces that observe the properties discussed in previous sections. The OMNI interface and its underlay interfaces are said to be in either the "UP" or "DOWN" state according to administrative actions in conjunction with the interface connectivity status. An OMNI interface transitions to UP or DOWN through administrative action and/or through state transitions of the underlay interfaces. When a first underlay interface transitions to UP, the OMNI interface also transitions to UP. When all underlay interfaces transition to DOWN, the OMNI interface also transitions to DOWN.

When a Client OMNI interface transitions to UP, it sends RS messages to register its MNP and an initial set of underlay interfaces that are also UP. The Client sends additional RS messages to refresh lifetimes and to register/deregister underlay interfaces as they

transition to UP or DOWN. The Client's OMNI interface sends initial RS messages over an UP underlay interface with its {TLA,XLA}-MNP as the source (or with a {TLA,XLA}-RND as the source if it does not yet have an MNP) and with destination set to link-scoped All-Routers multicast or the ULA of a specific (Hub) Proxy/Server. The OMNI interface includes an OMNI option per Section 12 with an OMNI Neighbor Coordination sub-option with (Preflen assertion, N/A/U flags and Window Synchronization parameters), an Interface Attributes sub-option for the underlay interface, a DHCPv6 Solicit sub-option if necessary, and with any other necessary OMNI sub-options such as authentication, Proxy/Server Departure, etc.

The Client then calculates the authentication signature or checksum and prepares to forward the RS over the underlay interface using OAL encapsulation and fragmentation if necessary. If the Client uses OAL encapsulation for RS messages sent to an unsynchronized FHS Proxy/Server over an INET interface, the entire RS message must fit within a single carrier packet (i.e., an atomic fragment) so that the FHS Proxy/Server can verify the authentication signature without having to reassemble. The OMNI interface selects an Identification value (see: Section 6.6), sets the OAL source address to the ULA-MNP corresponding to the RS source if known (otherwise to a {TLA,XLA}-RND), sets the OAL destination to an OMNI IPv6 anycast address or a known Proxy/Server ULA, optionally includes a Nonce and/or Timestamp, then performs fragmentation if necessary. When L2 encapsulation is used, the Client includes the discovered FHS Proxy/Server INADDR or an anycast address as the L2 destination then forwards the resulting carrier packet(s) into the underlay network. Note that the Client does not yet create a NCE, but instead remembers the Identification, Nonce and/or Timestamp values included in its RS message transmissions to match against any received RA messages.

When an FHS Proxy/Server receives the carrier packets containing an RS it sets aside the L2 headers, verifies the Identifications and reassembles if necessary, sets aside the OAL header, then verifies the RS authentication signature or checksum. The FHS Proxy/Server then creates/updates a NCE indexed by the Client's RS source address and caches the OMNI Interface Attributes and any Traffic Selector sub-options while also caching the L2 (UDP/IP) and OAL source and destination address information. The FHS Proxy/Server next caches the OMNI Neighbor Coordination sub-option Window Synchronization parameters and N flag to determine its role in processing NS(NUD) messages (see: Section 12.1) then examines the RS destination address. If the destination matches its own ULA, the FHS Proxy/Server assumes the Hub role and acts as the sole entry point for injecting the Client's XLA-MNP into the OMNI link routing system (i.e., after performing any necessary prefix delegation operations) while including a prefix length and setting the prefix to fd00::/64

and suffix to the 64-bit MNP. The FHS/Hub Proxy/Server then caches the OMNI Neighbor Coordination sub-option A/U flags to determine its role in processing NS(AR) messages and generating uNA messages (see: Section 12.1).

The FHS/Hub Proxy/Server then prepares to return an RA message directly to the Client by first populating the Cur Hop Limit, Flags, Router Lifetime, Reachable Time and Retrans Timer fields with values appropriate for the OMNI link. The FHS/Hub Proxy/Server next includes as the first RA message option an OMNI option with a neighbor coordination sub-option with Window Synchronization information, an authentication sub-option if necessary and a (proxied) copy of the Client's original Interface Attributes sub-option with its INET-facing interface information written in the FMT/SRT and LHS Proxy/Server ULA/INADDR fields. If the RS L2 destination IP address was anycast, the FHS/Hub Proxy/Server next includes a second Interface Attributes sub-option with omIndex set to '0' and with a unicast L2 IP address for its Client-facing interface in the INADDR field.

The FHS/Hub Proxy/Server next includes an Origin Indication sub-option that includes the RS L2 source INADDR information (see: Section 12.2.16.1), then includes any other necessary OMNI sub-options (either within the same OMNI option or in additional OMNI options). Following the OMNI option(s), the FHS/Hub Proxy/Server next includes any other necessary RA options such as PIOs with (A; L=0) that include the OMNI link MSPs [RFC8028], RIOs [RFC4191] with more-specific routes, Nonce and Timestamp options, etc. The FHS/Hub Proxy/Server then sets the RA source address to its own ULA and destination address to the Client's ULA-MNP (i.e., relative to the ULA /64 prefix for its Client-facing underlay interface) while also recording the corresponding XLA-MNP as an (alternate) index to the Client NCE, then calculates the authentication signature or checksum. The FHS/Hub Proxy/Server finally performs OAL encapsulation with source set to its own ULA and destination set to the OAL source that appeared in the RS, then fragments if necessary, encapsulates each fragment in appropriate L2 headers with source and destination address information reversed from the RS L2 information and returns the resulting carrier packets to the Client over the same underlay interface the RS arrived on.

When an FHS Proxy/Server receives an RS with a valid authentication signature or checksum and with destination set to link-scoped All-Routers multicast, it can either assume the Hub role itself the same as above or act as a proxy and select the ULA of another Proxy/Server to serve as the Hub. When an FHS Proxy/Server assumes the proxy role or receives an RS with destination set to the ULA of another Proxy/Server, it forwards the message while acting as a proxy. The FHS

Proxy/Server creates/updates a NCE for the Client (i.e., based on the RS source address) and caches the OAL source, Window Synchronization, N flag, Interface Attributes addressing information as above then writes its own INET-facing FMT/SRT and LHS Proxy/Server ULA/INADDR information into the appropriate Interface Attributes sub-option fields. The FHS Proxy/Server then calculates and includes the checksum, performs OAL encapsulation with source set to its own ULA and destination set to the ULA of the Hub Proxy/Server, fragments if necessary, encapsulates each fragment in appropriate L2 headers and sends the resulting carrier packets into the SRT secured spanning tree.

When the Hub Proxy/Server receives the carrier packets, it discards the L2 headers, reassembles if necessary to obtain the proxied RS, then performs DHCPv6 Prefix Delegation (PD) to obtain the Client's MNP if the RS source is a {TLA,XLA}-RND. The Hub Proxy/Server then creates/updates a NCE for the Client's XLA-MNP and caches any state (including the A/U flags, OAL addresses, Interface Attributes information and Traffic Selectors), then finally performs routing protocol injection. The Hub Proxy/Server then returns an RA that echoes the Client's (proxied) Interface Attributes sub-option and with any RA parameters the same as specified for the FHS/Hub Proxy/Server case above. The Hub Proxy/Server then sets the RA source address to its own ULA and destination address to the RS source address; if the RS source address is a {TLA,XLA}-RND, the Hub Proxy/Server also includes the MNP in a DHCPv6 PD Reply OMNI sub-option. The Hub Proxy/Server next calculates the checksum, then encapsulates the RA as an OAL packet with source set to its own ULA and destination set to the ULA of the FHS Proxy/Server that forwarded the RS. The Hub Proxy/Server finally fragments if necessary, encapsulates each fragment in appropriate L2 headers and sends the resulting carrier packets into the secured spanning tree.

When the FHS Proxy/Server receives the carrier packets it discards the L2 headers, reassembles if necessary to obtain the RA message, verifies the checksum then updates the OMNI interface NCE for the Client and creates/updates a NCE for the Hub. The FHS Proxy/Server then sets the P flag in the RA flags field [RFC4389] and proxys the RA by changing the OAL source to its own ULA, changing the OAL destination to the OAL address found in the Client's NCE, and changing the RA destination address to the ULA-MNP of the Client relative to its own /64 ULA prefix while also recording the corresponding XLA-MNP as an alternate index into the Client NCE. (If the RA destination address was a {TLA,XLA}-RND, the FHS Proxy Server determines the MNP by consulting the DHCPv6 PD Reply message sub-option.) The FHS Proxy/Server next includes Window Synchronization parameters responsive to those in the Client's RS, an Interface Attributes sub-option with omIndex '0' and with its unicast L2 IP

address if necessary (see above), an Origin Indication sub-option with the Client's cached INADDR and an authentication sub-option if necessary. The FHS Proxy/Server finally selects an Identification value per Section 6.6, calculates the authentication signature or checksum, fragments if necessary, encapsulates each fragment in L2 headers with addresses taken from the Client's NCE and returns the resulting carrier packets via the same underlay interface over which the RS was received.

When the Client receives the carrier packets, it discards the L2 headers, reassembles if necessary and removes the OAL header to obtain the RA message. The Client next verifies the authentication signature or checksum, then matches the RA message with its previously-sent RS by comparing the RS Sequence Number with the RA Acknowledgement Number and also comparing the Nonce and/or Timestamp values if present. If the values match, the Client then creates/updates OMNI interface NCEs for both the Hub and FHS Proxy/Server and caches the information in the RA message. In particular, the Client caches the RA source address as the Hub Proxy/Server ULA and uses the OAL source address to configure both an underlay interface-specific ULA for the Hub Proxy/Server and the ULA of this FHS Proxy/Server. The Client then uses the ULA-MNP in the RA destination address to configure its address within the ULA (multilink) subnet prefix of the FHS Proxy/Server. If the Client has multiple underlay interfaces, it creates additional FHS Proxy/Server NCEs and ULA-MNPs as necessary when it receives RAs over those interfaces (noting that multiple of the Client's underlay interfaces may be serviced by the same or different FHS Proxy/Servers). The Client finally adds the Hub Proxy/Server ULA to the default router list if necessary.

For each underlay interface, the Client next caches the (filled-out) Interface Attributes for its own omIndex and Origin Indication information that it received in an RA message over that interface so that it can include them in future NS/NA messages to provide neighbors with accurate FMT/SRT/LHS information. (If the message includes an Interface Attributes sub-option with omIndex '0', the Client also caches the INADDR as the underlay network-local unicast address of the FHS Proxy//Server via that underlay interface.) The Client then compares the Origin Indication INADDR information with its own underlay interface addresses to determine whether there may be NATs on the path to the FHS Proxy/Server; if the INADDR information differs, the Client is behind a NAT and must supply the Origin information in IPv6 ND message exchanges with prospective neighbors on the same SRT segment. The Client finally configures default routes and assigns the OMNI Subnet Router Anycast address corresponding to the MNP (e.g., 2001:db8:1:2::) to the OMNI interface.

Following the initial exchange, the FHS Proxy/Server MAY later send additional periodic and/or event-driven unsolicited RA messages per [RFC4861]. (The unsolicited RAs may be initiated either by the FHS Proxy/Server itself or by the Hub via the FHS as a proxy.) The Client then continuously manages its underlay interfaces according to their states as follows:

- * When an underlay interface transitions to UP, the Client sends an RS over the underlay interface with an OMNI option with sub-options as specified above.
- * When an underlay interface transitions to DOWN, the Client sends unsolicited NA messages over any UP underlay interface with an OMNI option containing Interface Attributes sub-options for the DOWN underlay interface with Link set to '0'. The Client sends isolated unsolicited NAs when reliability is not thought to be a concern (e.g., if redundant transmissions are sent on multiple underlay interfaces), or may instead set the PNG flag in the OMNI header to trigger a uNA reply.
- * When the Router Lifetime for the Hub Proxy/Server nears expiration, the Client sends an RS over any underlay interface to receive a fresh RA from the Hub. If no RA messages are received over a first underlay interface (i.e., after retrying), the Client marks the underlay interface as DOWN and should attempt to contact the Hub Proxy/Server via a different underlay interface. If the Hub Proxy/Server is unresponsive over additional underlay interfaces, the Client sends an RS message with destination set to the ULA of another Proxy/Server which will then assume the Hub role.
- * When all of a Client's underlay interfaces have transitioned to DOWN (or if the prefix registration lifetime expires), the Hub Proxy/Server withdraws the MNP the same as if it had received a message with a release indication.

The Client is responsible for retrying each RS exchange up to MAX_RTR_SOLICITATIONS times separated by RTR_SOLICITATION_INTERVAL seconds until an RA is received. If no RA is received over an UP underlay interface (i.e., even after attempting to contact alternate Proxy/Servers), the Client declares this underlay interface as DOWN. When changing to a new FHS or Hub Proxy/Server, the Client also includes a Proxy/Server Departure OMNI sub-option in new RS messages; the (new) FHS Proxy/Server will in turn send uNA messages to the old FHS and/or Hub Proxy/Server to announce the Client's departure as discussed in [I-D.templin-6man-aero].

The IPv6 layer sees the OMNI interface as an ordinary IPv6 interface. Therefore, when the IPv6 layer sends an RS message the OMNI interface returns an internally-generated RA message as though the message originated from an IPv6 router. The internally-generated RA message contains configuration information consistent with the information received from the RAs generated by the Hub Proxy/Server. Whether the OMNI interface IPv6 ND messaging process is initiated from the receipt of an RS message from the IPv6 layer or independently of the IPv6 layer is an implementation matter. Some implementations may elect to defer the OMNI interface internal RS/RA messaging process until an RS is received from the IPv6 layer, while others may elect to initiate the process proactively. Still other deployments may elect to administratively disable IPv6 layer RS/RA messaging over the OMNI interface, since the messages are not required to drive the OMNI interface internal RS/RA process. (Note that this same logic applies to IPv4 implementations that employ "ICMP Router Discovery" [RFC1256].)

