

BIER
Internet-Draft
Intended status: Experimental
Expires: 13 August 2022

T. Eckert
Futurewei Technologies USA
B. Xu
Huawei Technologies (2012Lab)
9 February 2022

Carrier Grade Minimalist Multicast (CGM2) using Bit Index Explicit
Replication (BIER) with Recursive BitString Structure (RBS) Addresses
draft-eckert-bier-cgm2-rbs-01

Abstract

This memo introduces the architecture of a multicast architecture derived from BIER-TE, which this memo calls Carrier Grade Minimalist Multicast (CGM2). It reduces limitations and complexities of BIER-TE by replacing the representation of the in-packet-header delivery tree of packets through a "flat" BitString of adjacencies with a hierarchical structure of BFR-local BitStrings called the Recursive BitString Structure (RBS) Address.

Benefits of CGM2 with RBS addresses include smaller/fewer BIFT in BFR, less complexity for the network architect and in the CGM2 controller (compared to a BIER-TE controller) and fewer packet copies to reach a larger set of BFER.

The additional cost of forwarding with RBS addresses is a slightly more complex processing of the RBS address in BFR compared to a flat BitString and the novel per-hop rewrite of the RBS address as opposed to bit-reset rewrite in BIER/BIER-TE.

CGM2 can support the traditional deployment model of BIER/BIER-TE with the BIER/BIER-TE domain terminating at service provider PE routers as BFIR/BFER, but it is also the intention of this document to expand CGM2 domains all the way into hosts, and therefore eliminating the need for an IP Multicast flow overlay, further reducing the complexity of Multicast services using CGM2. Note that this is not fully detailed in this version of the document.

This document does not specify an encapsulation for CGM2/RBS addresses. It could use existing encapsulations such as [RFC8296], but also other encapsulations such as IPv6 extension headers.

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 13 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. Overview	3
1.1. Introduction	3
1.2. Encapsulation Considerations	4
2. CGM2/RBS Architecture	5
3. CGM2/RBS forwarding plane	6
3.1. RBS BIFT	7
3.2. Reference encoding of RBS addresses	8
3.3. RBS Address	8
3.3.1. RecursiveUnit	8
3.3.2. AddressingField	9
4. BIER-RBS Example	9
4.1. BFR B	10
4.2. BFR R	12
4.3. BFR S	13
4.4. BFR C	14
4.5. BFR D	14
4.6. BFR E	15
5. RBS forwarding Pseudocode	16
6. Operational and design considerations (informational)	18
6.1. Comparison with BIER-TE / BIER	18

6.1.1.	Eliminating the need for large BIFT	18
6.1.2.	Reducing number of duplicate packet copies across BFR	19
6.1.3.	BIER-TE forwarding plane complexities	20
6.1.4.	BIER-TE controller complexities	20
6.1.5.	BIER-TE specification complexities	20
6.1.6.	Forwarding plane complexity	21
6.2.	CGM2 / RBS controller considerations	21
6.3.	Analysis of performance gain with CGM2	21
6.3.1.	Reference topology	21
6.3.2.	Comparison BIER and CGM2/RBS	23
6.4.	Example use case scenarios	24
7.	Acknowledgements	24
8.	Security considerations	24
9.	Changelog	24
10.	References	24
10.1.	Normative References	24
10.2.	Informative References	25
	Authors' Addresses	25

1. Overview

1.1. Introduction

Carrier Grade Minimalist Multicast (CGM2) is an architecture derived from the BIER-TE architecture [I-D.ietf-bier-te-arch] with the following changes/improvements.

CGM2 forwarding is based on the principles of BIER-TE forwarding: It is based on an explicit, in-packet, "source routed" tree indicated through bits for each adjacency that the packet has to traverse. Like in BIER-TE, adjacencies can be L2 to a subnet local neighbor in support of "native" deployment of CGM2 and/or L3, so-called "routed" adjacencies to support incremental or partial deployment of CGM2 as needed.

The address used to replicate packets in the network is not a flat network wide BitString as in BIER-TE, but a hierarchical structure of BitStrings called a Recursive BitString Structure (RBS) Address. The significance of the BitPositions (BP) in each BitString is only local to the BIFT of the router/BFR that is processing this specific BitString.

RBS addressing allows for a more compact representation of a large set of adjacencies especially in the common case of sparse set of receivers in large Service Provider Networks (SP).

CGM2 thereby eliminates the challenges in BIER [RFC8279] and BIER-TE having to send multiple copies of the same packet in large SP networks and the complexities especially for BIER-TE (but also BIER) to engineer multiple set identifier (SI) and/or sub-domains (SD) BIER-TE topologies for limited size BitStrings (e.g.: 265) to cover large network topologies.

Like BIER-TE, CGM2 is intended to leverage a Controller to minimize the control plane complexity in the network to only a simple unicast routing underlay required only for routed adjacencies.

The controller centric architecture provides most easily any type of required traffic optimization for its multicast traffic due to their need to perform often NP-complete calculations across the whole topology: reservation of bandwidth to support CIR/PIR traffic buffer/latency to support Deterministic Network (DetNet) traffic, cost optimized Steiner trees, failure point disjoint trees for higher resilience including DetNet deterministic services.

CGM2 can be deployed as BIER/BIER-TE are specified today, by encapsulating IP Multicast traffic at Provider Edge (PE) routers, but it is also considered to be highly desirable to extend CGM2 all the way into Multicast Sender/Receivers to eliminate the overhead of an Overlay Control plane for that (legacy) IP Multicast layer and the need to deal with yet another IP multicast group addressing space. In this deployment option Controller signaling extends directly (or indirectly via BFIR) into senders.

1.2. Encapsulation Considerations

This document does not define a specific BIER-RBS encapsulation nor does it preclude that multiple different encapsulations may be beneficial to better support different use-cases or operator/user technology preferences. Instead, it discusses considerations for specific choices.

BIER-RBS can easily re-use [RFC8296] encapsulation. The RBS address is inserted into the [RFC8296] BitString field. The BFR forwarding plane needs to be configured (from Controller or control plane) that the BIFT-id(s) used with RBS addresses are mapped to BIFT and forwarding rules with RBS semantic.

SI/SD fields of [RFC8296] may be used as in BIER-TE, but given that CGM2 is designed (as described in the Overview section) to simplify multicast services, a likely and desirable configuration would be to only use a single BIFT in each BFR for RBS addresses, and mapping these to a single SD and SI 0.

IP Multicast [RFC1112] was defined as an extension of IP [RFC791], reusing the same network header, and IPv6 multicast inherits the same approach. In comparison, [RFC8296] defines BIER encapsulation as a completely separate (from IP) layer 3 protocol, and duplicates both IP and MPLS header elements into the [RFC8296] header. This not only results in always unused, duplicate header parameters (such as TC vs. DSCP), but it also foregoes the option to use any non-considered IPv6 extension headers with BIER and would require the introduction of a whole new BIER specific socket API into host operating systems if it was to be supported natively in hosts.

Therefore an encapsulation of RBS addresses using an IP and/or IPv6 extension header may be more desirable in otherwise IP and/or IPv6 only deployments, for example when CGM2 is extended into hosts, because it would allow to support CGM2 via existing IP/IPv6 socket APIs as long as they support extension headers, which the most important host stacks do today.

2. CGM2/RBS Architecture

This section describes the basic CGM2 architecture via Figure 1 through its key differences over the BIER-TE architecture.

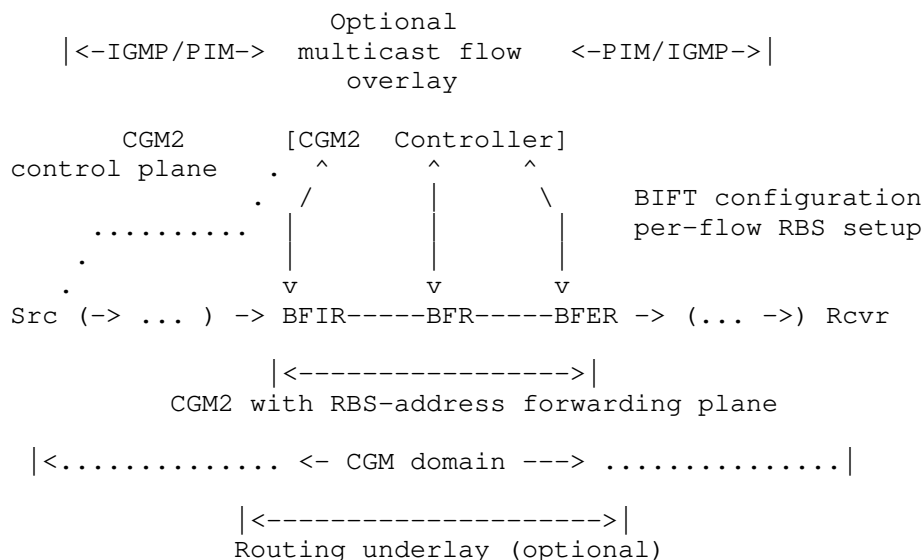


Figure 1: CGM2/RBS Architecture

In the "traditional" option, when deployed with a domain spanning from BFIR to BFER, the CGM2 architecture is very much like the BIER-TE architecture, in which the BIER-TE forwarding rules for (BitString,SI,SD) addresses are replaced by the RBS address forwarding rules.

The CGM2 Controller replaces the BIER-TE controller, populating during network configuration the BIFT, which are very much like BIER-TE BIFT, except that they do not cover a network-wide BP address space, but instead each BFR BIFT only needs as many BP in its BIFT as it has link-local adjacencies, and in partial deployments also additional L3 adjacencies to tunnel across non-CGM capable routers.

Per-flow operations in this "traditional" option is very much as in BIER/BIER-TE, with the CGM2 controller determining the RBS address (instead of the BIER-TE (BitString,SI,SD)) to be imposed as part of the RBS address header (compared to the BIER encapsulation [RFC8296]) on the BFIR.

To eliminate the need for an IP Multicast flow overlays, a CGM2 domain may extend all the way into Sender/Receiver hosts. This is called "end-to-end" deployment model. In that case, the sender host and CGM2 controller collaborate to determine the desired receivers for a packet as well as desired path policy/requirements, the controller indicates to the sender of the packet the necessary RBS address and address of the BFIR, and the Sender imposes an appropriate RBS address header together with a unicast encapsulation towards the BFIR.

CGM2 is also intended so especially simplify controller operations that also instantiate QoS policies for multicast traffic flows, such as bandwidth and latency reservations (e.g.: DetNet). As in BIER-TE, this is orthogonal to the operations of the CGM2/RBS address forwarding operations and will be covered in separate documents.

3. CGM2/RBS forwarding plane

Instead of a (flat) BitString as in BIER-TE that use a network wide shared BP address space for adjacencies across multiple BFR, CGM2 uses a structured address built from so-called RecursiveUnits (RU) that contain BitStrings, each of which is to be parsed by exactly one BFR along the delivery tree of the packet.

The equivalent to a BIER/BIER-TE BitString is therefore called the RecursiveUnit BitString Structure (RBS) Address. Forwarding for CGMP2 is therefore also called RBS forwarding.

3.1. RBS BIFT

RBS BIFT as shown in Figure 2 are, like BIER-TE BIFT, tables that are indexed by BP, containing for each BP an adjacency. The core difference over BIER-TE BIFT is that the BP of the BIFT are all local to the BFR, whereas in BIER-TE, the BP are shared across a BIER-TE domain, each BFR can only use a subset the BP for its own adjacencies, and only in some cases can BP be shared for adjacencies across two (or more) BFR. Because of this difference, most of the complexities of BIER-TE BIFT are not required with BIER-RBS BIFT, see Section 6.1.3.

BP	Recursive	Adjacency
1	1	adjacent BFR
2	0	punt/host
.....	...	
N

Figure 2: RBS BIFT

An RBS BIFT has a configured number of N addressable BP entries. When a BFR receives a packet with an RBS address, it expects that the BitString inside the RBS address that needs to be parsed by the BFR (see Section 3.3 has a length that matches N according to the encapsulation used for the RBS address. Therefore, N MUST support configuration in increments of the supported size of the BitString in the encapsulation of the RBS Address. In the reference encoding (see Section 3.3), the increment for N is 1 (bit). If an encapsulation would call for a byte accurate encoding of the BitString, N would have to be configurable in increments of 8.

BFR MUST support a value of N larger than the maximum number of adjacencies through which RBS forwarding/replication of a single packet is required, such as the number of physical interfaces on BFR that are intended to be deployed as a Provider Core (P) routers.

RBS BIFT introduce a new "Recursive" flag for each BP. These are used for adjacencies to other BFR to indicate that the BFR processing the packet RBS address BitString also has to expect for every BP with the recursive flag set another RU inside the RBS address.

3.2. Reference encoding of RBS addresses

Structure elements of the RBS Address and its components are parameterized according to a specific encapsulation for RBS addresses, such as the total size of the TotalLen field and the unit in which it is counted (see Section 3.3). These parameters are outside the scope of this document. Instead, this document defines example parameters that together form the so called "Reference encoding of RBS addresses". This encoding may or may not be adopted for any particular encapsulation of RBS addresses.

3.3. RBS Address

An RBS address is structured as shown in Figure 3.

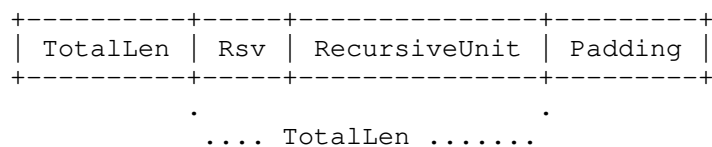


Figure 3: RBS Address

TotalLen counts in some unit, such as bits, nibbles or bytes the length of the RBS Address excluding itself and Padding. For the reference encoding, TotalLen is an 8-bit field that counts the size of the RBS address in bits, permitting for up to 256 bit long RBS addresses.

In case additional, non-recursive flags/fields are determined to be required in the RBS Address, they should be encoded in a field between TotalLen and RecursiveUnit, which is called Rsv. In the reference encoding, this field has a length of 0.

Padding is used to align the RBS address as required by the encapsulation. In the reference encoding, this alignment is to 8 bits (byte boundaries). Therefore, $\text{Padding (bits)} = (8 - \text{TotalLen} \% 8)$.

3.3.1. RecursiveUnit

The RecursiveUnit field is structured as shown in Figure 4.

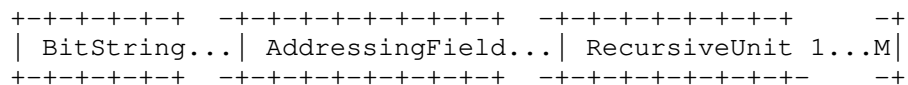


Figure 4: RBS RecursiveUnit

The BitString field indicates the bit positions (BPs) to which the packet is to be replicated using the BIFT of the BFR that is processing the Recursive unit.

For each of M BP set in the BitString of the RecursiveUnit for which the Recursive flag is set in the BIFT of the BFR, the RecursiveUnit contains a RecursiveUnit i , $i=1..M$, in order of increasing BP index.

If adjacencies between BFR are not configured as recursive in the BIFT, this recursive extraction does not happen for an adjacency, no RecursiveUnit i has to be encoded for the BP, and BFRs across such adjacencies would have to share the BP of a common BIFT as in BIER-TE. This option is not further discussed in this version of the document.

3.3.2. AddressingField

The AddressingField of an RBS address is structured as shown in Figure 5.

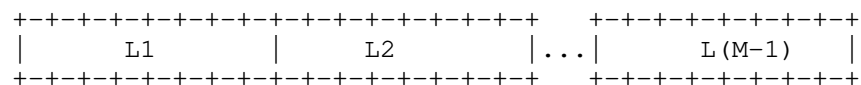


Figure 5: RBS AddressingField

The AddressingField consists of one or more fields L_i , $i=1 \dots (M-1)$. L_i is the length of RecursiveUnit i for the i 'th recursive bit set in the BitString preceding it.

In the reference encoding, the lengths are 8-bit fields indicating the length of RecursiveUnits in bits.

The length of the M'th RecursiveUnit is not explicitly encoded but has to be calculated from TotalLen.

4. BIER-RBS Example

Figure 6 shows an example for RBS forwarding.

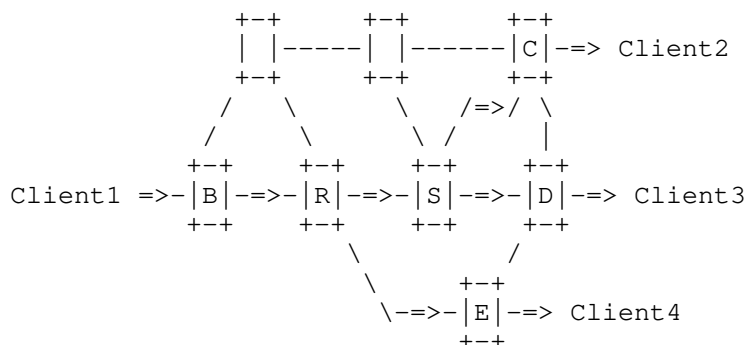


Figure 6: Example Network Topology

A packet from Client1 connected to BFIR B is intended to be replicated to Client2,3,4. The example initially assumes the traditional option of the architecture, in which the imposition of the header for the RBS address happens on BFIR B, for example based on functions of an IP multicast flow overlay.

A controller determines that the packet should be forwarded hop-by-hop across the network as shown in Figure 7.

```

Client 1 ->B(impose BIER-RBS)
      =>R(
        => E (dispose BIER-RBS)
          => Client4
        => S(
          =>C (dispose BIER-RBS)
            => Client2
          =>D (dispose BIER-RBS)
            => Client3
          )
        )
  
```

Figure 7: Desired example forwarding tree

4.1. BFR B

The 34 bit long (without padding) RBS address shown in Figure 8 is constructed to represent the desired tree from Figure 7 and is imposed at B onto the packet through an appropriate header supporting the reference encoding of RBS addresses.

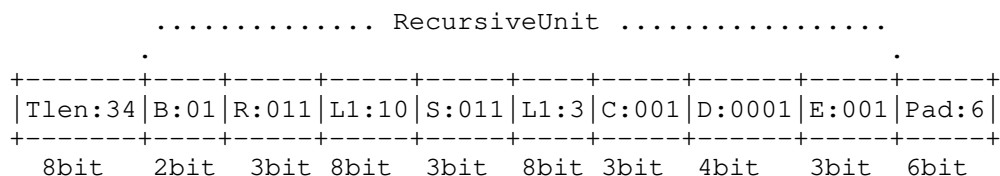


Figure 8: RBS Address imposed at BFIR-B

In Figure 8 and further the illustrations of RBS addresses, BitStrings are preceded by the name of the BFR for whom they are destined and their values are shown as binary with the lowest BP 1 starting on the left. TotalLength (Tlen:), AddressingField (L1:) and Padding (Pad:) fields are shown with decimal values.

RBS forwarding on B examines this address based on its RBS BIFT with N=2 BP entries, which is shown in Figure 9.

BP	Recursive	Adjacency
1	0	client1
2	1	R

Figure 9: BIER-RBS BIFT on B

This results in the parsing of the RBS address as shown in Figure 10, which shows that B does not need (nor can) parse all structural elements, but only those relevant to its own RBS forwarding procedure.