Note: The Router Lifetime value in RA messages indicates the time before which the Client must send another RS message over this underlay interface (e.g., 600 seconds), however that timescale may be significantly longer than the lifetime the MS has committed to retain the prefix registration (e.g., REACHABLETIME seconds). Proxy/Servers are therefore responsible for keeping MS state alive on a shorter timescale than the Client may be required to do on its own behalf.

Note: On certain multicast-capable underlay interfaces, Clients should send periodic unsolicited multicast NA messages and Proxy/Servers should send periodic unsolicited multicast RA messages as "beacons" that can be heard by other nodes on the link. If a node fails to receive a beacon after a timeout value specific to the link, it can initiate Neighbor Unreachability Detection (NUD) exchanges to test reachability.

Note: If a single FHS Proxy/Server services multiple of a Client's underlay interfaces, Window Synchronization will initially be repeated for the RS/RA exchange over each underlay interface, i.e., until the Client discovers the many-to-one relationship. This will naturally result in a single window synchronization that applies over the Client's multiple underlay interfaces for the same FHS Proxy/Server.

Note: Although the Client's FHS Proxy/Server is a first-hop segment node from its own perspective, the Client stores the Proxy/Server's FMT/SRT/ULA/INADDR as last-hop segment (LHS) information to supply to neighbors. This allows both the Client and Hub Proxy/Server to supply the information to neighbors that will perceive it as LHS information on the return path to the Client.

Note: The Hub Proxy/Server injects Client XLA-MNP into the OMNI link routing system by simply creating a route-to-interface forwarding table entry for fd00::{MNP}/N via the OMNI interface. The dynamic routing protocol will notice the new entry and propagate the route to its peers. If the Hub receives additional RS messages, it need not re-create the forwarding table entry (nor disturb the dynamic routing protocol) if an entry is already present. If the Hub ceases to receive RS messages from any of the Client's interfaces, it removes the Client XLA-MNP from the forwarding table (i.e., after a short delay) resulting in its removal also from the routing system.

Note: If the Client's initial RS message includes an anycast L2 destination address, the FHS Proxy/Server returns the solicited RA using the same anycast address as the L2 source while including an Interface Attributes sub-option with omIndex '0' and its true unicast address in the INADDR. When the Client sends additional RS messages, it includes this FHS Proxy/Server unicast address as the L2 destination and the FHS Proxy/Server returns the solicited RA using the same unicast address as the L2 source. This will ensure that RS/RA exchanges are not impeded by any NATs on the path while avoiding long-term exposure of messages that use an anycast address as the source.

Note: The Origin Indication sub-option is included only by the FHS Proxy/Server and not by the Hub (unless the Hub is also serving as an FHS).

Note: Clients should set the N/A/U flags consistently in successive RS messages and only change those settings when an FHS/Hub Proxy/Server service profile update is necessary.

Note: After a Client has discovered its ULA-MNPs for a given set of FHS Proxy/Servers, it should begin using its XLA-MNP as the IPv6 ND message source address and ULA-MNP as the OAL source address in future IPv6 ND messages and refrain from further use of TLAs. In any case, the Client SHOULD NOT gratuitously configure and use large numbers of additional TLAs, as doing so would simply result in address change churn in neighbor cache entries with no operational advantages.

Note: Although the Client adds the Hub Proxy/Server ULA to the default router list, it also caches the ULAs of the FHS Proxy/Servers on the path to the Hub over each underlying interface. When the Client needs to send a packet to a default router, it therefore selects an ULA corresponding to the selected interface which directs the packet to an FHS Proxy/Server for that interface. The FHS Proxy/Server then forwards the packet without disturbing the Hub.

15.1. Window Synchronization

In environments where Identification window synchronization is necessary, the RS/RA exchanges discussed above observe the principles specified in Section 6.6. Window synchronization is conducted between the Client and each FHS Proxy/Server used to contact the Hub Proxy/Server, i.e., and not between the Client and the Hub. This is due to the fact that the Hub Proxy/Server is responsible only for forwarding control and data messages via the secured spanning tree to FHS Proxy/Servers, and is not responsible for forwarding messages directly to the Client under a synchronized window. Also, in the reverse direction the FHS Proxy/Servers handle all default forwarding actions without forwarding Client-initiated data to the Hub.

When a Client needs to perform window synchronization via a new FHS Proxy/Server, it sets the RS source address to its own {TLA,XLA}-MNP (or a {TLA,XLA}-RND) and destination address to the ULA of the Hub Proxy/Server (or to All-Routers multicast in an initial RS), then sets the SYN flag and includes an initial Sequence Number for Window Synchronization. The Client then performs OAL encapsulation using its own ULA-MNP (or the TLA-RND) as the source and the ULA of the FHS Proxy/Server as the destination and includes an Interface Attributes sub-option then forwards the resulting carrier packets to the FHS Proxy/Server. The FHS Proxy/Server then extracts the RS message and caches the Window Synchronization parameters then re-encapsulates with its own ULA as the source and the ULA of the Hub Proxy/Server as the target.

The FHS Proxy/Server then forwards the resulting carrier packets via the secured spanning tree to the Hub Proxy/Server, which updates the Client's Interface Attributes and returns a unicast RA message with source set to its own ULA and destination set to the RS source address and with the Client's Interface Attributes echoed. The Hub Proxy/Server then performs OAL encapsulation using its own ULA as the source and the ULA of the FHS Proxy/Server as the destination, then forwards the carrier packets via the secured spanning tree to the FHS Proxy/Server. The FHS Proxy/Server then proxys the message as discussed in the previous section and includes responsive Window Synchronization information. The FHS Proxy/Server then forwards the message to the Client which updates its window synchronization information for the FHS Proxy/Server as necessary.

Following the initial RS/RA-driven window synchronization, the Client can re-assert new windows with specific FHS Proxy/Servers by performing NS/NA exchanges between its own XLA-MNPs and the ULAs of the FHS Proxy/Servers without having to disturb the Hub.

15.2. Router Discovery in IP Multihop and IPv4-Only Networks

On some *NETs, a Client may be located multiple IP hops away from the nearest OMNI link Proxy/Server. Forwarding through IP multihop *NETs is conducted through the application of a routing protocol (e.g., a MANET/VANET routing protocol over omni-directional wireless interfaces, an inter-domain routing protocol in an enterprise network, etc.). Example routing protocols optimized for MANET/VANET operations include [RFC3684] and [RFC5614] which operate according to the link model articulated in [RFC5889] and subnet model articulated in [RFC5942].

A Client located potentially multiple *NET hops away from the nearest Proxy/Server prepares an RS message, sets the source address to its {TLA,XLA}-MNP (or to a {TLA,XLA}-RND if it does not yet have an MNP), and sets the destination to link-scoped All-Routers multicast or the unicast ULA of a Proxy/Server the same as discussed above. The OMNI interface then employs OAL encapsulation, sets the OAL source address to a TLA and sets the OAL destination to an OMNI IPv6 anycast address based on either a native IPv6 or IPv4-Compatible IPv6 prefix (see: Section 10).

For IPv6-enabled *NETs, if the underlay interface does not configure an IPv6 GUA the Client injects the TLA into the IPv6 multihop routing system and forwards the message without further encapsulation. Otherwise, the Client encapsulates the message in UDP/IPv6 L2 headers, sets the source to the underlay interface IPv6 address and sets the destination to the same OMNI IPv6 anycast address. The Client then forwards the message into the IPv6 multihop routing system which conveys it to the nearest Proxy/Server that advertises a matching OMNI IPv6 anycast prefix. If the nearest Proxy/Server is too busy, it should forward (without Proxying) the OAL-encapsulated RS to another nearby Proxy/Server connected to the same IPv6 (multihop) network that also advertises the matching OMNI IPv6 anycast prefix.

For IPv4-only *NETs, the Client encapsulates the RS message in UDP/IPv4 L2 headers, sets the source to the underlay interface IPv4 address and sets the destination to the OMNI IPv4 anycast address. The Client then forwards the message into the IPv4 multihop routing system which conveys it to the nearest Proxy/Server that advertises the corresponding IPv4 prefix. If the nearest Proxy/Server is too busy and/or does not configure the specified OMNI IPv6 anycast address, it should forward (without Proxying) the OAL-encapsulated RS to another nearby Proxy/Server connected to the same IPv4 (multihop) network that configures the OMNI IPv6 anycast address. (In environments where reciprocal RS forwarding cannot be supported, the first Proxy/Server should instead return an RA based on its own MSP(s).)

When an intermediate *NET hop that participates in the routing protocol receives the encapsulated RS, it forwards the message according to its routing tables (note that an intermediate node could be a fixed infrastructure element such as a roadside unit or another MANET/VANET node). This process repeats iteratively until the RS message is received by a penultimate *NET hop within single-hop communications range of a Proxy/Server, which forwards the message to the Proxy/Server.

When a Proxy/Server that configures the OMNI IPv6 anycast OAL destination receives the message, it decapsulates the RS and assumes either the Hub or FHS role (in which case, it forwards the RS to a candidate Hub). The Hub Proxy/Server then prepares an RA message with source address set to its own ULA and destination address set to the RS source address if it is acting only as the Hub (or to the Client ULA-MNP within its ULA subnet prefix if it is also acting as the FHS Proxy/Server). The Hub Proxy/Server then performs OAL encapsulation with the RA OAL source/destination set to the RS OAL destination/source and forwards the RA either to the FHS Proxy/Server or directly to the Client.

When the Hub or FHS Proxy/Server forwards the RA to the Client, it encapsulates the message in L2 encapsulation headers (if necessary) with (src, dst) set to the (dst, src) of the RS L2 encapsulation headers. The Proxy/Server then forwards the message to a *NET node within communications range, which forwards the message according to its routing tables to an intermediate node. The multihop forwarding process within the *NET continues repetitively until the message is delivered to the original Client, which decapsulates the message and performs autoconfiguration the same as if it had received the RA directly from a Proxy/Server on the same physical link. The Client then injects the ULA-MNP into the IPv6 multihop routing system if necessary, then begins using the ULA-MNP as its OAL source address and suspends use of its TLA since it now has a unique address within the FHS Proxy/Server's "Multilink Subnet".

Note: When the RS message includes anycast OAL and/or L2 encapsulation destinations, the FHS Proxy/Server must use the same anycast addresses as the OAL and/or L2 encapsulation sources to support forwarding of the RA message and any initial data packets over any NATs on the path. When the Client receives the RA, it will discover its unicast ULA-MNP and/or L2 encapsulation addresses and can forward future packets using the unicast (instead of anycast) addresses to populate NAT state in the forward path. (If the Client does not have immediate data to send to the FHS Proxy/Server, it can instead send an OAL "bubble" - see Section 6.10.) After the Client begins using unicast OAL/L2 encapsulation addresses in this way, the FHS Proxy/Server should also begin using the same unicast addresses in the reverse direction.

Note: When an OMNI interface configures a TLA, any nodes that forward an encapsulated RS message with the ULA as the OAL source must not consider the message as being specific to a particular OMNI link. TLAs can therefore also serve as the source and destination addresses of unencapsulated IPv6 data communications within the local routing region, and if the TLAs are injected into the local network routing protocol their prefix length must be set to 128.

15.3. DHCPv6-based Prefix Registration

When a Client is not pre-provisioned with an MNP (or, when the Client requires additional MNP delegations), it requests the MS to select MNPs on its behalf and set up the correct routing state. The DHCPv6 service [RFC8415] supports this requirement.

When a Client requires the MS to select MNPs, it sends an RS message with source set to a {TLA,XLA}-RND. If the Client requires only a single MNP delegation, it can then include a OMNI Node Identification sub-option plus an OMNI Neighbor Coordination sub-option with Preflen

set to the length of the desired MNP. If the Client requires multiple MNP delegations and/or more complex DHCPv6 services, it instead includes a DHCPv6 Message sub-option containing a Client Identifier, one or more IA_PD options and a Rapid Commit option then sets the 'msg-type' field to "Solicit", and includes a 3 octet 'transaction-id'. The Client then sets the RS destination to link-scoped All-Routers multicast and sends the message using OAL encapsulation and fragmentation if necessary as discussed above.

When the Hub Proxy/Server receives the RS message, it performs OAL reassembly if necessary. Next, if the RS source is a {TLA,XLA}-RND and/or the OMNI option includes a DHCPv6 message sub-option, the Hub Proxy/Server acts as a "Proxy DHCPv6 Client" in a message exchange with the locally-resident DHCPv6 server. If the RS did not contain a DHCPv6 message sub-option, the Hub Proxy/Server generates a DHCPv6 Solicit message on behalf of the Client using an IA_PD option with the prefix length set to the OMNI Neighbor Coordination header Preflen value and with a Client Identifier formed from the OMNI option Node Identification sub-option; otherwise, the Hub Proxy/Server uses the DHCPv6 Solicit message contained in the OMNI option. The Hub Proxy/Server then sends the DHCPv6 message to the DHCPv6 Server, which delegates MNPs and returns a DHCPv6 Reply message with PD parameters. (If the Hub Proxy/Server wishes to defer creation of Client state until the DHCPv6 Reply is received, it can instead act as a Lightweight DHCPv6 Relay Agent per [RFC6221] by encapsulating the DHCPv6 message in a Relay-forward/reply exchange with Relay Message and Interface ID options. In the process, the Hub Proxy/Server packs any state information needed to return an RA to the Client in the Relay-forward Interface ID option so that the information will be echoed back in the Relay-reply.)

When the Hub Proxy/Server receives the DHCPv6 Reply, it creates XLA-MNPs based on the delegated MNPs and creates OMNI interface XLA-MNP forwarding table entries (i.e., to prompt the dynamic routing protocol). The Hub Proxy/Server then sends an RA back to the FHS Proxy/Server with the DHCPv6 Reply message included in an OMNI DHCPv6 message sub-option. The Hub Proxy/Server sets the RA destination address to the RS source address, sets the RA source address to its own ULA, performs OAL encapsulation and fragmentation, performs L2 encapsulation and sends the RA to the Client via the FHS Proxy/Server as discussed above.

When the FHS Proxy/Server receives the RA, it changes the RA destination address to the ULA-MNP for the Client within its own ULA subnet prefix then forwards the RA to the Client. When the Client receives the RA, it reassembles and discards the OAL encapsulation then creates a default route, assigns Subnet Router Anycast addresses and uses the RA destination address or DHCPv6-delegated MNP to

automatically configure its primary ULA-MNP. The Client will then use these primary MNP-based addresses as the source address of any IPv6 ND messages it sends as long as it retains ownership of the MNP.

Note: when the Hub Proxy/Server is also the FHS Proxy/Server, it forwards the RA message directly to the Client with the destination set to the Client's ULA-MNP (i.e., instead of forwarding via another Proxy/Server).

15.4. OMNI Link Extension

Clients can provide an OMNI link ingress point for other nodes on their (downstream) ENETs that also act as Clients. When Client A has already coordinated with an (upstream) ANET/INET Proxy/Server, Client B on an ENET serviced by Client A can send OAL-encapsulated RS messages with addresses set the same as specified in Section 15.2. When Client A receives the RS message, it infers from the OAL encapsulation that Client B is seeking to establish itself as a Client instead of just a simple ENET Host.

Client A then returns an RA message the same as a Proxy/Server would do as specified in Section 15.2 except that it instead uses its own ULA-MNP as the RA and OAL source addresses and performs (recursive) DHCPv6 Prefix Delegation. The MNP delegation in the RA message must be a sub-MNP from the MNP delegated to Client A. For example, if Client A receives the MNP 2001:db8:1000::/48 it can provide a sub-delegation such as 2001:db8:1000:2000::/56 to Client B. Client B can in turn sub-delegate 2001:db8:1000:2000::/56 to its own ENET(s), where there may be a further prospective Client C that would in turn request OMNI link services via Client B.

To support this Client-to-Client chaining, Clients send IPv6 ND messages addressed to the OMNI link anycast address via their ANET/INET (i.e., upstream) interfaces, but advertise the OMNI link anycast address into their ENET (i.e., downstream) networks where there may be further prospective Clients wishing to join the chain. The ENET of the upstream Client is therefore seen as an ANET by downstream Clients, and the upstream Client is seen as a Proxy/Server by downstream Clients.

16. Secure Redirection

If the underlay network link model is multiple access, the FHS Proxy/Server is responsible for assuring that address duplication cannot corrupt the neighbor caches of other nodes on the link. When the Client sends an RS message on a multiple access underlay network, the Proxy/Server verifies that the Client is authorized to use the address and responds with an RA (or forwards the RS to the Hub) only

if the Client is authorized.

After verifying Client authorization and returning an RA, the Proxy/Server MAY return IPv6 ND Redirect messages to direct Clients located on the same underlay network to exchange packets directly without transiting the Proxy/Server. In that case, the Clients can exchange packets according to their unicast L2 addresses discovered from the Redirect message instead of using the dogleg path through the Proxy/Server. In some underlay networks, however, such direct communications may be undesirable and continued use of the dogleg path through the Proxy/Server may provide better performance. In that case, the Proxy/Server can refrain from sending Redirects, and/or Clients can ignore them.

17. Proxy/Server Resilience

*NETs SHOULD deploy Proxy/Servers in Virtual Router Redundancy Protocol (VRRP) [RFC5798] configurations so that service continuity is maintained even if one or more Proxy/Servers fail. Using VRRP, the Client is unaware which of the (redundant) FHS Proxy/Servers is currently providing service, and any service discontinuity will be limited to the failover time supported by VRRP. Widely deployed public domain implementations of VRRP are available.

Proxy/Servers SHOULD use high availability clustering services so that multiple redundant systems can provide coordinated response to failures. As with VRRP, widely deployed public domain implementations of high availability clustering services are available. Note that special-purpose and expensive dedicated hardware is not necessary, and public domain implementations can be used even between lightweight virtual machines in cloud deployments.

18. Detecting and Responding to Proxy/Server Failures

In environments where fast recovery from Proxy/Server failure is required, FHS Proxy/Servers SHOULD use proactive Neighbor Unreachability Detection (NUD) in a manner that parallels Bidirectional Forwarding Detection (BFD) [RFC5880] to track Hub Proxy/Server reachability. FHS Proxy/Servers can then quickly detect and react to failures so that cached information is re-established through alternate paths. Proactive NUD control messaging is carried only over well-connected ground domain networks (i.e., and not low-end links such as aeronautical radios) and can therefore be tuned for rapid response.

FHS Proxy/Servers perform proactive NUD for Hub Proxy/Servers for which there are currently active Clients. If a Hub Proxy/Server fails, the FHS Proxy/Server can quickly inform Clients of the outage

by sending multicast RA messages. The FHS Proxy/Server sends RA messages to Clients with source set to the ULA of the Hub, with destination address set to All-Nodes multicast (ff02::1) [RFC4291] and with Router Lifetime set to 0.

The FHS Proxy/Server SHOULD send MAX_FINAL_RTR_ADVERTISEMENTS RA messages separated by small delays [RFC4861]. Any Clients that have been using the (now defunct) Hub Proxy/Server will receive the RA messages.

19. Transition Considerations

When a Client connects to an *NET link for the first time, it sends an RS message with an OMNI option. If the first hop router recognizes the option, it responds according to the appropriate FHS/Hub Proxy/Server role resulting in an RA message with an OMNI option returned to the Client. The Client then engages this FHS Proxy/Server according to the OMNI link model specified above. If the first hop router is a legacy IPv6 router, however, it instead returns an RA message with no OMNI option and with a non-OMNI unicast source LLA as specified in [RFC4861]. In that case, the Client engages the *NET according to the legacy IPv6 link model and without the OMNI extensions specified in this document.

If the *NET link model is multiple access, there must be assurance that address duplication cannot corrupt the neighbor caches of other nodes on the link. When the Client sends an RS message on a multiple access *NET link with an OMNI option, first hop routers that recognize the option ensure that the Client is authorized to use the address and return an RA with a non-zero Router Lifetime only if the Client is authorized. First hop routers that do not recognize the OMNI option instead return an RA that makes no statement about the Client's authorization to use the source address. In that case, the Client should perform Duplicate Address Detection to ensure that it does not interfere with other nodes on the link.

An alternative approach for multiple access *NET links to ensure isolation for Client-Proxy/Server communications is through link-layer address mappings as discussed in Appendix D. This arrangement imparts a (virtual) point-to-point link model over the (physical) multiple access link.

20. OMNI Interfaces on Open Internetworks

Client OMNI interfaces configured over IPv6-enabled underlay interfaces on an open Internetwork without an OMNI-aware first-hop router receive IPv6 RA messages with no OMNI options, while OMNI interfaces configured over IPv4-only underlay interfaces receive no IPv6 RA messages at all (but may receive IPv4 RA messages [RFC1256]). Client OMNI interfaces that receive RA messages with OMNI options configure addresses, on-link prefixes, etc. on the underlay interface that received the RA according to standard IPv6 ND and address resolution conventions [RFC4861] [RFC4862]. Client OMNI interfaces configured over IPv4-only underlay interfaces configure IPv4 address information on the underlay interfaces using mechanisms such as DHCPv4 [RFC2131].

Client OMNI interfaces configured over underlay interfaces connected to open Internetworks can apply security services such as VPNs to connect to a Proxy/Server, or can establish a direct link to the Proxy/Server through some other means (see Section 4). In environments where an explicit VPN or direct link may be impractical or undesirable, Client OMNI interfaces can instead send IPv6 ND messages with OMNI options that include authentication signatures.

OMNI interfaces use UDP/IP as L2 encapsulation headers for transmission over open Internetworks with UDP service port number 8060 (see: Section 25.13 and Section 3.6 of [I-D.templin-6man-aero]) for both IPv4 and IPv6 underlay interfaces. The OMNI interface submits original IP packets for OAL encapsulation, then encapsulates the resulting OAL fragments in UDP/IP L2 headers to form carrier packets. (The first four bits following the UDP header determine whether the OAL headers are uncompressed/compressed as discussed in Section 6.4.) The OMNI interface sets the UDP length to the encapsulated OAL fragment length and sets the IP length to an appropriate value at least as large as the UDP datagram.

For Client-Proxy/Server (e.g., "Vehicle-to-Infrastructure (V2I)") neighbor exchanges, the source must include an OMNI option with an authentication sub-option in all IPv6 ND messages. The source can apply HIP security services per [RFC7401] using the IPv6 ND message OMNI option as a "shipping container" to convey an authentication signature in a (unidirectional) HIP "Notify" message. For Client-Client (e.g., "Vehicle-to-Vehicle (V2V)") neighbor exchanges, two Clients can exchange HIP "Initiator/Responder" messages coded in OMNI options of multiple IPv6 NS/NA messages for mutual authentication according to the HIP protocol. (Note: a simple Hashed Message Authentication Code (HMAC) such as specified in [RFC4380] or the QUIC-TLS connection-oriented service [RFC9000] can be used as an alternate authentication service in some environments.)

When an OMNI interface includes an authentication sub-option, it must appear as the first sub-option of the first OMNI option in the IPv6 ND message which must appear immediately following the IPv6 ND message header. When an OMNI interface prepares a HIP message sub-option, it includes its own (H)HIT as the Sender's HIT and the neighbor's (H)HIT if known as the Receiver's HIT (otherwise 0). If (H)HITs are not available within the OMNI operational environment, the source can instead include other IPv6 address types instead of (H)HITs as long as the Sender and Receiver have some way to associate information embedded in the IPv6 address with the neighbor; such information could include a node identifier, vehicle identifier, MAC address, etc.

Before calculating the authentication signature, the source includes any other necessary sub-options (such as Interface Attributes and Origin Indication) and sets both the IPv6 ND message Checksum and authentication signature fields to 0. The source then calculates the authentication signature over the full length of the IPv6 ND message beginning with a pseudo-header of the IPv6 header (i.e., the same as specified in [RFC4443]) and extending over the length of the message. (If the IPv6 ND message is part of an OAL super-packet, the source instead calculates the authentication signature over the remainder of the super-packet.) The source next writes the authentication signature into the sub-option signature field and forwards the message.

After establishing a VPN or preparing for UDP/IP encapsulation, OMNI interfaces send RS/RA messages for Client-Proxy/Server coordination (see: Section 15) and NS/NA messages for route optimization, window synchronization and mobility management (see: [I-D.templin-6man-aero]). These control plane messages must be authenticated while other control and data plane messages are delivered the same as for ordinary best-effort traffic with source address and/or Identification window-based data origin verification. Upper layer protocol sessions over OMNI interfaces that connect over open Internetworks without an explicit VPN should therefore employ transport- or higher-layer security to ensure authentication, integrity and/or confidentiality.

Clients should avoid using INET Proxy/Servers as general-purpose routers for steady streams of carrier packets that do not require authentication. Clients should instead perform route optimization to coordinate with other INET nodes that can provide forwarding services (or preferably coordinate directly with peer Clients directly) instead of burdening the Proxy/Server. Procedures for coordinating with peer Clients and discovering INET nodes that can provide better forwarding services are discussed in [I-D.templin-6man-aero].

Clients that attempt to contact peers over INET underlay interfaces often encounter NATs in the path. OMNI interfaces accommodate NAT traversal using UDP/IP encapsulation and the mechanisms discussed in [I-D.templin-6man-aero]. FHS Proxy/Servers include Origin Indications in RA messages to allow Clients to detect the presence of NATs.

Note: Following the initial IPv6 ND message exchange, OMNI interfaces configured over INET underlay interfaces maintain neighbor relationships by transmitting periodic IPv6 ND messages with OMNI options that include HIP "Update" and/or "Notify" messages. When HMAC authentication is used instead of HIP, the Client and Proxy/Server exchange all IPv6 ND messages with HMAC signatures included based on a shared-secret. When QUIC-TLS connections are used, the Client and Proxy/Server observe QUIC-TLS conventions [RFC9000][RFC9001].

Note: OMNI interfaces configured over INET underlay interfaces should employ the Identification window synchronization mechanisms specified in Section 6.6 in order to exclude spurious carrier packets that might otherwise clutter the reassembly cache. This is especially important in environments where carrier packet spoofing and/or corruption is a threat.

Note: NATs may be present on the path from a Client to its FHS Proxy/Server, but never on the path from the FHS Proxy/Server to the Hub where only INET and/or spanning tree hops occur. Therefore, the FHS Proxy/Server does not communicate Client origin information to the Hub where it would serve no purpose.