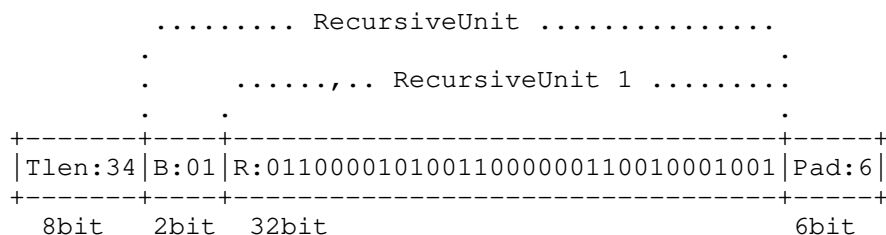


Figure 10: RBS Address as processed by BFIR-B

There is only one BP towards BFR R set in the BitString B:01, so the RecursiveUnit 1 follows directly after the end of the BitString B:01 and it covers the whole Tlen - length of BitString (34 - 2 = 32 bit).

B rewrites the RBS address by replacing the RecursiveUnit with RecursiveUnit 1 and adjusts the Padding to zero bits. The resulting RBS address is shown in Figure 11. It then sends the packet copy with that rewritten RBS address to BFR R.

4.2. BFR R

BFR R receives from BFR B the packet with that RBS address shown in Figure 11.

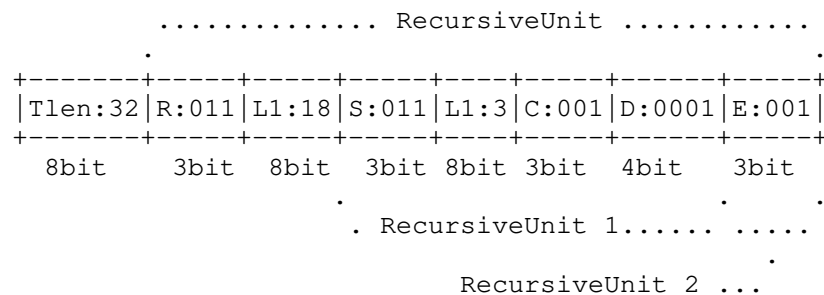


Figure 11: RBS Address processed by BFR-R

BFR R parses the RBS Address as shown in Figure 12 using its RBS BIFT of N=3 BP entries shown in Figure 13.

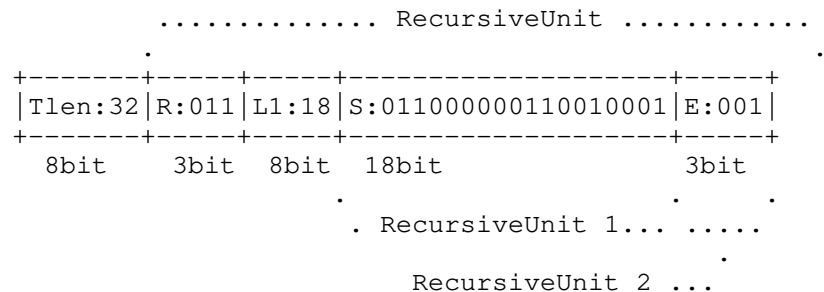


Figure 12: RBS Address processed by BFR-R

Because there are two recursive BP set in the BitString for R, one for BFR S and one for BFR E, one Length field L1 is required in the AddressingField, indicating the length of the RecursiveUnit 1 for BFR S, followed by the remainder of the RBS address being the RecursiveUnit 2 for BFR E.

BP	Recursive	Adjacency
1	1	B
2	1	S
3	1	E

Figure 13: RBS BIFT on BFR R

BFR R accordingly creates one copy for BFR S using RecursiveUnit 1, and only copy for BFR E using RecursiveUnit 2, updating Padding accordingly for each copy.

4.3. BFR S

BFR S receives from BFR B the packet and parses the RBS address as shown in Figure 14 using its RBS BIFT of N=3 BP shown in Figure 15.

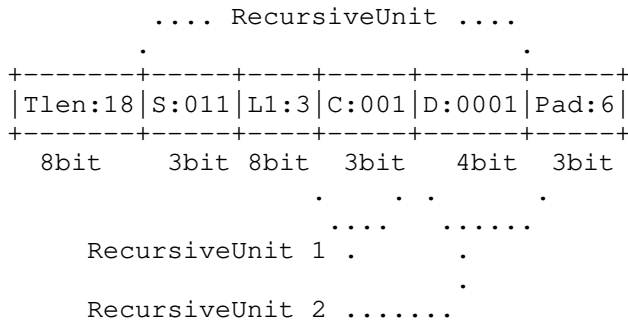


Figure 14: RBS Address processed by BFR-S

BP	Recursive	Adjacency
1	1	R
2	1	C
3	1	D

Figure 15: RBS BIFT on BFR-S

BFR S accordingly sends one packet copy with RecursiveUnit 1 in the RBS address to BFR C and a second packet copy with RecursiveUnit 2 to BFR D.

4.4. BFR C

BFR C receives from BFR S the packet and parses the RBS address according to its N=3 BP entries BIFT (shown in Figure 17) as shown in Figure 16.

```

+-----+-----+-----+
|Tlen:3 |C:001|Pad:5|
+-----+-----+-----+
   8bit    3bit 5bit

```

Figure 16: RBS Address processed by BFR-C

BP	Recursive	Adjacency
1	1	S
2	1	D
3	0	local_decap

Figure 17: RBS BIFT on BFR-C

BFR S accordingly creates one packet copy for BP 3 where the RBS address encapsulation is disposed of, and the packet is ultimately forwarded to Client 2, for example because of an IP multicast payload for which the multicast flow overlay identifies Client 2 as an interested receiver, as in BIER/BIER-TE.

To avoid having to use an IP flow overlay, the BIFT could instead have one BP allocated for every non-RBS destination, in this example BP 3 would then explicitly be allocated for Client 2, and instead of disposing of the RBS address encapsulation, BFR C would impose or rewrite a unicast encapsulation to make the packet become a unicast packet directed to Client 2. This option is not further detailed in this version of the document.

4.5. BFR D

The procedures for processing of the packet on BFR D are very much the same as on BFR C. Figure 18 shows the RBS address at BFR D, Figure 19 shows the N=4 bit RBS BIFT of BFR D.

Tlen:4	D:0001	Pad:4
8bit	4bit	4bit

Figure 18: RBS Address processed by BFR-D

BP	Recursive	Adjacency
1	1	S
2	1	C
3	1	E
4	0	local_decap

Figure 19: RBS BIFT on BFR-D

4.6. BFR E

The procedures for processing of the packet on BFR E are very much the same as on BFR C and D. Figure 20 shows the RBS address at BFR D, Figure 21 shows the N=E bit RBS BIFT of BFR E.

Tlen:3	E:001	Pad:5
8bit	3bit	5bit

Figure 20: RBS Address processed by BFR-E

BP	Recursive	Adjacency
1	1	R
2	1	D
3	0	local_decap

Figure 21: RBS BIFT on BFR-E

5. RBS forwarding Pseudocode

The following example RBS forwarding Pseudocode assumes the reference encoding of bit-accurate length of BitStrings and RecursiveUnits as well as 8-bit long TotalLen and AddressingField Lengths. All packet field addressing and address/offset calculations is therefore bit-accurate instead of byte accurate (which is what most CPU memory access today is).


```

void ForwardRBSPacket (Packet)
{
    RBS = GetPacketMulticastAddr(Packet);
    Total_len = RBS;
    Rsv = Total_len + length(Total_Len);
    BitStringA = Rsv + length(Rsv);
    AddressingField = BitStringA + BIFT.entries;

    // [1] calculate number of recursive bits set in BitString
    CopyBitString(*BitStringA, *RecursiveBits, BIFT.entries);
    And(*RecursiveBits, *BIFTRecursiveBits, BIFT.entries);
    N = CountBits(*RecursiveBits, BIFT.entries);

    // Start of first RecursiveUnit in RBS address
    // After AddressingField array with 8-bit length fields
    RecursiveUnit = AddressingField + (N - 1) * 8;

    RemainLength = *Total_len - length(Rsv)
                  - BIFT.entries;

    Index = GetFirstBitPosition(*BitStringA);
    while (Index) {
        PacketCopy = Copy(Packet);

        if (BIFT.BP[Index].recursive) {
            if(N == 1) {
                RecursiveUnitLength = RemainLength;
            } else {
                RecursiveUnitLength = *AddressingField;
                N--;
                AddressingField += 8;
                RemainLength -= RecursiveUnitLength;
                RemainLength -= 8; // 8 bit of AddressingField
            }
            RewriteRBS(PacketCopy, RecursiveUnit, RecursiveUnitLength);
            SendTo(PacketCopy, BIFT.BP[Index].adjacency);

            RecursiveUnit += RecursiveUnitLength;
        } else {
            DisposeRBSHeader(PacketCopy);
            SendTo(PacketCopy, BIFT.BP[Index].adjacency);
        }
        Index = GetNextBitPosition(*BitStringA, Index);
    }
}

```

Figure 22: RBS address forwarding Pseudocode

Explanations for Figure 22.

RBS is the (bit accurate) address of the RBS address in packet header memory. BitStringA is the address of the RBS address BitString in memory. length(Total_Len) and length(Rsv) are the bit length of the two RBS address fields, e.g.: 8 bit and 0 bit for the reference encoding.

The BFR local BIFT has a total number of BIFT.entries addressable BP 1...BIFTentries. The BitString therefore has BIFT.entries bits.

BIFT.RecursiveBits is a BitString pre-filled by the control plane with all the BP with the recursive flag set. This is constructed from the Recursive flag setting of the BP of the BIFT. The code starting at [1] therefore counts the number of recursive BP in the packets BitString.