21. Time-Varying MNPs

In some use cases, it is desirable, beneficial and efficient for the Client to receive a constant MNP that travels with the Client wherever it moves. For example, this would allow air traffic controllers to easily track aircraft, etc. In other cases, however (e.g., intelligent transportation systems), the Client may be willing to sacrifice a modicum of efficiency in order to have time-varying MNPs that can be changed every so often to defeat adversarial tracking.

The prefix delegation services discussed in Section 15.3 allows Clients that desire time-varying MNPs to obtain short-lived prefixes to send RS messages with a {TLA,XLA}-RND source address and/or with an OMNI option with DHCPv6 Option sub-options. The Client would then be obligated to renumber its internal networks whenever its MNP (and therefore also its OMNI address) changes. This should not present a challenge for Clients with automated network renumbering services, but may disrupt persistent sessions that would prefer to use a constant address.

22. (H)HITs and Temporary ULA (TLA)s

Clients that generate (H)HITs but do not have pre-assigned MNPs can request MNP delegations by issuing IPv6 ND messages that use the (H)HIT instead of a TLA. For example, when a Client creates an RS message it can set the source to a (H)HIT and destination to link-scoped All-Routers multicast. The IPv6 ND message includes an OMNI option with a HIP message sub-option, and need not include a Node Identification sub-option if the Client's (H)HIT appears in the HIP message. The Client then encapsulates the message in an IPv6 header with the (H)HIT as the source address. The Client then sends the message as specified in Section 15.2.

When the Hub Proxy/Server receives the RS message, it notes that the source was a (H)HIT, then invokes the DHCPv6 protocol to request an MNP prefix delegation while using the (H)HIT (in the form of a DUID) as the Client Identifier. The Hub Proxy/Server then prepares an RA message with source address set to its own ULA and destination set to the source of the RS message. The Hub Proxy/Server next includes an OMNI option with a HIP message sub-option and any DHCPv6 prefix delegation parameters. The Proxy/Server finally encapsulates the RA in an OAL header with source address set to its own ULA and destination set to the RS OAL source address, then returns the encapsulated RA to the Client either directly or by way of the FHS Proxy/Server as a proxy.

Clients can also use (H)HITs and/or TLAs for direct Client-to-Client communications outside the context of any OMNI link supporting infrastructure. When two Clients encounter one another they can use their (H)HITs and/or TLAs as original IPv6 packet source and destination addresses to support direct communications. Clients can also inject their (H)HITs and/or TLAs into an IPv6 multihop routing protocol to enable multihop communications as discussed in Section 15.2. Clients can further exchange other IPv6 ND messages using their (H)HITs and/or TLAs as source and destination addresses.

Lastly, when Clients are within the coverage range of OMNI link infrastructure a case could be made for injecting (H)HITs and/or TLAs into the global MS routing system. For example, when the Client sends an RS to an FHS Proxy/Server it could include a request to inject the (H)HIT / TLA into the routing system instead of requesting an MNP prefix delegation. This would potentially enable OMNI link-wide communications using only (H)HITs or TLAs, and not MNPs. This document notes the opportunity, but makes no recommendation.

23. Address Selection

Clients assign LLAs to the OMNI interface, but do not use LLAs as IPv6 ND message source/destination addresses nor for addressing ordinary original IP packets exchanged with OMNI link neighbors.

Clients use ULA-MNPs as source/destination IPv6 addresses in the encapsulation headers of OAL packets and use XLA-MNPs as the IPv6 source addresses of the IPv6 ND messages themselves. Clients use TLAs when an MNP is not available, or as source/destination IPv6 addresses for communications within a MANET/VANET local area. Clients can also use (H)HITs instead of ULAs for local communications when operation outside the context of a specific ULA domain and/or source address attestation is necessary.

Clients use MNP-based GUAs as original IP packet source and destination addresses for communications with Internet destinations when they are within range of OMNI link supporting infrastructure that can inject the MNP into the routing system. Clients can also use MNP-based GUAs within multihop routing regions that are currently disconnected from infrastructure as long as the corresponding ULA-MNPs have been injected into the routing system.

Clients use anycast GUAs as OAL and/or L2 encapsulation destination addresses for RS messages used to discover the nearest FHS Proxy/Server. When the Proxy/Server returns a solicited RA, it must also use the same anycast address as the RA OAL/L2 encapsulation source in order to successfully traverse any NATs in the path. The Client should then immediately transition to using the FHS Proxy/Server's discovered unicast OAL/L2 address as the destination in order to minimize dependence on the Proxy/Server's use of an anycast source address.

24. Error Messages

An OAL destination or intermediate node may need to return ICMPv6-like error messages (e.g., Destination Unreachable, Packet Too Big, Time Exceeded, etc.) [RFC4443] to an OAL source. Since ICMPv6 error messages do not themselves include authentication codes, OAL nodes can return error messages as an OMNI ICMPv6 Error sub-option in a secured IPv6 ND uNA message.

25. IANA Considerations

The following IANA actions are requested in accordance with [RFC8126] and [RFC8726]:

25.1. "Protocol Numbers" Registry

The IANA is instructed to allocate an Internet Protocol number TBD1 from the 'protocol numbers' registry for the Overlay Multilink Network Interface (OMNI) protocol. Guidance is found in [RFC5237] (registration procedure is IESG Approval or Standards Action).

25.2. "IEEE 802 Numbers" Registry

During final publication stages, the IESG will be requested to procure an IEEE EtherType value TBD2 for OMNI according to the statement found at <https://www.ietf.org/about/groups/iesg/statements/ethertypes/>.

Following this procurement, the IANA is instructed to register the value TBD2 in the 'ieee-802-numbers' registry for Overlay Multilink Network Interface (OMNI) encapsulation on Ethernet networks. Guidance is found in [RFC7042] (registration procedure is Expert Review).

25.3. "IPv4 Special-Purpose Address" Registry

The IANA is instructed to assign TBD3/N as an "OMNI IPv4 anycast" address/prefix in the "IPv4 Special-Purpose Address" registry in a similar fashion as for [RFC3068]. The IANA is requested to work with the authors to obtain a TBD3/N public IPv4 prefix, whether through an RIR allocation, a delegation from IANA's "IPv4 Recovered Address Space" registry or through an unspecified third party donation.

25.4. "IPv6 Neighbor Discovery Option Formats" Registry

The IANA is instructed to allocate an official Type number TBD4 from the "IPv6 Neighbor Discovery Option Formats" registry for the OMNI option (registration procedure is RFC required). Implementations set Type to 253 as an interim value [RFC4727].

25.5. "Ethernet Numbers" Registry

The IANA is instructed to allocate one Ethernet unicast address TBD5 (suggested value '00-52-14') in the 'ethernet-numbers' registry under "IANA Unicast 48-bit MAC Addresses" (registration procedure is Expert Review). The registration should appear as follows:

Addresses	Usage	Reference
-----	-----	-----
00-52-14	Overlay Multilink Network (OMNI) Interface	[RFCXXXX]

Figure 35: IANA Unicast 48-bit MAC Addresses

25.6. "ICMPv6 Code Fields: Type 2 - Packet Too Big" Registry

The IANA is instructed to assign two new Code values in the "ICMPv6 Code Fields: Type 2 - Packet Too Big" registry (registration procedure is Standards Action or IESG Approval). The registry should appear as follows:

Code	Name	Reference
---	----	-----
0	PTB Hard Error	[RFC4443]
1	PTB Soft Error (loss)	[RFCXXXX]
2	PTB Soft Error (no loss)	[RFCXXXX]

Figure 36: ICMPv6 Code Fields: Type 2 - Packet Too Big Values

(Note: this registry also to be used to define values for setting the "unused" field of ICMPv4 "Destination Unreachable - Fragmentation Needed" messages.)

25.7. "OMNI Option Sub-Type Values" (New Registry)

The OMNI option defines a 5-bit Sub-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Option Sub-Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	Pad1	[RFCXXXX]
1	PadN	[RFCXXXX]
2	Neighbor Coordination	[RFCXXXX]
3	Interface Attributes	[RFCXXXX]
4	Multilink Forwarding Params	[RFCXXXX]
5	Traffic Selector	[RFCXXXX]
6	Geo Coordinates	[RFCXXXX]
7	DHCPv6 Message	[RFCXXXX]
8	HIP Message	[RFCXXXX]
9	PIM-SM Message	[RFCXXXX]
10	Fragmentation Report	[RFCXXXX]
11	Node Identification	[RFCXXXX]
12	ICMPv6 Error	[RFCXXXX]
13	QUIC-TLS Message	[RFCXXXX]
14	Proxy/Server Departure	[RFCXXXX]
15-29	Unassigned	
30	Sub-Type Extension	[RFCXXXX]
31	Reserved by IANA	[RFCXXXX]

Figure 37: OMNI Option Sub-Type Values

25.8. "OMNI Geo Coordinates Type Values" (New Registry)

The OMNI Geo Coordinates sub-option (see: Section 12.2.7) contains an 8-bit Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Geo Coordinates Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	NULL	[RFCXXXX]
1-252	Unassigned	[RFCXXXX]
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 38: OMNI Geo Coordinates Type

25.9. "OMNI Node Identification ID-Type Values" (New Registry)

The OMNI Node Identification sub-option (see: Section 12.2.12) contains an 8-bit ID-Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI Node Identification ID-Type Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	UUID	[RFCXXXX]
1	HIT	[RFCXXXX]
2	HHIT	[RFCXXXX]
3	Network Access Identifier	[RFCXXXX]
4	FQDN	[RFCXXXX]
5	IPv6 Address	[RFCXXXX]
6-252	Unassigned	[RFCXXXX]
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 39: OMNI Node Identification ID-Type Values

25.10. "OMNI Option Sub-Type Extension Values" (New Registry)

The OMNI option defines an 8-bit Extension-Type field for Sub-Type 30 (Sub-Type Extension), for which IANA is instructed to create and maintain a new registry entitled "OMNI Option Sub-Type Extension Values". Initial values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	RFC4380 UDP/IP Header Option	[RFCXXXX]
1	RFC6081 UDP/IP Trailer Option	[RFCXXXX]
2-252	Unassigned	
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 40: OMNI Option Sub-Type Extension Values

25.11. "OMNI RFC4380 UDP/IP Header Option" (New Registry)

The OMNI Sub-Type Extension "RFC4380 UDP/IP Header Option" defines an 8-bit Header Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI RFC4380 UDP/IP Header Option". Initial registry values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	Origin Indication (IPv4)	[RFC4380]
1	Authentication Encapsulation	[RFC4380]
2	Origin Indication (IPv6)	[RFCXXXX]
3-252	Unassigned	
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 41: OMNI RFC4380 UDP/IP Header Option

25.12. "OMNI RFC6081 UDP/IP Trailer Option" (New Registry)

The OMNI Sub-Type Extension for "RFC6081 UDP/IP Trailer Option" defines an 8-bit Trailer Type field, for which IANA is instructed to create and maintain a new registry entitled "OMNI RFC6081 UDP/IP Trailer Option". Initial registry values are given below (registration procedure is RFC required):

Value	Sub-Type name	Reference
-----	-----	-----
0	Unassigned	
1	Nonce	[RFC6081]
2	Unassigned	
3	Alternate Address (IPv4)	[RFC6081]
4	Neighbor Discovery Option	[RFC6081]
5	Random Port	[RFC6081]
6	Alternate Address (IPv6)	[RFCXXXX]
7-252	Unassigned	
253-254	Reserved for Experimentation	[RFCXXXX]
255	Reserved by IANA	[RFCXXXX]

Figure 42: OMNI RFC6081 Trailer Option

25.13. Additional Considerations

The IANA has assigned the UDP port number "8060" for an earlier experimental version of AERO [RFC6706]. This document reclaims the UDP port number "8060" for 'aero' as the service port for UDP/IP encapsulation. (Note that, although [RFC6706] is not widely implemented or deployed, any messages coded to that specification can be easily distinguished and ignored since they include an invalid ICMPv6 message type number '0'.) The IANA is therefore instructed to update the reference for UDP port number "8060" from "RFC6706" to "RFCXXXX" (i.e., this document) while retaining the existing name 'aero'.

The IANA has assigned a 4 octet Private Enterprise Number (PEN) code "45282" in the "enterprise-numbers" registry. This document is the normative reference for using this code in DHCP Unique IDentifiers based on Enterprise Numbers ("DUID-EN for OMNI Interfaces") (see: Section 11). The IANA is therefore instructed to change the enterprise designation for PEN code "45282" from "LinkUp Networks" to "Overlay Multilink Network Interface (OMNI)".

The IANA has assigned the ifType code "301 - omni - Overlay Multilink Network Interface (OMNI)" in accordance with Section 6 of [RFC8892]. The registration appears under the IANA "Structure of Management Information (SMI) Numbers (MIB Module Registrations) - Interface Types (ifType)" registry.

No further IANA actions are required.

26. Security Considerations

Security considerations for IPv4 [RFC0791], IPv6 [RFC8200] and IPv6 Neighbor Discovery [RFC4861] apply. OMNI interface IPv6 ND messages SHOULD include Nonce and Timestamp options [RFC3971] when transaction confirmation and/or time synchronization is needed. (Note however that when OAL encapsulation is used the (echoed) OAL Identification value can provide sufficient transaction confirmation.)

OMNI interfaces configured over secured ANET/ENET interfaces inherit the physical and/or link-layer security properties (i.e., "protected spectrum") of the connected networks. OMNI interfaces configured over open INET interfaces can use symmetric securing services such as VPNs or can by some other means establish a direct link. When a VPN or direct link may be impractical or undesirable, however, the security services specified in [RFC7401], [RFC4380] or [RFC9000] can be employed. While the OMNI link protects control plane messaging, applications must still employ end-to-end transport- or higher-layer security services to protect the data plane.