Because the AddressingField does not have an entry for the last (or only) RecursiveUnit, its length has to be calculated by taking TotalLen into account.

RewriteRBS needs to replace RBS address with the RecursiveUnit address, keeping only Rsv, recalculating TotalLen and adding appropriate Padding.

For non-recursive BP, the Pseudocode assumes disposition of the RBSheader. This is not strictly necessary but non-disposing cases are outside of scope of this version of the document.

6. Operational and design considerations (informational)

6.1. Comparison with BIER-TE / BIER

This section discusses informationally, how and where CGM2 can avoid different complexities of BIER/BIER-TE, and where it introduces new complexities.

6.1.1. Eliminating the need for large BIFT

In a BIER domain with M BFER, every BFR requires M BIFT entries. If the supported BSL is N and $M > 2^N$, then $S = (M / 2^N)$ set indices (SI) are required, and S copies of the packet have to be sent by the BFIR to reach all targeted BFER.

In CGM2, the number of BIFT entries does not need to scale with the number of BFER or paths through the network, but can be limited to only the number of L2 adjacencies of the BFR. Therefore CGM2 requires minimum state maintenance on each BFR, and multiple SI are not required.

6.1.2. Reducing number of duplicate packet copies across BFR

If the total size of an RBS encoded delivery tree is larger than a supported maximum RBS header size, then the CGM2 controller simply needs to divide the tree into multiple subtrees, each only addressing a part of the BFER (leaves) of the target tree and pruning any unnecessary branches.

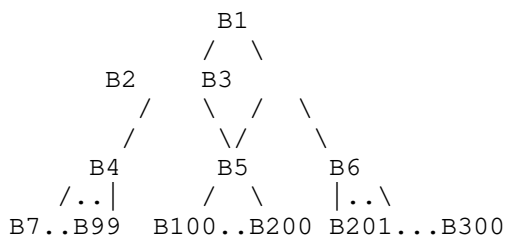


Figure 23: Simple Topology Example

Consider the simple topology in Figure 23 and a multicast packet that needs to reach all BFER B7...B300. Assume that the desired maximum RBM header size is such that a RBS address size of ≤ 256 bits is desired. The CGM2 controller could create an RBS address $B1 \Rightarrow B2 \Rightarrow B4 \Rightarrow (B7..B99)$, for a first packet, an RBS address $B1 \Rightarrow B3 \Rightarrow B5 \Rightarrow (B100..B200)$ for a second packet and a third RBS address $B1 \Rightarrow B3 \Rightarrow B6 \Rightarrow B201...B300$.

The elimination of larger BIFT state in BFR through multiple SI in BIER/BIER-TE does come at the expense of replicating initial hops of a tree in RBS addresses, such as in the example the encoding of $B1 \Rightarrow B3$ in the example.

Consider that the assignment of BFIR-ids with BIER in the above example is not carefully engineered. It is then easily possible that the BFR-ids for B7..B99 are not sequentially, but split over a larger BFIR-id space. If the same is true for all BFER, then it is possible that each of the three BFR B4, B5 and B6 has attached BFER from three different SI and one may need to send for example three multiple packets to B7 to address all BFER B7..B99 or to B5 to address all B100..B200 or B6 to address all B201...B300. These unnecessary duplicate packets across B4, B5 or B6 are because of the addressing principle in BIER and are not necessary in CGM2, as long as the total length of an RBS address does not require it.

For more analysis, see Section 6.3.

6.1.3. BIER-TE forwarding plane complexities

BIER-TE introduces forwarding plane complexities to allow reducing the BSL required. While all of these could be supported / implemented with CGM2, this document contends that they are not necessary, therefore providing significant overall simplifications.

- * BIER-TE supports multiple adjacencies in a single BIFT Index to allow compressing multiple adjacencies into a single Index for traffic that is known to always require replications to all those adjacencies (such as when flooding TV traffic).
- * BIER-TE support ECMP adjacencies which have to calculate which out of 2 or more possible adjacencies a packet should be forwarded to.
- * BIER-TE supports special Do-Not-Clear (DNC) behavior of adjacencies to permit reuse of such a bit for adjacencies on multiple consecutive BFR. This behavior specifically also raises the risk of looping packets.

6.1.4. BIER-TE controller complexities

BIER-TE introduces BIER-TE controller plane mechanisms that allow to reuse bits of the flat BIER-TE BitStrings across multiple BFR solely to reduce the number of BP required but without introducing additional complexities for the BIER-TE forwarding plane.

- * Shared BP for all Leaf BFR.
- * Shared BP for both Interfaces of p2p links.
- * Shared bits for multi-access subnets (LANs).

These bit-sharing mechanisms are unnecessary and inapplicable to CGM2 because there is no need to share BP across the BIFT of multiple BFR.

6.1.5. BIER-TE specification complexities

The BIER-TE specification distinguishes between forward (link scope) and routed (underlay routed) adjacencies to highlight, explain and emphasize on the ability of BIER-TE to be deployed in an overlay fashion especially also to reduce the necessary BSL, even when all routers in the domain could or do support BIER-TE.

In CGM2, routed adjacencies are considered to be only required in partial deployments to forward across non-CGM2 enabled routers. This specification does therefore not highlight link scope vs. routed adjacencies as core distinct features.

6.1.6. Forwarding plane complexity

CGM2 introduces some more processing calculation steps to extract the BitString that needs to be examined by a BFR from the RBS address. These additional steps are considered to be non-problematic for todays programmable forwarding planes such as P4.

Whereas BIER-TE clears bit on each hops processing, CGM2 rewrites the address on every hop by extracting the recursive unit for the next hop and make it become the packet copies address. This rewrite shortens the RBS address. This hopefully has only the same complexity as (tunnel) encapsulations/decapsulations in existing forwarding planes.

6.2. CGM2 / RBS controller considerations

TBD. Any aspects not covered in Section 6.1.

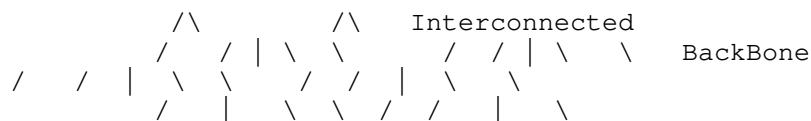
6.3. Analysis of performance gain with CGM2

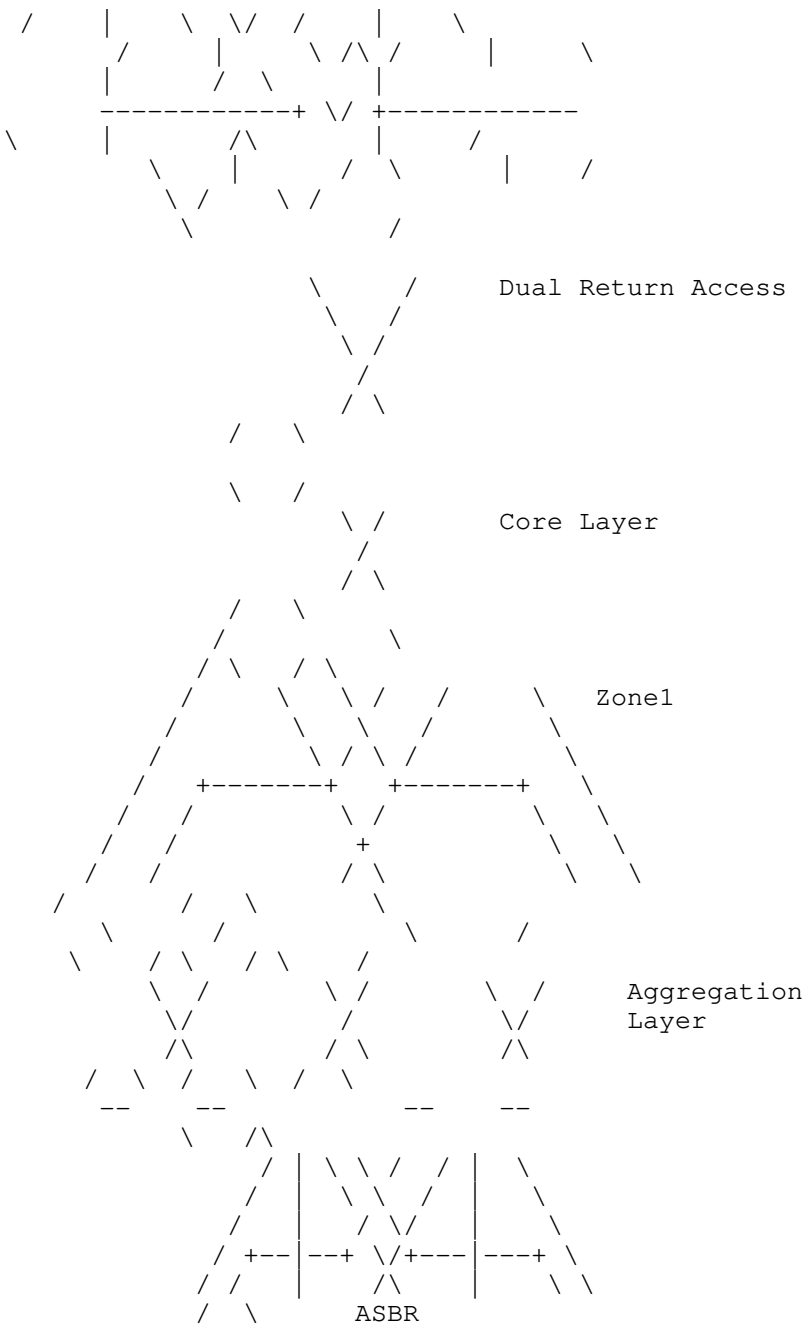
TBD: Comparison of number of packets/header sizes required in large real-world operator topology between BIER/BIER-TE and CGM2.
Analysis: Gain in dense topology.