Strong network layer security for control plane messages and forwarding path integrity for data plane messages between Proxy/Servers MUST be supported. In one example, the AERO service [I-D.templin-6man-aero] constructs an SRT spanning tree with Proxy/Servers as leaf nodes and secures the spanning tree links with network layer security mechanisms such as IPsec [RFC4301] or WireGuard [WG]. Secured control plane messages are then constrained to travel only over the secured spanning tree paths and are therefore protected from attack or eavesdropping. Other control and data plane messages can travel over route optimized paths that do not strictly follow the secured spanning tree, therefore end-to-end sessions should employ transport- or higher-layer security services. Additionally, the OAL Identification value can provide a first level of data origin authentication to mitigate off-path spoofing in some environments.

Identity-based key verification infrastructure services such as iPSK may be necessary for verifying the identities claimed by Clients. This requirement should be harmonized with the manner in which (H)HITs are attested in a given operational environment.

Security considerations for specific access network interface types are covered under the corresponding IP-over-(foo) specification (e.g., [RFC2464], [RFC2492], etc.).

Security considerations for IPv6 fragmentation and reassembly are discussed in Section 6.12. In environments where spoofing is considered a threat, OMNI nodes SHOULD employ Identification window synchronization and OAL destinations SHOULD configure an (end-system-based) firewall.

27. Implementation Status

AERO/OMNI Release-3.2 was tagged on March 30, 2021, and is undergoing internal testing. Additional internal releases expected within the coming months, with first public release expected end of 1H2021.

Many AERO/OMNI functions are implemented and undergoing final integration. OAL fragmentation/reassembly buffer management code has been cleared for public release.

28. Document Updates

This document does not itself update other RFCs, but suggests that the following could be updated through future IETF initiatives:

- * [RFC1191]
- * [RFC2675]
- * [RFC4291]
- * [RFC4443]
- * [RFC8201]

Updates can be through, e.g., standards action, the errata process, etc. as appropriate.

29. Acknowledgements

The first version of this document was prepared per the consensus decision at the 7th Conference of the International Civil Aviation Organization (ICAO) Working Group-I Mobility Subgroup on March 22, 2019. Consensus to take the document forward to the IETF was reached at the 9th Conference of the Mobility Subgroup on November 22, 2019. Attendees and contributors included: Guray Acar, Danny Bharj, Francois D'Humieres, Pavel Drasil, Nikos Fistas, Giovanni Garofolo, Bernhard Haindl, Vaughn Maiolla, Tom McParland, Victor Moreno, Madhu

Niraula, Brent Phillips, Liviu Popescu, Jacky Pouzet, Aloke Roy, Greg Saccone, Robert Segers, Michal Skorepa, Michel Solery, Stephane Tamalet, Fred Templin, Jean-Marc Vacher, Bela Varkonyi, Tony Whyman, Fryderyk Wrobel and Dongsong Zeng.

The following individuals are acknowledged for their useful comments: Amanda Baber, Stuart Card, Donald Eastlake, Adrian Farrel, Michael Matyas, Robert Moskowitz, Madhu Niraula, Greg Saccone, Stephane Tamalet, Eliot Lear, Eduard Vasilenko, Eric Vyncke. Pavel Drasil, Zdenek Jaron and Michal Skorepa are especially recognized for their many helpful ideas and suggestions. Akash Agarwal, Madhuri Madhava Badgandi, Sean Dickson, Don Dillenburg, Joe Dudkowski, Vijayasarathy Rajagopalan, Ron Sackman, Bhargava Raman Sai Prakash and Katherine Tran are acknowledged for their hard work on the implementation and technical insights that led to improvements for the spec.

Discussions on the IETF 6man and atn mailing lists during the fall of 2020 suggested additional points to consider. The authors gratefully acknowledge the list members who contributed valuable insights through those discussions. Eric Vyncke and Erik Kline were the intarea ADs, while Bob Hinden and Ole Troan were the 6man WG chairs at the time the document was developed; they are all gratefully acknowledged for their many helpful insights. Many of the ideas in this document have further built on IETF experiences beginning in the 1990s, with insights from colleagues including Ron Bonica, Brian Carpenter, Ralph Droms, Christian Huitema, Thomas Narten, Dave Thaler, Joe Touch, Pascal Thubert, and many others who deserve recognition.

Early observations on IP fragmentation performance implications were noted in the 1986 Digital Equipment Corporation (DEC) "qe reset" investigation, where fragment bursts from NFS UDP traffic triggered hardware resets resulting in communication failures. Jeff Chase, Fred Glover and Chet Juzszczak of the Ultrix Engineering Group led the investigation, and determined that setting a smaller NFS mount block size reduced the amount of fragmentation and suppressed the resets. Early observations on L2 media MTU issues were noted in the 1988 DEC FDDI investigation, where Raj Jain, KK Ramakrishnan and Kathy Wilde represented architectural considerations for FDDI networking in general including FDDI/Ethernet bridging. Jeff Mogul (who led the IETF Path MTU Discovery working group) and other DEC colleagues who supported these early investigations are also acknowledged.

Throughout the 1990's and into the 2000's, many colleagues supported and encouraged continuation of the work. Beginning with the DEC Project Sequoia effort at the University of California, Berkeley, then moving to the DEC research lab offices in Palo Alto CA, then to Sterling Software at the NASA Ames Research Center, then to SRI in

Menlo Park, CA, then to Nokia in Mountain View, CA and finally to the Boeing Company in 2005 the work saw continuous advancement through the encouragement of many. Those who offered their support and encouragement are gratefully acknowledged.

This work is aligned with the NASA Safe Autonomous Systems Operation (SASO) program under NASA contract number NNA16BD84C.

This work is aligned with the FAA as per the SE2025 contract number DTFAWA-15-D-00030.

This work is aligned with the Boeing Information Technology (BIT) Mobility Vision Lab (MVL) program.

30. References

30.1. Normative References

- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <<https://www.rfc-editor.org/info/rfc768>>.
- [RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, DOI 10.17487/RFC0793, September 1981, <<https://www.rfc-editor.org/info/rfc793>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3971] Arkko, J., Ed., Kempf, J., Zill, B., and P. Nikander, "SEcure Neighbor Discovery (SEND)", RFC 3971, DOI 10.17487/RFC3971, March 2005, <<https://www.rfc-editor.org/info/rfc3971>>.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<https://www.rfc-editor.org/info/rfc4191>>.
- [RFC4193] Hinden, R. and B. Haberman, "Unique Local IPv6 Unicast Addresses", RFC 4193, DOI 10.17487/RFC4193, October 2005, <<https://www.rfc-editor.org/info/rfc4193>>.

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <<https://www.rfc-editor.org/info/rfc4443>>.
- [RFC4727] Fenner, B., "Experimental Values In IPv4, IPv6, ICMPv4, ICMPv6, UDP, and TCP Headers", RFC 4727, DOI 10.17487/RFC4727, November 2006, <<https://www.rfc-editor.org/info/rfc4727>>.
- [RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <<https://www.rfc-editor.org/info/rfc4861>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC6088] Tsirtsis, G., Giarreta, G., Soliman, H., and N. Montavont, "Traffic Selectors for Flow Bindings", RFC 6088, DOI 10.17487/RFC6088, January 2011, <<https://www.rfc-editor.org/info/rfc6088>>.
- [RFC8028] Baker, F. and B. Carpenter, "First-Hop Router Selection by Hosts in a Multi-Prefix Network", RFC 8028, DOI 10.17487/RFC8028, November 2016, <<https://www.rfc-editor.org/info/rfc8028>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <<https://www.rfc-editor.org/info/rfc8200>>.
- [RFC8201] McCann, J., Deering, S., Mogul, J., and R. Hinden, Ed., "Path MTU Discovery for IP version 6", STD 87, RFC 8201, DOI 10.17487/RFC8201, July 2017, <<https://www.rfc-editor.org/info/rfc8201>>.

- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

30.2. Informative References

- [ATN] Maiolla, V., "The OMNI Interface - An IPv6 Air/Ground Interface for Civil Aviation, IETF Liaison Statement #1676, <https://datatracker.ietf.org/liaison/1676/>", 3 March 2020.
- [ATN-IPS] WG-I, ICAO., "ICAO Document 9896 (Manual on the Aeronautical Telecommunication Network (ATN) using Internet Protocol Suite (IPS) Standards and Protocol), Draft Edition 3 (work-in-progress)", 10 December 2020.
- [CKSUM] Stone, J., Greenwald, M., Partridge, C., and J. Hughes, "Performance of Checksums and CRC's Over Real Data, IEEE/ACM Transactions on Networking, Vol. 6, No. 5", October 1998.
- [CRC] Jain, R., "Error Characteristics of Fiber Distributed Data Interface (FDDI), IEEE Transactions on Communications", August 1990.
- [I-D.ietf-drip-rid]
Moskowitz, R., Card, S. W., Wiethuechter, A., and A. Gurtov, "DRIP Entity Tag (DET) for Unmanned Aircraft System Remote ID (UAS RID)", Work in Progress, Internet-Draft, draft-ietf-drip-rid-24, 24 April 2022, <<https://www.ietf.org/archive/id/draft-ietf-drip-rid-24.txt>>.
- [I-D.ietf-intarea-tunnels]
Touch, J. and M. Townsley, "IP Tunnels in the Internet Architecture", Work in Progress, Internet-Draft, draft-ietf-intarea-tunnels-10, 12 September 2019, <<https://www.ietf.org/archive/id/draft-ietf-intarea-tunnels-10.txt>>.

[I-D.ietf-ipwave-vehicular-networking]

Jeong, J. (., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", Work in Progress, Internet-Draft, draft-ietf-ipwave-vehicular-networking-28, 30 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-ipwave-vehicular-networking-28.txt>>.

[I-D.templin-6man-aero]

Templin, F. L., "Automatic Extended Route Optimization (AERO)", Work in Progress, Internet-Draft, draft-templin-6man-aero-45, 22 April 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-aero-45.txt>>.

[I-D.templin-6man-fragrep]

Templin, F. L., "IPv6 Fragment Retransmission and Path MTU Discovery Soft Errors", Work in Progress, Internet-Draft, draft-templin-6man-fragrep-07, 29 March 2022, <<https://www.ietf.org/archive/id/draft-templin-6man-fragrep-07.txt>>.

[I-D.templin-6man-lla-type]

Templin, F. L., "The IPv6 Link-Local Address Type Field", Work in Progress, Internet-Draft, draft-templin-6man-lla-type-02, 23 November 2020, <<https://www.ietf.org/archive/id/draft-templin-6man-lla-type-02.txt>>.

[I-D.templin-intarea-parcels]

Templin, F. L., "IP Parcels", Work in Progress, Internet-Draft, draft-templin-intarea-parcels-10, 29 March 2022, <<https://www.ietf.org/archive/id/draft-templin-intarea-parcels-10.txt>>.

[IPV4-GUA] Postel, J., "IPv4 Address Space Registry,

<https://www.iana.org/assignments/ipv4-address-space/ipv4-address-space.xhtml>", 14 December 2020.

[IPV6-GUA] Postel, J., "IPv6 Global Unicast Address Assignments,

<https://www.iana.org/assignments/ipv6-unicast-address-assignments/ipv6-unicast-address-assignments.xhtml>", 14 December 2020.

[RFC1035]

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<https://www.rfc-editor.org/info/rfc1035>>.

- [RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <<https://www.rfc-editor.org/info/rfc1122>>.
- [RFC1146] Zweig, J. and C. Partridge, "TCP alternate checksum options", RFC 1146, DOI 10.17487/RFC1146, March 1990, <<https://www.rfc-editor.org/info/rfc1146>>.
- [RFC1149] Waitzman, D., "Standard for the transmission of IP datagrams on avian carriers", RFC 1149, DOI 10.17487/RFC1149, April 1990, <<https://www.rfc-editor.org/info/rfc1149>>.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", RFC 1191, DOI 10.17487/RFC1191, November 1990, <<https://www.rfc-editor.org/info/rfc1191>>.
- [RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <<https://www.rfc-editor.org/info/rfc1256>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<https://www.rfc-editor.org/info/rfc2464>>.
- [RFC2473] Conta, A. and S. Deering, "Generic Packet Tunneling in IPv6 Specification", RFC 2473, DOI 10.17487/RFC2473, December 1998, <<https://www.rfc-editor.org/info/rfc2473>>.
- [RFC2492] Armitage, G., Schuler, P., and M. Jork, "IPv6 over ATM Networks", RFC 2492, DOI 10.17487/RFC2492, January 1999, <<https://www.rfc-editor.org/info/rfc2492>>.
- [RFC2675] Borman, D., Deering, S., and R. Hinden, "IPv6 Jumbograms", RFC 2675, DOI 10.17487/RFC2675, August 1999, <<https://www.rfc-editor.org/info/rfc2675>>.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <<https://www.rfc-editor.org/info/rfc2863>>.