6.3.1. Reference topology

Reference topology description:

1. Typical topology of Beijing Mobile in China.
2. All zones dual homing access to backbone.
3. Core layer: 4 nodes full mesh connected
4. Aggregation layer: 8 nodes are divided into two layers, with full interconnection between the layers and dual homing access to the core layer on the upper layer.
5. Aggregation rings: 8 rings, 6 nodes per ring
6. Access rings: 3600 nodes, 18 nodes per ring.





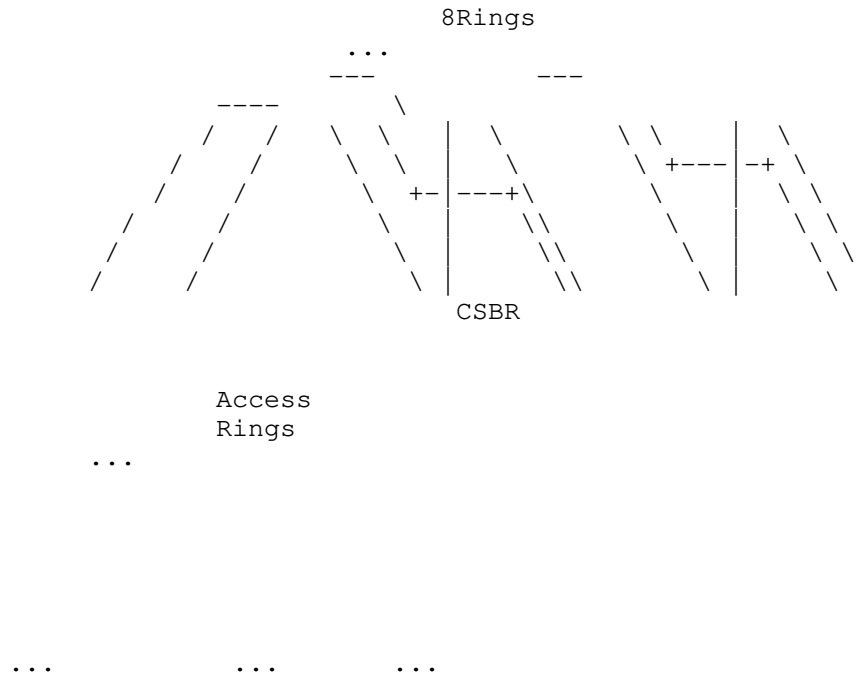


Figure 24: Reference Topology

6.3.2. Comparison BIER and CGM2/RBS

The following performance comparison is based on Figure 24.

1. CGM2: We randomly select egress points as group members, with the total number ranging from 10 to 28800 (for sake of simplicity, we assume merely one client per egress point). The egress points are randomly distributed in the topology with 10 runs for each value, showing the average result in our graphs. The total number of samples is 60
2. BIER: We divide the overall topology into 160 BIER domains, each of which includes 180 egress points, providing the total of 28000 egress points.

3. Simulation: In order to compare the BIER against the in-packet tree encoding mechanism, we limit the size of the header to 256 bits (the typical size of a BIER header).

Conclusion: 1. BIER reaches its 160 packet replication limit at about 500 users, while the in-packet tree encoding reaching its limit of 125 replications at about 12000 users. And the following decrease of replications is caused by the use of node-local broadcast as a further optimization. 2. For the sake of comparison, the same 256-bit encapsulation limit is imposed on CGM2, but we can completely break the 256-bit encapsulation limit, thus allowing the source to send fewer multicast streams. 3. CCGM2 encoding performs significantly better than BIER in that it requires less packet replications and there network bandwidth.

6.4. Example use case scenarios

TBD.

7. Acknowledgements

This work is based on the design published by Sheng Jiang, Xu Bing, Yan Shen, Meng Rui, Wan Junjie and Wang Chuang {jiangsheng|bing.xu|yashen|mengrui|wanjunjie2|wangchuang}@huawei.com, see [CGM2Design].

8. Security considerations

TBD.

9. Changelog

[RFC-Editor: please remove this section].

This document is written in <https://github.com/cabo/kramdown-rfc2629> markup language. This documents source is maintained at <https://github.com/toerless/bier-cgm2-rbs>, please provide feedback to the appropriate IETF mailing list and submit an Issue to the GitHub.

01 - Added section 6.3 about performance comparison and co-author (Robin).

00 - Initial version from [CGM2Design].

10. References

10.1. Normative References

[I-D.ietf-bier-te-arch]

Eckert, T., Menth, M., and G. Cauchie, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, draft-ietf-bier-te-arch-12, 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-arch-12.txt>>.

[RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <<https://www.rfc-editor.org/info/rfc791>>.

[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>>.

10.2. Informative References

[CGM2Design]

Jiang, S., Xu, B.(., Shen, Y., Rui, M., Junjie, W., and W. Chuang, "Novel Multicast Protocol Proposal Introduction", 10 October 2021, <<https://github.com/BingXu1112/CGMM/blob/main/Novel%20Multicast%20Protocol%20Proposal%20Introduction.pptx>>.

Authors' Addresses

Toerless Eckert
Futurewei Technologies USA
2220 Central Expressway
Santa Clara, CA 95050
United States of America

Email: tte@cs.fau.de

Bing (Robin) Xu
Huawei Technologies (2012Lab)
China

Email: bing.xu@huawei.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 SN. 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 NVO3 (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

Internet Area Working Group
Internet-Draft
Intended status: Experimental
Expires: October 16, 2022

Z. Chen
Huawei
S. Jiang
April 14, 2022

Native Minimal Protocols with Flexibility at Edge Networks
draft-jiang-intarea-nmp-edge-01

Abstract

This document introduces a flexible native minimal protocol for fast short packet transmission in edge networks, and can communicate with IPv6 nodes through gateways.

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 October 16, 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 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. Requirements Notation	3
3. Overview	3
4. Protocol Design	4
4.1. Packet Header Format	4
4.1.1. Data Packet Header Format	5
4.1.2. Control Packet Format	6
4.2. Control Messages	6
4.2.1. Address Request and Assignment Messages	6
4.2.2. Address Lease Extension Messages	7
4.3. DNS Delegation Messages	8
4.4. Functionalities of Gateway	9
4.4.1. Address Management	9
4.4.2. Address Translation	10
5. Renumber Considerations	10
6. Security Considerations	10
7. IANA Considerations	10
8. Acknowledgments	10
9. Normative References	10
Authors' Addresses	11

1. Introduction

TCP/IP protocol suites are adopted widely in different areas. However, there are still numerous edge networks uses non-IP technologies like ZigBee, BLE, CAN-bus, and Modbus for different reasons (e.g., power-constrained devices, low transport rate media). For such networks, application-layer gateways (or protocol translators) are usually deployed to connect them with the Internet, as shown in Figure 1.

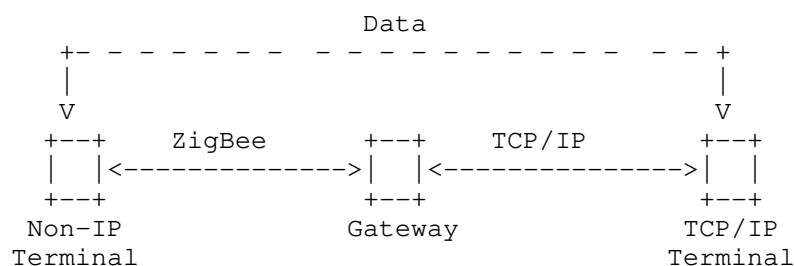


Figure 1: Communication Architecture with Application Gateway

The application-layer translation mechanism MAY bring three main drawbacks: 1. End-to-end security channel like IPSec or TLS is not

supported, a malicious gateway may manipulate data that is transmitted between two terminals. 2. Non-IP terminals are invisible to the TCP/IP network, which makes it hard to conduct QoS or OAM operations, e.g., guaranteeing SLA of a specific non-IP terminal's traffic, or "ping" a non-IP terminal. 3. When a non-IP terminal joins or one leaves the network, corresponding rules SHOULD be configured on the gateway, thus increasing operation costs (i.e., OPEX).

Therefore, it would be beneficial to make those non-IP terminals adopt TCP/IP protocol suites, thus eliminating aforementioned drawbacks. The Internet Protocol Version 6 (IPv6) is expected to achieve the goal, however, it is challenging in some cases due to its long address and header length (40 bytes in total). For instance, it would consume more energy for power constrained terminals like IoT devices, and would amplify flow completion time on low-rate transport media or one with low MTU, thus decreasing user experiences.

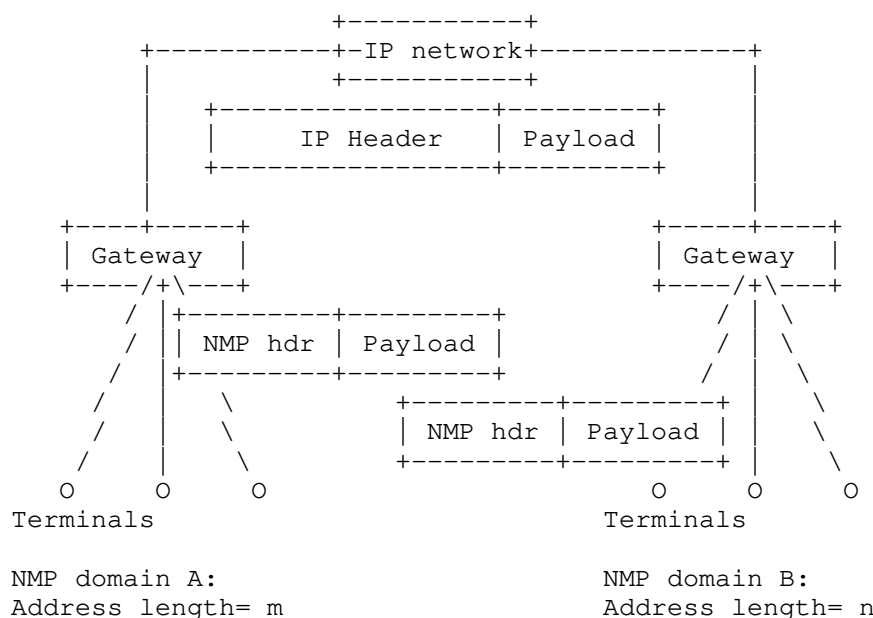
To this end, this document proposes Native Minimal Protocol (NMP), which is applied at edge networks by using minimal address length and fields. Simultaneously, NMP eliminates the drawbacks that may be brought by application layer gateways.

2. Requirements Notation

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] and [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Overview

NMP is an inter-node communication method and network layer protocol for edge network with native addresses. It is designed for extreme minimal IoT devices that communicate with each other and sometimes with normal IP nodes. NMP nodes and NMP gateways use native short addresses to identify themselves and use these addresses as source and destination addresses for network communication. NMP data packets and signaling packets are encapsulated in a simplified manner. The NMP-IPv6 translation function is deployed on the gateway to implement IP connections on the edge network. See Figure 2.



Only Support Extreme Simplified Control Messages within NMP Domain

Figure 2: Overview of Native Minimal Protocol

4. Protocol Design

4.1. Packet Header Format

The first bit at the beginning of the packet header indicates whether the packet is encapsulated with extreme concise format or not. If the first bit is 0, packet format is specified as follows Figure 3.

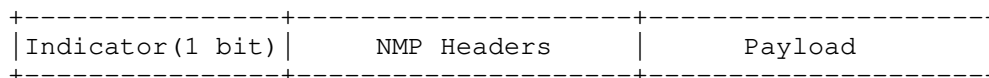


Figure 3: Basic Format of Native Minimal Protocol Packet

- o Indicator one-bit indicator to indicate whether extreme concise format is used. 0 - NMP headers follows; 1 - undefined.
- o NMP Headers Record the fields of packet header in Section 4.1.1 .

- o Payload The payload of the packet. For control plane packets, the control plane messages defined in Section 4.1.2 are carried in this part.

4.1.1. Data Packet Header Format

For data packet header, NMP uses bitmap with variable length to indicate which in-line headers appear in the packet. The specification is in Figure 4.

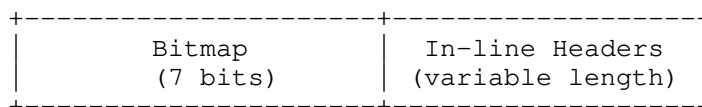


Figure 4: NMP Header Format

- o Bitmap A variable-length bitmap with at least 7 bits is used to indicate whether a NMP field is carried in a packet. The bit of 1 indicates that the packet carries this field and is located in the following in-line headers field. The value 0 indicates that the packet does not contain this field. The length of bitmap is defined as follows.

bitmap format	number of bits for indicator	scope fo headers
xxx xxx0	6 bits	header 1 ~ 6

- o In-line Headers The headers in NMP packet. Each header corresponds to a position in preceding bitmap.

Bitpos	Header Name	Header Length
1	TTL	8 bit
2	Total Length	16 bit
3	Next Header	8 bit
4	Reserved	N/A
5	Destination	Variable Length
6	Source	Variable Length
7	Next Bitmap Byte	N/A

4.1.2. Control Packet Format

Message Type (8 bits)	Checksum (8 bits)
--------------------------	----------------------

Figure 5: Control Packet Header Format

- o Type This field carries value to indicate the type of this control message.
- o Checksum The checksum is the 8-bit one's complement of the one's complement sum of the entire control message, starting with the message type field, and prepended with a "1" of indicator header fields, as specified in Section 4.1. For computing the checksum, the checksum field is first set to zero.

4.2. Control Messages

4.2.1. Address Request and Assignment Messages

A NMP host broadcasts an Address Request (AR) message to request an address from the gateway of the NMP domain. The gateway sends an Address Assignment (AA) message to the host to configure the host's NMP address. #### Format of Address Request The value of the message Type is 1. The message body is defined as follows.

-----+-----
ID length (8 bits) Host ID
-----+-----

- o ID length Length of this field is 1 octet. This header specifies the length of the Host ID field, in octets.
- o Host ID Indicates the identifier of the host that accesses the NMP network. The identifier can be a MAC address or another globally unique identifier.

4.2.1.1. Format of Address Assignment

The value of the message Type is 2. The message body is defined as follows.

-----+-----+-----+-----+-----
NMP Address Length NMP Gateway ID length Host ID
8 bits Address Address (8 bits)
-----+-----+-----+-----+-----

- o NMP Address Length Length of this field is 1 octet. This parameter specifies the length of the NMP address used in the local NMP domain.
- o NMP Address Network layer address assigned to the host node. The length is specified by the NMP Address Length field.
- o Gateway Address Network layer address of the gateway. The length is the same as length of NMP Address
- o ID length Length of this field is 1 octet. This header specifies the length of the Host ID field, in octets.
- o Host ID Indicates the identifier of the host that accesses the NMP network. The identifier can be a MAC address or another globally unique identifier.

4.2.2. Address Lease Extension Messages

To reduce the complexity of the NMP host, the gateway records the lease information of each NMP address. When the lease of a host address expires, the gateway sends a Renewal Challenge message to the host and waits for an response from the host. If a Renewal Response message is received from the host, the lease information is updated

based on the preconfigured strategy. Otherwise, the gateway releases the NMP address.

4.2.2.1. Renewal Challenge Message

The value of the message Type is 3. The message body is defined as follows.

NMP Address Length 8 bits	NMP Address	ID length (8 bits)	Host ID
------------------------------	----------------	-----------------------	---------

- o NMP Address Length Length of this field is 1 octet. This parameter specifies the length of the NMP address used in the local NMP domain.
- o NMP Address Network layer address assigned to the host node. The length is specified by the NMP Address Length field.
- o ID length Length of this field is 1 octet. This header specifies the length of the Host ID field, in octets.
- o Host ID Indicates the identifier of the host that accesses the NMP network. The identifier can be a MAC address or another globally unique identifier.

4.2.2.2. Renewal Response Message

The value of the message Type is 4. The message body is defined in Section 4.2.2.1.

4.3. DNS Delegation Messages

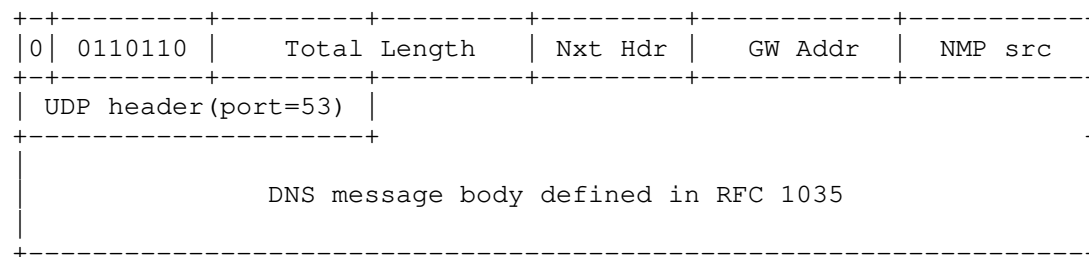
Many IoT products are written into the domain name of the IoT service platform when they are manufactured. The IP address of the server needs to be obtained through the DNS to establish communication.

Within the NMP domain, some modifications are required to traditional DNS messages in [RFC1035]. The NMP host sends a DNS query packet to the gateway. It Must set next header indicator to 1, the value of in-line next header is 17. Destination port in UDP header is 53. Destination of the packet is set to NMP address of gateway. When the gateway receives the packet, it directly translates the network layer information and sends a regular DNS packet to the DNS server configured on the gateway.

NMP is replaced by IPv6 protocol after the gateway. The source address is changed to 'IPv6 address prefix stored in the gateway + padding bit + NMP address', the destination address is changed to the DNS server address configured on the gateway, and the payload information remains unchanged.

When the DNS response packet sent by the DNS server reaches the gateway, the gateway resolves the response packet and allocates an available NMP address to the destination IPv6 address. The NMP address is used as an in-network mirror of the IPv6 address and replaces the target address in the DNS response packet. Then, the gateway sends the DNS response packet to the NMP host.

The header format of an DNS delegation message is defined as follows. For details about the format of a DNS message body, see [RFC1035].



4.4. Functionalities of Gateway

4.4.1. Address Management

The NMP gateway initializes the NMP address pool based on the network configuration and assigns an address to itself. This address is used as the default gateway by the hosts in the domain.

Intra-domain address management functionaliteis includes: * intra-domain host address allocation The gateway listens to the NMP address request message, allocates the corresponding NMP address based on the message content, generates an address assignment message, and returns the message to the host. The assigned addresses must meet the uniqueness requirements within the NMP domain. * intra-domain host address life cycle management The gateway manages the validity period of NMP addresses. The lease renewal challenge mechanism is used to renew or release host addresses.

4.4.2. Address Translation

The NMP address space can be mapped to specific subspaces of IPv6 address space. When traffic is destined to a destination outside the domain, the gateway translates the host address (source address) in the domain into an IPv6 address. For details about the translation method, see TBD.

For traffic from outside of the domain, determines whether the destination is within the domain. If the destination is within the domain, then the gateway translates the destination address to the corresponding NMP address.

5. Renumber Considerations

The NMP renumbering problem is not beyond the scope of [RFC6866] and [RFC7010], [RFC5887].