- [RFC2923] Lahey, K., "TCP Problems with Path MTU Discovery", RFC 2923, DOI 10.17487/RFC2923, September 2000, <<https://www.rfc-editor.org/info/rfc2923>>.
- [RFC2983] Black, D., "Differentiated Services and Tunnels", RFC 2983, DOI 10.17487/RFC2983, October 2000, <<https://www.rfc-editor.org/info/rfc2983>>.
- [RFC3056] Carpenter, B. and K. Moore, "Connection of IPv6 Domains via IPv4 Clouds", RFC 3056, DOI 10.17487/RFC3056, February 2001, <<https://www.rfc-editor.org/info/rfc3056>>.
- [RFC3068] Huitema, C., "An Anycast Prefix for 6to4 Relay Routers", RFC 3068, DOI 10.17487/RFC3068, June 2001, <<https://www.rfc-editor.org/info/rfc3068>>.
- [RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <<https://www.rfc-editor.org/info/rfc3168>>.
- [RFC3330] IANA, "Special-Use IPv4 Addresses", RFC 3330, DOI 10.17487/RFC3330, September 2002, <<https://www.rfc-editor.org/info/rfc3330>>.
- [RFC3366] Fairhurst, G. and L. Wood, "Advice to link designers on link Automatic Repeat reQuest (ARQ)", BCP 62, RFC 3366, DOI 10.17487/RFC3366, August 2002, <<https://www.rfc-editor.org/info/rfc3366>>.
- [RFC3684] Ogier, R., Templin, F., and M. Lewis, "Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)", RFC 3684, DOI 10.17487/RFC3684, February 2004, <<https://www.rfc-editor.org/info/rfc3684>>.
- [RFC3692] Narten, T., "Assigning Experimental and Testing Numbers Considered Useful", BCP 82, RFC 3692, DOI 10.17487/RFC3692, January 2004, <<https://www.rfc-editor.org/info/rfc3692>>.
- [RFC3810] Vida, R., Ed. and L. Costa, Ed., "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, DOI 10.17487/RFC3810, June 2004, <<https://www.rfc-editor.org/info/rfc3810>>.

- [RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <<https://www.rfc-editor.org/info/rfc3819>>.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <<https://www.rfc-editor.org/info/rfc4122>>.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <<https://www.rfc-editor.org/info/rfc4301>>.
- [RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <<https://www.rfc-editor.org/info/rfc4380>>.
- [RFC4389] Thaler, D., Talwar, M., and C. Patel, "Neighbor Discovery Proxies (ND Proxy)", RFC 4389, DOI 10.17487/RFC4389, April 2006, <<https://www.rfc-editor.org/info/rfc4389>>.
- [RFC4429] Moore, N., "Optimistic Duplicate Address Detection (DAD) for IPv6", RFC 4429, DOI 10.17487/RFC4429, April 2006, <<https://www.rfc-editor.org/info/rfc4429>>.
- [RFC4541] Christensen, M., Kimball, K., and F. Solensky, "Considerations for Internet Group Management Protocol (IGMP) and Multicast Listener Discovery (MLD) Snooping Switches", RFC 4541, DOI 10.17487/RFC4541, May 2006, <<https://www.rfc-editor.org/info/rfc4541>>.
- [RFC4605] Fenner, B., He, H., Haberman, B., and H. Sandick, "Internet Group Management Protocol (IGMP) / Multicast Listener Discovery (MLD)-Based Multicast Forwarding ("IGMP/MLD Proxying")", RFC 4605, DOI 10.17487/RFC4605, August 2006, <<https://www.rfc-editor.org/info/rfc4605>>.
- [RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, DOI 10.17487/RFC4821, March 2007, <<https://www.rfc-editor.org/info/rfc4821>>.
- [RFC4963] Heffner, J., Mathis, M., and B. Chandler, "IPv4 Reassembly Errors at High Data Rates", RFC 4963, DOI 10.17487/RFC4963, July 2007, <<https://www.rfc-editor.org/info/rfc4963>>.

- [RFC5213] Gundavelli, S., Ed., Leung, K., Devarapalli, V., Chowdhury, K., and B. Patil, "Proxy Mobile IPv6", RFC 5213, DOI 10.17487/RFC5213, August 2008, <<https://www.rfc-editor.org/info/rfc5213>>.
- [RFC5214] Templin, F., Gleeson, T., and D. Thaler, "Intra-Site Automatic Tunnel Addressing Protocol (ISATAP)", RFC 5214, DOI 10.17487/RFC5214, March 2008, <<https://www.rfc-editor.org/info/rfc5214>>.
- [RFC5237] Arkko, J. and S. Bradner, "IANA Allocation Guidelines for the Protocol Field", BCP 37, RFC 5237, DOI 10.17487/RFC5237, February 2008, <<https://www.rfc-editor.org/info/rfc5237>>.
- [RFC5558] Templin, F., Ed., "Virtual Enterprise Traversal (VET)", RFC 5558, DOI 10.17487/RFC5558, February 2010, <<https://www.rfc-editor.org/info/rfc5558>>.
- [RFC5614] Ogier, R. and P. Spagnolo, "Mobile Ad Hoc Network (MANET) Extension of OSPF Using Connected Dominating Set (CDS) Flooding", RFC 5614, DOI 10.17487/RFC5614, August 2009, <<https://www.rfc-editor.org/info/rfc5614>>.
- [RFC5798] Nadas, S., Ed., "Virtual Router Redundancy Protocol (VRRP) Version 3 for IPv4 and IPv6", RFC 5798, DOI 10.17487/RFC5798, March 2010, <<https://www.rfc-editor.org/info/rfc5798>>.
- [RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, DOI 10.17487/RFC5880, June 2010, <<https://www.rfc-editor.org/info/rfc5880>>.
- [RFC5889] Baccelli, E., Ed. and M. Townsley, Ed., "IP Addressing Model in Ad Hoc Networks", RFC 5889, DOI 10.17487/RFC5889, September 2010, <<https://www.rfc-editor.org/info/rfc5889>>.
- [RFC5942] Singh, H., Beebe, W., and E. Nordmark, "IPv6 Subnet Model: The Relationship between Links and Subnet Prefixes", RFC 5942, DOI 10.17487/RFC5942, July 2010, <<https://www.rfc-editor.org/info/rfc5942>>.
- [RFC6081] Thaler, D., "Teredo Extensions", RFC 6081, DOI 10.17487/RFC6081, January 2011, <<https://www.rfc-editor.org/info/rfc6081>>.

- [RFC6214] Carpenter, B. and R. Hinden, "Adaptation of RFC 1149 for IPv6", RFC 6214, DOI 10.17487/RFC6214, April 2011, <<https://www.rfc-editor.org/info/rfc6214>>.
- [RFC6221] Miles, D., Ed., Ooghe, S., Dec, W., Krishnan, S., and A. Kavanagh, "Lightweight DHCPv6 Relay Agent", RFC 6221, DOI 10.17487/RFC6221, May 2011, <<https://www.rfc-editor.org/info/rfc6221>>.
- [RFC6247] Eggert, L., "Moving the Undeployed TCP Extensions RFC 1072, RFC 1106, RFC 1110, RFC 1145, RFC 1146, RFC 1379, RFC 1644, and RFC 1693 to Historic Status", RFC 6247, DOI 10.17487/RFC6247, May 2011, <<https://www.rfc-editor.org/info/rfc6247>>.
- [RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, November 2011, <<https://www.rfc-editor.org/info/rfc6438>>.
- [RFC6543] Gundavelli, S., "Reserved IPv6 Interface Identifier for Proxy Mobile IPv6", RFC 6543, DOI 10.17487/RFC6543, May 2012, <<https://www.rfc-editor.org/info/rfc6543>>.
- [RFC6706] Templin, F., Ed., "Asymmetric Extended Route Optimization (AERO)", RFC 6706, DOI 10.17487/RFC6706, August 2012, <<https://www.rfc-editor.org/info/rfc6706>>.
- [RFC6935] Eubanks, M., Chimento, P., and M. Westerlund, "IPv6 and UDP Checksums for Tunneled Packets", RFC 6935, DOI 10.17487/RFC6935, April 2013, <<https://www.rfc-editor.org/info/rfc6935>>.
- [RFC6936] Fairhurst, G. and M. Westerlund, "Applicability Statement for the Use of IPv6 UDP Datagrams with Zero Checksums", RFC 6936, DOI 10.17487/RFC6936, April 2013, <<https://www.rfc-editor.org/info/rfc6936>>.
- [RFC6980] Gont, F., "Security Implications of IPv6 Fragmentation with IPv6 Neighbor Discovery", RFC 6980, DOI 10.17487/RFC6980, August 2013, <<https://www.rfc-editor.org/info/rfc6980>>.
- [RFC7042] Eastlake 3rd, D. and J. Abley, "IANA Considerations and IETF Protocol and Documentation Usage for IEEE 802 Parameters", BCP 141, RFC 7042, DOI 10.17487/RFC7042, October 2013, <<https://www.rfc-editor.org/info/rfc7042>>.

- [RFC7094] McPherson, D., Oran, D., Thaler, D., and E. Osterweil, "Architectural Considerations of IP Anycast", RFC 7094, DOI 10.17487/RFC7094, January 2014, <<https://www.rfc-editor.org/info/rfc7094>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7401] Moskowitz, R., Ed., Heer, T., Jokela, P., and T. Henderson, "Host Identity Protocol Version 2 (HIPv2)", RFC 7401, DOI 10.17487/RFC7401, April 2015, <<https://www.rfc-editor.org/info/rfc7401>>.
- [RFC7421] Carpenter, B., Ed., Chown, T., Gont, F., Jiang, S., Petrescu, A., and A. Yourtchenko, "Analysis of the 64-bit Boundary in IPv6 Addressing", RFC 7421, DOI 10.17487/RFC7421, January 2015, <<https://www.rfc-editor.org/info/rfc7421>>.
- [RFC7542] DeKok, A., "The Network Access Identifier", RFC 7542, DOI 10.17487/RFC7542, May 2015, <<https://www.rfc-editor.org/info/rfc7542>>.
- [RFC7739] Gont, F., "Security Implications of Predictable Fragment Identification Values", RFC 7739, DOI 10.17487/RFC7739, February 2016, <<https://www.rfc-editor.org/info/rfc7739>>.
- [RFC7761] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., and L. Zheng, "Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised)", STD 83, RFC 7761, DOI 10.17487/RFC7761, March 2016, <<https://www.rfc-editor.org/info/rfc7761>>.
- [RFC7847] Melia, T., Ed. and S. Gundavelli, Ed., "Logical-Interface Support for IP Hosts with Multi-Access Support", RFC 7847, DOI 10.17487/RFC7847, May 2016, <<https://www.rfc-editor.org/info/rfc7847>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

- [RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <<https://www.rfc-editor.org/info/rfc8402>>.
- [RFC8726] Farrel, A., "How Requests for IANA Action Will Be Handled on the Independent Stream", RFC 8726, DOI 10.17487/RFC8726, November 2020, <<https://www.rfc-editor.org/info/rfc8726>>.
- [RFC8892] Thaler, D. and D. Romascanu, "Guidelines and Registration Procedures for Interface Types and Tunnel Types", RFC 8892, DOI 10.17487/RFC8892, August 2020, <<https://www.rfc-editor.org/info/rfc8892>>.
- [RFC8899] Fairhurst, G., Jones, T., Tüxen, M., Rüngeler, I., and T. Völker, "Packetization Layer Path MTU Discovery for Datagram Transports", RFC 8899, DOI 10.17487/RFC8899, September 2020, <<https://www.rfc-editor.org/info/rfc8899>>.
- [RFC8900] Bonica, R., Baker, F., Huston, G., Hinden, R., Troan, O., and F. Gont, "IP Fragmentation Considered Fragile", BCP 230, RFC 8900, DOI 10.17487/RFC8900, September 2020, <<https://www.rfc-editor.org/info/rfc8900>>.
- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", RFC 8981, DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.
- [RFC9001] Thomson, M., Ed. and S. Turner, Ed., "Using TLS to Secure QUIC", RFC 9001, DOI 10.17487/RFC9001, May 2021, <<https://www.rfc-editor.org/info/rfc9001>>.
- [RFC9002] Iyengar, J., Ed. and I. Swett, Ed., "QUIC Loss Detection and Congestion Control", RFC 9002, DOI 10.17487/RFC9002, May 2021, <<https://www.rfc-editor.org/info/rfc9002>>.
- [WG] WireGuard, W., "WireGuard, Fast, Modern, Secure VPN Tunnel, <https://wireguard.com/>", 7 March 2022.

Appendix A. OAL Checksum Algorithm

The OAL Checksum Algorithm adopts the 8-bit Fletcher algorithm specified in Appendix I of [RFC1146] as also analyzed in [CKSUM]. [RFC6247] declared [RFC1146] historic for the reason that the algorithms had never seen widespread use with TCP, however this document adopts the 8-bit Fletcher algorithm for a different purpose. Quoting from Appendix I of [RFC1146], the OAL Checksum Algorithm proceeds as follows:

"The 8-bit Fletcher Checksum Algorithm is calculated over a sequence of data octets (call them $D[1]$ through $D[N]$) by maintaining 2 unsigned 1's-complement 8-bit accumulators A and B whose contents are initially zero, and performing the following loop where i ranges from 1 to N:

$$A := A + D[i]$$
$$B := B + A$$

It can be shown that at the end of the loop A will contain the 8-bit 1's complement sum of all octets in the datagram, and that B will contain $(N)D[1] + (N-1)D[2] + \dots + D[N]$."

To calculate the OAL checksum, the above algorithm is applied over the N-octet concatenation of the OAL pseudo-header and the encapsulated IP packet or packets. Specifically, the algorithm is first applied over the 40 octets of the OAL pseudo-header as data octets $D[1]$ through $D[40]$, then continues over the entire length of the original IP packet(s) as data octets $D[41]$ through $D[N]$.