6. Security Considerations

Checksum is used to defend against malformed packets and null packet attacks caused by network bit errors. ICMPv6 uses a 16-bit checksum. NMP uses an 8-bit checksum to reduce the computing load on the host side and improve the packet encapsulation efficiency. This leads to a higher probability of network errors.

7. IANA Considerations

If NMP is running on Ethernet, a new Ethtype is required. In addition to Ethernet, other link-layer protocols that need to carry multiple upper-layer protocols need to assign specific identifiers to NMP to instruct devices to process network-layer packets according to this document.

This document requires to define new registry for NMP control message types, six of which are defined in this document.

8. Acknowledgments

The authors would like to acknowledge the contributions Guangpeng Li and Zhaochen Shi provided during the development of the solution.

9. Normative References

- [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>>.

- [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>>.
- [RFC5887] Carpenter, B., Atkinson, R., and H. Flinck, "Renumbering Still Needs Work", RFC 5887, DOI 10.17487/RFC5887, May 2010, <<https://www.rfc-editor.org/info/rfc5887>>.
- [RFC6866] Carpenter, B. and S. Jiang, "Problem Statement for Renumbering IPv6 Hosts with Static Addresses in Enterprise Networks", RFC 6866, DOI 10.17487/RFC6866, February 2013, <<https://www.rfc-editor.org/info/rfc6866>>.
- [RFC7010] Liu, B., Jiang, S., Carpenter, B., Venaas, S., and W. George, "IPv6 Site Renumbering Gap Analysis", RFC 7010, DOI 10.17487/RFC7010, September 2013, <<https://www.rfc-editor.org/info/rfc7010>>.
- [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

Sheng Jiang
Huawei Technologies
Beiqing Road, Haidian District
Beijing 100095
P.R. China

Email: jiangsheng@huawei.com

Zhe Chen
Huawei Technologies
Beiqing Road, Haidian District
Beijing 100095
China

Email: chenzhe17@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 15 July 2022

D. Li
J. Wu
Tsinghua University
M. Huang
Huawei
L. Qin
Tsinghua University
N. Geng
Huawei
11 January 2022

Source Address Validation: Use Cases and Gap Analysis
draft-li-sav-gap-analysis-01

Abstract

This document identifies the importance and use cases of source address validation (SAV) at both intra-domain level and inter-domain level (see [RFC5210]). Existing intra-domain and inter-domain SAV mechanisms, either Ingress ACL filtering [RFC2827], unicast Reverse Path Forwarding (uRPF) [RFC3704], or Enhanced Feasible-Path uRPF (EFP-uRPF) [RFC8704] has limitations in scalability or accuracy. This document provides gap analysis of the existing SAV mechanisms.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 8174 [RFC8174].

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 15 July 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	2
2. Terminology	4
3. Use Cases	4
3.1. Use Case 1: Intra-domain SAV	5
3.2. Use Case 2: Inter-domain SAV	6
4. Gap Analysis	6
4.1. Existing Intra-domain SAV mechanisms	6
4.2. Existing Inter-domain SAV mechanisms	7
5. SAV Requirements	9
6. Security Considerations	9
7. Contributors	9
8. Acknowledgments	9
9. Normative References	9
Authors' Addresses	10

1. Introduction

Source Address Validation (SAV) is important for defending against source address forgery attacks and accurately tracing back to the attackers. Considering that the Internet is extremely large and complex, it is very difficult to solve the source address spoofing problem at a single "level" or through a single SAV mechanism. On the one hand, it is unrealistic to require all networks to deploy a single SAV mechanism. On the other hand, the failure of a single SAV mechanism will completely disable SAV.

To address the issue, Source Address Validation Architecture (SAVA) was proposed [RFC5210]. According to the operating feature of the Internet, SAVA presents a hierarchical architecture which carries out source IP address validation at three checking levels, i.e., access network, intra-domain, and inter-domain. Different levels provide

different granularities of source IP address authenticity. In contrast to the single-level/point model, SAVA allows incremental deployment of SAV mechanisms while keeps effective because of its multiple-fence design. So, enhancing the source IP address validity in all the three checking levels is of high importance. Furthermore, one or more independent and loosely-coupled SAV mechanisms can coexist and cooperate under SAVA, which is friendly to different users (e.g., providers) with different policies or considerations. Obviously, the quality of SAV mechanisms for their target checking levels is key to the performance of SAV.

There are many SAV mechanisms for different checking levels. For the access network level, Source Address Validation Improvement (SAVI) was proposed to force each host to use legitimate source IP address[RFC7039]. SAVI acts as a purely network-based solution without special dependencies on hosts. It dynamically binds each legitimate IP address to a specific port/MAC address and verifies each packet's source address through the binding relationship. One of the most attractive features of SAVI is that it supports the maximally fine granularity of individual IP addresses, which previous ingress filtering mechanisms cannot provide.

At the intra-domain level, static Access Control List (ACL) is a typical solution of SAV. Operators can configure some matching rules to specify which kind of packets are acceptable (or unacceptable). The information of ACL should be updated manually so as to keep consistent with the newest filtering criteria, which inevitably limits the flexibility and accuracy of SAV. Strict unicast Reverse Path Forwarding (uRPF) [RFC3704] is another solution suitable to intra-domain. Routers deploying strict uRPF accept a data packet only when i) the local FIB contains a prefix encompassing the packet's source address and ii) the corresponding forwarding action for the prefix matches the packet's incoming interface. Otherwise, the packet will be dropped. However, in the scenarios (e.g., multihoming cases) where data packets are under asymmetric routing, strict uRPF often improperly blocks legitimate traffic.

At the inter-domain level, a combination of Enhanced Feasible-Path uRPF (EFP-uRPF) and loose uRPF is recommended in[RFC8704]. Particularly, EFP-uRPF is suggested to be applied on customer interfaces. EFP-uRPF on an AS can prevent its customers from spoofing its upstream ASes' source addresses but fails in the case of two customers spoofing each other. On lateral peer interfaces and transit provider interfaces, loose uRPF [RFC3704] is taken. The routers deploying loose uRPF accept any packets whose source addresses appear in the local FIB tables. Due to the loss of directionality, loose uRPF often improperly permits spoofed traffic.

To summarize, given that it is impossible to deploy SAVI on every access network in the Internet, the "fences" at intra- and inter-domain levels are very important for filtering source address forgery packets that are let go by access networks. However, there exist some instinctive drawbacks in the existing SAV mechanisms designed for both the intra- and inter-domain levels, which leads to inevitable improper permit or improper block problems. A more complete SAV mechanism is required for both intra- and inter-domain levels.

This document identifies the use cases of intra- and inter-domain SAVs. These cases will help analyze the instinctive drawbacks of the existing SAV mechanisms. After that, some SAV requirements will be presented.

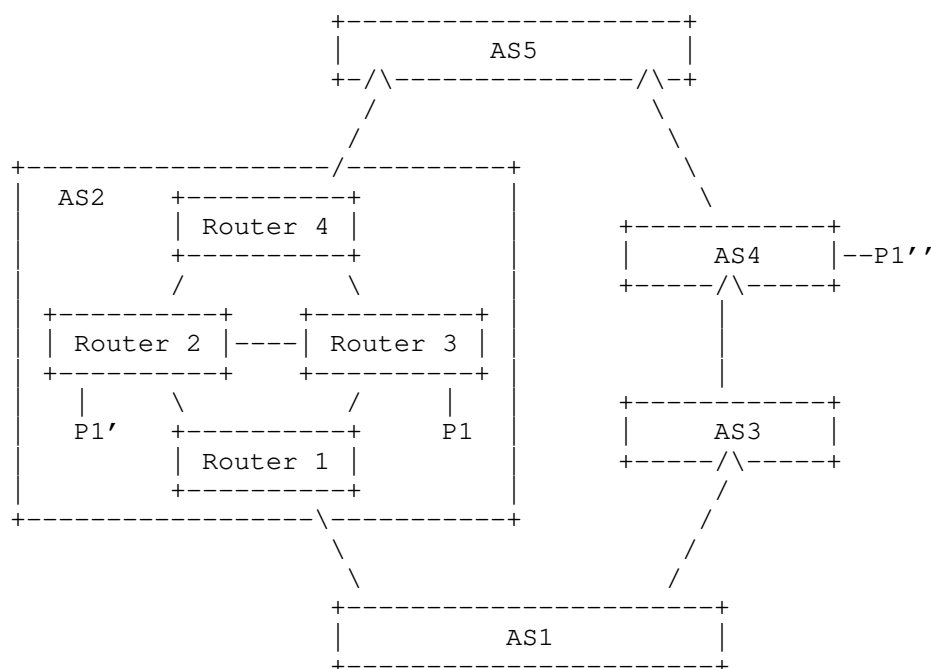
2. Terminology

EAST-WEST traffic denotes the traffic originated and terminated within an AS. Intra-domain SAV aims to check EAST-WEST traffic and prevents hosts/routers from spoofing other source IP blocks in the same AS.

NORTH-SOUTH traffic denotes the traffic arriving from an external AS. Particularly, the traffic arriving from the customer AS is Northward traffic. The traffic received from the provider/peer AS is Southward traffic. Inter-domain SAV aims to verify the authenticity of the source address of NORTH-SOUTH traffic.

3. Use Cases

Figure 1 illustrates the use cases of SAV in both intra- and inter-domain levels. AS1-AS5 belong to the same customer cone, and AS1 is the stub AS. The topology of AS2 is presented while other ASes' inner structures are hidden for brevity.



P1 is the source IP address prefix of Router3.

P1' is the spoofed P1 by Router2 located in the same AS as Router3.

P1'' is the spoofed P1 by Routers located in another AS, i.e., AS4.