Appendix B. IPv6 ND Message Authentication and Integrity

OMNI interface IPv6 ND messages are subject to authentication and integrity checks at multiple levels. When an OMNI interface sends an IPv6 ND message over an INET interface, it includes an authentication sub-option with a valid signature but does not include an IPv6 ND message checksum. The OMNI interface that receives the message verifies the OAL checksum as a first-level integrity check, then verifies the authentication signature (while ignoring the IPv6 ND message checksum) to ensure IPv6 ND message authentication and integrity.

When an OMNI interface sends an IPv6 ND message over an underlay interface connected to a secured network, it omits the authentication sub-option but instead calculates/includes an IPv6 ND message checksum. The OMNI interface that receives the message applies any lower-layer authentication and integrity checks, then verifies both

the OAL checksum and the IPv6 ND message checksum. (Note that optimized implementations can verify both the OAL and IPv6 ND message checksums in a single pass over the data.) When an OMNI interface sends IPv6 ND messages to a synchronized neighbor, it includes an authentication sub-option only if authentication is necessary; otherwise, it calculates/includes the IPv6 ND message checksum.

When the OMNI interface calculates the authentication signature or IPv6 ND message checksum, it performs the calculation beginning with a pseudo-header of the IPv6 ND message header and extends over all following OAL packet data. In particular, for OAL super-packets any additional original IP packets included beyond the end of the IPv6 ND message are simply considered as extensions of the IPv6 ND message for the purpose of the calculation.

OAL destinations discard carrier packets with unacceptable Identifications and submit the encapsulated fragments in all others for reassembly. The reassembly algorithm rejects any fragments with unacceptable sizes, offsets, etc. and reassembles all others. Following reassembly, the OAL checksum algorithm provides an integrity assurance layer that compliments any integrity checks already applied by lower layers as well as a first-pass filter for any checks that will be applied later by upper layers.

Appendix C. VDL Mode 2 Considerations

ICAO Doc 9776 is the "Technical Manual for VHF Data Link Mode 2" (VDLM2) that specifies an essential radio frequency data link service for aircraft and ground stations in worldwide civil aviation air traffic management. The VDLM2 link type is "multicast capable" [RFC4861], but with considerable differences from common multicast links such as Ethernet and IEEE 802.11.

First, the VDLM2 link data rate is only 31.5Kbps - multiple orders of magnitude less than most modern wireless networking gear. Second, due to the low available link bandwidth only VDLM2 ground stations (i.e., and not aircraft) are permitted to send broadcasts, and even so only as compact layer 2 "beacons". Third, aircraft employ the services of ground stations by performing unicast RS/RA exchanges upon receipt of beacons instead of listening for multicast RA messages and/or sending multicast RS messages.

This beacon-oriented unicast RS/RA approach is necessary to conserve the already-scarce available link bandwidth. Moreover, since the numbers of beaconing ground stations operating within a given spatial range must be kept as sparse as possible, it would not be feasible to have different classes of ground stations within the same region observing different protocols. It is therefore highly desirable that all ground stations observe a common language of RS/RA as specified in this document.

Note that links of this nature may benefit from compression techniques that reduce the bandwidth necessary for conveying the same amount of data. The IETF lpwan working group is considering possible alternatives: [<https://datatracker.ietf.org/wg/lpwan/documents>].

Appendix D. Client-Proxy/Server Isolation Through Link-Layer Address Mapping

Per [RFC4861], IPv6 ND messages may be sent to either a multicast or unicast link-scoped IPv6 destination address. However, IPv6 ND messaging should be coordinated between the Client and Proxy/Server only without invoking other nodes on the underlay network. This implies that Client-Proxy/Server control messaging should be isolated and not overheard by other nodes on the link.

To support Client-Proxy/Server isolation on some links, Proxy/Servers can maintain an OMNI-specific unicast link-layer address ("MSADDR"). For Ethernet-compatible links, this specification reserves one Ethernet unicast address TBD5 (see: IANA Considerations). For non-Ethernet statically-addressed links MSADDR is reserved per the assigned numbers authority for the link-layer addressing space. For still other links, MSADDR may be dynamically discovered through other means, e.g., link-layer beacons.

Clients map the L3 addresses of all IPv6 ND messages they send (i.e., both multicast and unicast) to MSADDR instead of to an ordinary unicast or multicast link-layer address. In this way, all of the Client's IPv6 ND messages will be received by Proxy/Servers that are configured to accept packets destined to MSADDR. Note that multiple Proxy/Servers on the link could be configured to accept packets destined to MSADDR, e.g., as a basis for supporting redundancy.

Therefore, Proxy/Servers must accept and process packets destined to MSADDR, while all other devices must not process packets destined to MSADDR. This model has well-established operational experience in Proxy Mobile IPv6 (PMIP) [RFC5213][RFC6543].

Appendix E. Change Log

<< RFC Editor - remove prior to publication >>

Differences from earlier versions:

- * Submit for RFC publication.

Author's Address

Fred L. Templin (editor)
The Boeing Company
P.O. Box 3707
Seattle, WA 98124
United States of America
Email: fltemplin@acm.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 13, 2022

JC. Zuniga
SIGFOX
CJ. Bernardos
UC3M
A. Andersdotter
CENTR
July 12, 2021

MAC address randomization
draft-zuniga-mac-address-randomization-01

Abstract

Internet privacy has become a major concern over the past few years. Users are becoming more aware that their online activity leaves a vast digital footprint, that communications are not always properly secured, and that their location and actions can be easily tracked. One of the main factors for the location tracking issue is the wide use of long-lasting identifiers, such as MAC addresses.

There have been several initiatives at the IETF and the IEEE 802 standards committees to overcome some of these privacy issues. This document provides an overview of these activities, with the intention to inform the technical community about them, and help coordinate between present and futures standardization activities.

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 <https://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 January 13, 2022.

Copyright Notice

Copyright (c) 2021 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 (<https://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	2
2. Terminology	3
3. Background	3
3.1. MAC address usage	3
3.2. MAC address randomization	4
3.3. Privacy Workshop, Tutorial and Experiments at IETF and IEEE 802 meetings	5
4. Recent RCM activities at the IEEE 802	6
5. Recent MAC randomization-related activities at the WBA	7
6. MAC randomization-related activities at the IETF	7
7. OS current practices	9
8. IANA Considerations	10
9. Security Considerations	10
10. Acknowledgments	10
11. References	10
11.1. Normative References	10
11.2. Informative References	11
Authors' Addresses	14

1. Introduction

Internet privacy is becoming a huge concern, as more and more mobile devices are getting directly (e.g., via cellular or Wi-Fi) or indirectly (e.g., via a smartphone using Bluetooth) connected to the Internet. This ubiquitous connectivity, together with not very secure protocol stacks and the lack of proper education about privacy make it very easy to track/monitor the location of users and/or eavesdrop their physical and online activities. This is due to many factors, such as the vast digital footprint that users leave on the Internet, for instance sharing information on social networks, cookies used by browsers and servers to provide a better navigation

experience, connectivity logs that allow tracking of a user's Layer-2 (L2/MAC) or Layer-3 (L3) address, web trackers, etc.; and/or the weak (or even null in some cases) authentication and encryption mechanisms used to secure communications.

This privacy concern affects all layers of the protocol stack, from the lower layers involved in the actual access to the network (e.g., the MAC/Layer-2 and Layer-3 addresses can be used to obtain the location of a user) to higher layer protocol identifiers and user applications [wifi_internet_privacy]. In particular, IEEE 802 MAC addresses have historically been an easy target for tracking users [wifi_tracking].

There have been several initiatives at the IETF and the IEEE 802 standards committees to overcome some of these privacy issues. This document provides an overview of these activities, with the intention to inform the community and help coordinate between present and futures standardization activities.

2. Terminology

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].

The following terms are used in this document:

MAC: Medium Access Control

3. Background

3.1. MAC address usage

Most mobile devices used today are Wi-Fi enabled (i.e. they are equipped with an IEEE 802.11 wireless local area network interface). Wi-Fi interfaces, as any other kind of IEEE 802-based network interface, like Ethernet (i.e. IEEE 802.3) have a Layer-2 address also referred to as MAC address, which can be seen by anybody who can receive the signal transmitted by the network interface. The format of these addresses is shown in Figure 1.

Figure 1: IEEE 802 MAC Address Format (TBD)

MAC addresses can either be universally administered or locally administered. Universally administered and locally administered addresses are distinguished by setting the second-least-significant bit of the most significant byte of the address (the U/L bit).

A universally administered address is uniquely assigned to a device by its manufacturer. Most physical devices are provided with a universally administered address, which is composed of two parts: (i) the Organizationally Unique Identifier (OUI), which are the first three octets in transmission order and identify the organization that issued the identifier, and (ii) Network Interface Controller (NIC) Specific, which are the following three octets, assigned by the organization that manufactured the NIC, in such a way that the resulting MAC address is globally unique.

Locally administered addresses override the burned-in address, and they can either be set-up by the network administrator, or by the Operating System (OS) of the device to which the address pertains. However, as explained in further sections of this document, there are new initiatives at the IEEE 802 and other organizations to specify ways in which these locally administered addresses should be assigned, depending on the use case.

3.2. MAC address randomization

Since universally administered MAC addresses are by definition globally-unique, when a device uses this MAC address to transmit data -especially over the air- it is relatively easy to track this device by simple medium observation. Since a device is usually directly associated to an individual, this poses a privacy concern [link_layer_privacy].

MAC addresses can be easily observed by a third party, such as a passive device listening to communications in the same network. In an 802.11 network, a station exposes its MAC address in two different situations:

- o While actively scanning for available networks, the MAC address is used in the Probe Request frames sent by the device (aka IEEE 802.11 STA).
- o Once associated to a given Access Point (AP), the MAC address is used in frame transmission and reception, as one of the addresses used in the address fields of an IEEE 802.11 frame.

One way to overcome this privacy concern is by using randomly generated MAC addresses. As described in the previous section, the IEEE 802 addressing includes one bit to specify if the hardware address is locally or globally administered. This allows generating local addresses without the need of any global coordination mechanism to ensure that the generated address is still unique within the local network. This feature can be used to generate random addresses, which decouple the globally-unique identifier from the device and

therefore make it more difficult to track a user device from its MAC/L2 address [enhancing_location_privacy].

3.3. Privacy Workshop, Tutorial and Experiments at IETF and IEEE 802 meetings

As an outcome to the STRINT W3C/IAB Workshop [strint], on July 2014 a Tutorial on Pervasive Surveillance of the Internet - Designing Privacy into Internet Protocols was given at the IEEE 802 Plenary meeting in San Diego [privacy_tutorial]. The Tutorial provided an update on the recent developments regarding Internet privacy, the actions that other SDOs such as IETF were taking, and guidelines that were being followed when developing new Internet protocol specifications (e.g. [RFC6973]). The Tutorial highlighted some Privacy concerns applicable specifically to Link Layer technologies and provided suggestions on how IEEE 802 could help addressing them.

Following the discussions and interest within the IEEE 802 community, on 18 July 2014 the IEEE 802 Executive Committee (EC) created an IEEE 802 EC Privacy Recommendation Study Group (SG) [ieee_privacy_ecsg]. The work and discussions from the group have generated multiple outcomes, such as: 802E PAR: Recommended Practice for Privacy Considerations for IEEE 802 Technologies [IEEE_802E], and the 802c PAR: Standard for Local and Metropolitan Area Networks - Overview and Architecture Amendment - Local Medium Access Control (MAC) Address Usage [IEEE_802c].

In order to test the effects of MAC address randomization, major trials were conducted at the IETF and IEEE 802 meetings between November 2014 and March 2015 - IETF91, IETF92 and IEEE 802 Plenary in Berlin. The purpose of the experiments was to evaluate the use of MAC address randomization from two different perspectives: (i) the effect on the connectivity experience of the end-user, also checking if applications and operating systems (OSs) were affected; and (ii) the potential impact on the network infrastructure itself. Some of the findings were published in [wifi_internet_privacy].

During the experiments it was observed that the probability of address duplication in a network with this characteristics is negligible. The experiments also showed that other protocol identifiers can be correlated and therefore be used to still track an individual. Hence, effective privacy tools should not work in isolation at a single layer, but they should be coordinated with other privacy features at higher layers.

Since then, MAC randomization has further been implemented by mobile operating systems to provide better privacy for mobile phone users

when connecting to public wireless networks [privacy_ios], [privacy_windows], [privacy_android].

4. Recent RCM activities at the IEEE 802

Practical experiences of Randomized And Changing MAC Addresses (RCM) in live devices helped researchers fine-tune their understanding of attacks against randomization mechanisms [when_mac_randomization_fails]. At IEEE 802.11 these research experiences eventually formed the basis for a specified mechanism introduced in the IEEE 802.11aq in 2018 which randomize MAC addresses that recommends mechanisms to avoid pitfalls [IEEE_802_11_aq].

More recent developments include turning on MAC randomization in mobile operating systems by default, which has an impact on the ability of network operators to personalize or customize services [rcm_user_experience_csd]. Therefore, follow-on work in the IEEE 802.11 mapped effects of potentially large uptake of randomized MAC identifiers on a number of commonly offered operator services in 2019[rcm_tig_final_report]. In the summer of 2020 this work emanated in two new standards projects with the purpose of developing mechanisms that do not decrease user privacy and enable an optimal user experience when the MAC address of a device in an Extended Service Set is randomized or changes [rcm_user_experience_par] and user privacy solutions applicable to IEEE Std 802.11 [rcm_privacy_par].

The IEEE 802.1 working group has also published a specification that defines a local MAC address space structure, known as the Structured Local Address Plan (SLAP). This structure designates a range of local MAC addresses for protocols using a Company ID (CID) assigned by the IEEE Registration Authority. Another range of local MAC addresses is designated for assignment by administrators. The specification recommends a range of local MAC addresses for use by IEEE 802 protocols [IEEE_802c].

Work within the IEEE 802.1 Security task group on privacy recommendations for all IEEE 802 network technologies has also looked into general recommendations on identifiers, reaching the conclusion that temporary and transient identifiers are preferably in network technology design if there are no compelling reasons of service quality for a newly introduced identifier to be permanent. This work has been specified in the recently published IEEE P802E: Recommended Practice for Privacy Considerations for IEEE 802 Technologies [IEEE_802E]. The IEEE P802E specification will form part of the basis for the review of user privacy solutions applicable to IEEE Std 802.11 (aka Wi-Fi) devices as part of the RCM [rcm_privacy_csd] efforts.

Currently, two task groups in IEEE 802.11 are dealing with issues related to RCM:

- o The IEEE 802.11bh task group, looking at mitigating the repercussions that RCM creates on 802.11 networks and related services, and
- o The IEEE 802.11bi task group, which will define modifications to the IEEE Std 802.11 medium access control (MAC) specification to specify new mechanisms that address and improve user privacy.

5. Recent MAC randomization-related activities at the WBA

At the Wireless Broadband Alliance (WBA), the Testing and Interoperability Work Group has been looking at the issues related to MAC address randomization and has identified a list of potential impacts of these changes to existing systems and solutions, mainly related to Wi-Fi identification.

As part of this work, WBA has documented a set of use cases that a Wi-Fi Identification Standard should address in order to scale and achieve longer term sustainability of deployed services. A first version of this document has been liaised with the IETF as part of the MAC Address Device Identification for Network and Application Services (MADINAS) activities through the "Wi-Fi Identification In a post MAC Randomization Era v1.0" paper [wba_paper].

6. MAC randomization-related activities at the IETF

Several IP address assignment mechanisms such as the IPv6 stateless autoconfiguration techniques (SLAAC) [RFC4862] generate the Interface Identifier (IID) of the address from its MAC address (via EUI64), which then becomes visible to all IPv6 communication peers. This potentially allows for global tracking of a device at L3 from any point on the Internet. Besides, the prefix part of the address provides meaningful insights of the physical location of the device in general, which together with the MAC address-based IID, makes it easier to perform global device tracking.

There are some solutions that might mitigate this privacy threat, such as the use of temporary addresses [RFC4191], the use of opaque IIDs [RFC7217], [I-D.gont-6man-deprecate-eui64-based-addresses]. Next, we briefly describe how these solutions work.

[RFC4191] identifies and describes the privacy issues associated with embedding MAC stable addressing information into the IPv6 addresses (as part of the IID) and describes some mechanisms to mitigate the associated problems. The specification is meant for IPv6 nodes that

auto-configure IPv6 addresses based on the MAC address (EUI-64 mechanism). It defines how to create additional addresses (generally known as "temporary addresses") based on a random interface identifier for the purpose of initiating outgoing sessions. These "random" or temporary addresses are meant to be used for a short period of time (hours to days) and would then be deprecated. Deprecated addresses can continue to be used for already established connections, but are not used to initiate new connections. New temporary addresses are generated periodically to replace temporary addresses that expire. In order to do so, a node produces a sequence of temporary global scope addresses from a sequence of interface identifiers that appear to be random in the sense that it is difficult for an outside observer to predict a future address (or identifier) based on a current one, and it is difficult to determine previous addresses (or identifiers) knowing only the present one. The main problem with the temporary addresses is that they should not be used by applications that listen for incoming connections (as these are supposed to be waiting on permanent/well-known identifiers). Besides, if a node changes network and comes back to a previously visited one, the temporary addresses that the node would use will be different, and this might be an issue in certain networks where addresses are used for operational purposes (e.g., filtering or authentication). [RFC7217], summarized next, partially addresses the problems aforementioned.

[RFC7217] defines a method for generating IPv6 IIDs to be used with IPv6 Stateless Address Autoconfiguration (SLAAC), such that an IPv6 address configured using this method is stable within each subnet, but the corresponding IID changes when the host moves from one network to another. This method is meant to be an alternative to generating Interface Identifiers based on MAC addresses, such that the benefits of stable addresses can be achieved without sacrificing the security and privacy of users. The method defined to generate the IPv6 IID is based on computing a hash function which takes as input information that is stable and associated to the interface (e.g., MAC address or local interface identifier), stable information associated to the visited network (e.g., IEEE 802.11 SSID), the IPv6 prefix, and a secret key, plus some other additional information. This basically ensures that a different IID is generated when any of the input fields changes (such as the network or the prefix), but that the IID is the same within each subnet.

In addition to the former documents, [I-D.ietf-dhc-mac-assign] proposes an extension to DHCPv6 that allows a scalable approach to link-layer address assignments where preassigned link-layer address assignments (such as by a manufacturer) are not possible or unnecessary. [I-D.ietf-dhc-slap-quadrant] proposes extensions to DHCPv6 protocols to enable a DHCPv6 client or a DHCPv6 relay to

indicate a preferred SLAP quadrant to the server, so that the server may allocate MAC addresses in the quadrant requested by the relay or client.

Not only MAC and IP addresses can be used for tracking purposes. Some DHCP options carry unique identifiers. These identifiers can enable device tracking even if the device administrator takes care of randomizing other potential identifications like link-layer addresses or IPv6 addresses. [RFC7844] introduces anonymity profiles, designed for clients that wish to remain anonymous to the visited network. The profiles provide guidelines on the composition of DHCP or DHCPv6 messages, designed to minimize disclosure of identifying information. [RFC7844] also indicates that the link-layer address, IP address, and DHCP identifier shall evolve in synchrony.

Lately, the MAC Address Device Identification for Network and Application Services (MADINAS) IETF BoF has discussed the need to examine the effect of RCM schemes on network and application services in several scenarios identified as relevant.

7. OS current practices

Most modern OSes (especially mobile ones) do implement by default some MAC address randomization policy. Table 1 summarizes current practices for Android and iOS, as the time of writing this document (original source: <https://www.fing.com/news/private-mac-address-on-ios-14>, updated based on findings from the authors).

Android 10+	iOS 14+
The randomized MAC address is bound to the SSID	The randomized MAC address is bound to the BSSID
The randomized MAC address is stable across reconnections for the same network	The randomized MAC address is stable across reconnections for the same network
The randomized MAC address is reset when the device forgets a WiFi network	The randomized MAC address is reset when the device forgets a WiFi network
MAC address randomization is enabled by default for all the new WiFi networks. But if the device previously connected to a WiFi network identifying itself with the real MAC address, no randomized MAC address will be used (unless manually enabled)	MAC address randomization is enabled by default for all the new WiFi networks

Table 1: Android and iOS MAC address randomization practices

8. IANA Considerations

N/A.

9. Security Considerations

TBD.

10. Acknowledgments

TBD.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

11.2. Informative References

[enhancing_location_privacy]

Gruteser, M. and D. Grunwald, "Enhancing location privacy in wireless LAN through disposable interface identifiers: a quantitative analysis", *Mobile Networks and Applications*, vol. 10, no. 3, pp. 315-325 , 2005.

[I-D.gont-6man-deprecate-eui64-based-addresses]

Gont, F., Cooper, A., Thaler, D., and W. Liu, "Deprecating EUI-64 Based IPv6 Addresses", draft-gont-6man-deprecate-eui64-based-addresses-00 (work in progress), October 2013.

[I-D.ietf-dhc-mac-assign]

Volz, B., Mrugalski, T., and C. J. Bernardos, "Link-Layer Address Assignment Mechanism for DHCPv6", draft-ietf-dhc-mac-assign-09 (work in progress), September 2020.

[I-D.ietf-dhc-slap-quadrant]

Bernardos, C. J. and A. Mourad, "Structured Local Address Plan (SLAP) Quadrant Selection Option for DHCPv6", draft-ietf-dhc-slap-quadrant-12 (work in progress), October 2020.

[IEEE_802_11_aq]

Group, 8. W. -. W. L. W., "IEEE 802.11aq-2018 - IEEE Standard for Information technology--Telecommunications and information exchange between systems Local and metropolitan area networks--Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Preassociation Discovery", IEEE 802.11 , 2018.

[IEEE_802c]

architecture, 8. W. -. 8. L., "IEEE 802c-2017 - IEEE Standard for Local and Metropolitan Area Networks:Overview and Architecture--Amendment 2: Local Medium Access Control (MAC) Address Usage", IEEE 802c , 2017.

[IEEE_802E]

architecture, 8. W. -. 8. L., "IEEE 802E-2020 - IEEE Recommended Practice for Privacy Considerations for IEEE 802 Technologies", IEEE 802E , 2020.

[ieee_privacy_ecsg]

IEEE 802 Privacy EC SG, "IEEE 802 EC Privacy Recommendation Study Group",
<<http://www.ieee802.org/PrivRecsg/>>.

[link_layer_privacy]

O'Hanlon, P., Wright, J., and I. Brown, "Privacy at the link layer", Contribution at W3C/IAB workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT) , February 2014.

[privacy_android]

Google/Open Handset Alliance, "Android Privacy: MAC Randomization",
<<https://source.android.com/devices/tech/connect/wifi-mac-randomization>>.

[privacy_ios]

Apple, "Use private Wi-Fi addresses in iOS 14, iPadOS 14, and watchOS 7",
<<https://support.apple.com/en-us/HT211227>>.

[privacy_tutorial]

Cooper, A., Hardie, T., Zuniga, JC., Chen, L., and P. O'Hanlon, "Tutorial on Pervasive Surveillance of the Internet - Designing Privacy into Internet Protocols",
<<https://mentor.ieee.org/802-ec/dcn/14/ec-14-0043-01-00EC-internet-privacy-tutorial.pdf>>.

[privacy_windows]

Microsoft, "Windows: How to use random hardware addresses", <<https://support.microsoft.com/en-us/windows/how-to-use-random-hardware-addresses-ac58de34-35fc-31ff-c650-823fc48eblbc>>.

[rcm_privacy_csd]

SG, 8. W. R., "IEEE 802.11 Randomized And Changing MAC Addresses Study Group CSD on user experience mechanisms", doc.:IEEE 802.11-20/1346r1 , 2020.

[rcm_privacy_par]

SG, 8. W. R., "IEEE 802.11 Randomized And Changing MAC Addresses Study Group PAR on privacy mechanisms", doc.:IEEE 802.11-19/854r7 , 2020.

[rcm_tig_final_report]

TIG, 8. W. R., "IEEE 802.11 Randomized And Changing MAC Addresses Topic Interest Group Report", doc.:IEEE 802.11-19/1442r9 , 2019.

- [rcm_user_experience_csd]
SG, 8. W. R., "IEEE 802.11 Randomized And Changing MAC Addresses Study Group CSD on user experience mechanisms", doc.:IEEE 802.11-20/1117r3 , 2020.
- [rcm_user_experience_par]
SG, 8. W. R., "IEEE 802.11 Randomized And Changing MAC Addresses Study Group PAR on user experience mechanisms", doc.:IEEE 802.11-20/742r5 , 2020.
- [RFC4191] Draves, R. and D. Thaler, "Default Router Preferences and More-Specific Routes", RFC 4191, DOI 10.17487/RFC4191, November 2005, <<https://www.rfc-editor.org/info/rfc4191>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<https://www.rfc-editor.org/info/rfc6973>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<https://www.rfc-editor.org/info/rfc7844>>.
- [strint] W3C/IAB, "A W3C/IAB workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)", <<https://www.w3.org/2014/strint/>>.
- [wba_paper]
Alliance, W. B., "Wi-Fi Identification Scope for Liasing - In a post MAC Randomization Era", doc.:WBA Wi-Fi ID Intro: Post MAC Randomization Era v1.0 - IETF liaison , March 2020.

`[when_mac_randomization_fails]`

Martin, J., Mayberry, T., Donahue, C., Foppe, L., Brown, L., Riggins, C., Rye, E., and D. Brown, "A Study of MAC Address Randomization in Mobile Devices and When it Fails", arXiv:1703.02874v2 [cs.CR] , 2017.

`[wifi_internet_privacy]`

Bernardos, CJ., Zuniga, JC., and P. O'Hanlon, "Wi-Fi Internet Connectivity and Privacy: Hiding your tracks on the wireless Internet", Standards for Communications and Networking (CSCN), 2015 IEEE Conference on , October 2015.

`[wifi_tracking]`

The Independent, "London's bins are tracking your smartphone", <<https://www.independent.co.uk/life-style/gadgets-and-tech/news/updated-london-s-bins-are-tracking-your-smartphone-8754924.html>>.

Authors' Addresses

Juan Carlos Zuniga
SIGFOX
Montreal QC
Canada

Email: j.c.zuniga@ieee.org

Carlos J. Bernardos
Universidad Carlos III de Madrid
Av. Universidad, 30
Leganes, Madrid 28911
Spain

Phone: +34 91624 6236
Email: cjbc@it.uc3m.es
URI: <http://www.it.uc3m.es/cjbc/>

Amelia Andersdotter
CENTR
Belliardstraat 20 (6th floor)
Brussels 1040
Belgium

Email: amelia@centr.org
URI: <https://www.centr.org>