Figure 1: Illustration of the use cases of SAV in both intra- and inter-domain levels

3.1. Use Case 1: Intra-domain SAV

In some scenarios especially very large ASes, hosts/routers in the same AS may spoof each other's IP addresses. In Figure 1, Router2 spoofs P1 that originates from Router3. With Intra-domain SAV, EAST-WEST traffic can be checked, and source address spoofing attacks can be prevented. In the figure, Router1, Router3, and Router4 will drop the packets with P1' while accept those with P1, when they deploy Intra-domain SAV mechanisms. Overall, Intra-domain SAV can prevent the source address spoofing from the same AS.

3.2. Use Case 2: Inter-domain SAV

In Figure 1, AS4 spoofs AS2's IP address prefix, i.e., P1 originated from Router3. AS5 will receive the Northward traffic from AS2 and AS4 with legitimate and spoofed IP addresses, respectively. An SAV mechanism is necessary for AS5 to drop the illegal traffic. From the view point of Southward traffic, AS1 may also receive spoofed traffic from AS3 (if AS3 accepts the data packets with source prefix P1"). So, the deployment of SAV on AS1 is also important. Overall, Inter-domain SAV is necessary and can improve the confidence of the source IP address validity among ASes.

4. Gap Analysis

High accuracy is the basic requirement of any intra- or inter-domain SAV mechanism. For any SAV mechanism, improper block problems must be avoided because legitimate traffic must not be influenced. On that basis, SAV should also reduce improper permit problems as much as possible. However, existing SAV mechanisms can not well meet these requirements.

4.1. Existing Intra-domain SAV mechanisms

Operators can configure static ACLs on border routers to validate source addresses. The main drawback of ACL-based SAV is the high operational overhead. Limited application scenarios make the ACL-based method unable to do sufficient SAV on EAST-WEST traffic.

Strict uRPF can generate SAV tables automatically, but it also has limited application scenarios. Figure 2 illustrates an intra-domain scenario. In the scenario, AS1 runs strict uRPF. An access network having IP address prefix 10.0.0.0/15 is attached to two border routers (Router1 and Router2) of AS1. Due to customer's policy, it advertises 10.0.0.0/16 to Router1 and 10.1.0.0/16 to Router2. Then, Router1 and Router2 will advertise the learned IP address prefixes to other routers in AS1 through intra-domain routing protocols such as OSPF and IS-IS.

Although customer only advertises 10.0.0.0/16 to Router1, it may send packets with source IP addresses belonging to 10.1.0.0/16 to Router1 due to load balancing requirements. Suppose the destination node is Router5. Then the path to destination is Customer->Router1->Router3->Router5, while the reverse path is Router5->Router4->Router2->Customer. The round trip routing path is asymmetric, which cannot be dealt with well by strict uRPF.

Specifically speaking, strict uRPF is faced with improper block problems under asymmetric routing scenarios. When Router1/Router3 runs strict uRPF, it learns SAV rules that packets with source address prefix of 10.0.0.0/16 must enter the router on interface '#'. When the packets with source addresses of 10.1.0.0/16 arrive, they will be dropped, which results in improperly blocking legitimate traffic. Similarly, when strict uRPF is deployed on Router2, the improper block problem still exists.

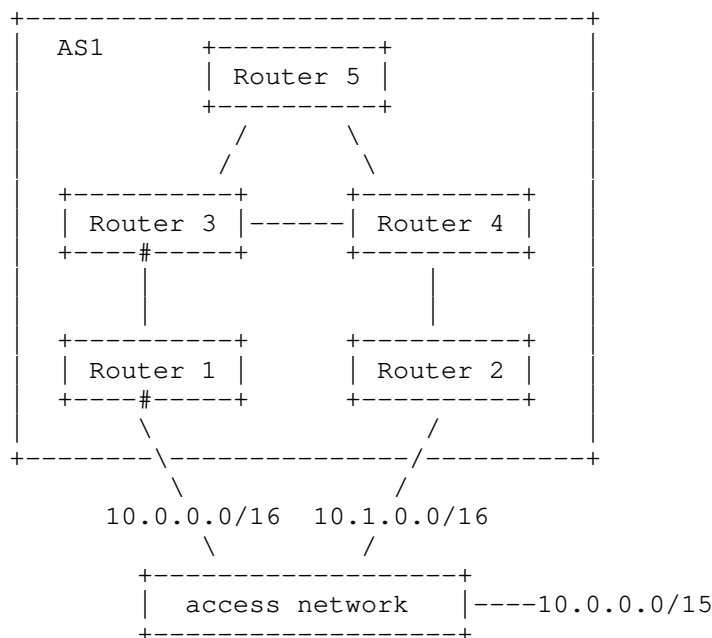


Figure 2: An intra-domain scenario

4.2. Existing Inter-domain SAV mechanisms

The most popular inter-domain SAV is suggested by [RFC8704], which combines EFP-uRPF algorithm B and loose uRPF. In particular, EFP-uRPF algorithm B is for Northward traffic validation. It sacrifices the directionality of customer interfaces for reducing improper permit cases. Loose uRPF is for validating Southward traffic on lateral peer and transit provider interfaces. It sacrifices directionality of Southward traffic completely. Such a combined method sacrificing directionality will lead to improper permit problems sometimes.

Figure 3 illustrates a common inter-domain scenario where the above inter-domain SAV method will fail. In the figure, there are two customer ASes, i.e., AS1 and AS2. Both of them are attached to a

provider AS, i.e., AS4. AS4 has a lateral peer and a provider, i.e., AS3 and AS5. Particularly, AS1 has IP address prefix P1 and advertises it to AS4. IP address prefix P2 is allocated to AS2 and is also advertised to AS4. AS3 has IP address prefix P3 and AS5 has IP address prefix P5. P3 and P5 are also advertised to AS4 through BGP. All arrows represent BGP advertisements. Assume AS4 deploys inter-domain SAV policies, i.e., a combination of EFP-uRPF algorithm B and loose uRPF.

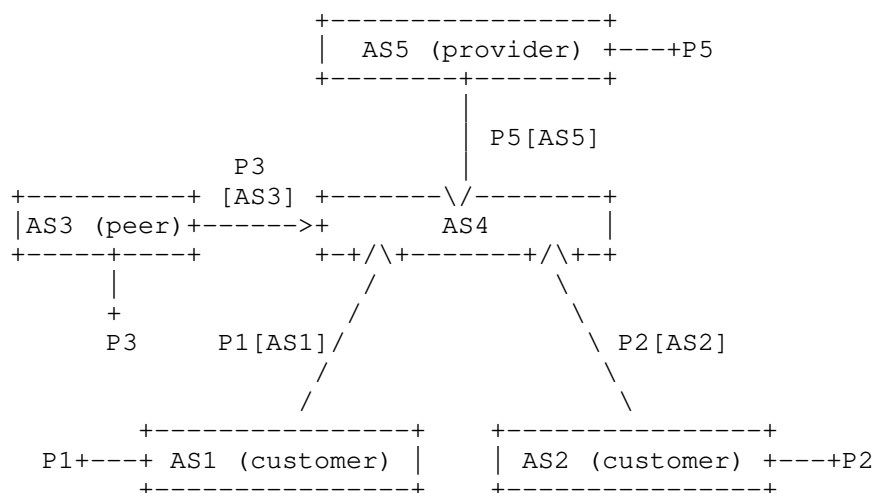


Figure 3: An inter-domain scenario

For Northward traffic, AS4 applies EFP-uRPF. Under EFP-uRPF, AS4 will generate SAV rules considering P1 and P2 are legitimate on both the two customer interfaces. When AS1 spoofs IP address prefix P2 of AS2, the malicious Northward traffic cannot be filtered by AS4. The same is true when AS2 forges P1 of AS1. That is to say, EFP-uRPF cannot prevent source address spoofing among customers even though it only focus on Northward traffic.

For Southward traffic, AS4 deploys loose uRPF for the interfaces of AS3 and AS5. It will learn that the packets with source addresses of P3 or P5 can be accepted without validating the specific arrival interface. Since loose uRPF loses directionality completely, it obviously will fail in dealing with the source address spoofing between its lateral peer and provider, i.e., AS3 and AS5.

5. SAV Requirements

High accuracy, i.e. avoiding improper block problems while trying best to reduce improper permit problems, is the basic requirement of an ideal SAV mechanism. As described above, existing SAV mechanisms cannot meet this requirement. The root cause of their limitations is that they all achieve SAV based on local forwarding information base (FIB) or routing information base (RIB), which may not match the real forwarding direction from the source. In order to guarantee the accuracy, SAV should follow the real data-plane forwarding path. To solve this problem and provide accurate SAV for arbitrary network scenarios, it is required to exchange/explore/probe the forwarding-path information among routers/ASes. In other words, network-wide protocols should be considered.

The network-wide protocols should also consider some practical issues:

- * High scalability. The protocols should not induce much overhead (e.g., bandwidth cost of path probing). Fast convergence under environment changes is also important for improving the scalability in different scales of networks.
- * High deployability. A strategy of incremental deployment needs to be considered. If some routers/ASes do not support the new protocols, improper block should be avoided.
- * High security. The protocols should include mechanisms to guarantee the integrity of protocol packets. Security risks such as Man-in-the-Middle Attack should be avoided.

6. Security Considerations

TBD

7. Contributors

TBD

8. Acknowledgments

TBD

9. Normative References

- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC5210] Wu, J., Bi, J., Li, X., Ren, G., Xu, K., and M. Williams, "A Source Address Validation Architecture (SAVA) Testbed and Deployment Experience", RFC 5210, DOI 10.17487/RFC5210, June 2008, <<https://www.rfc-editor.org/info/rfc5210>>.
- [RFC7039] Wu, J., Bi, J., Bagnulo, M., Baker, F., and C. Vogt, Ed., "Source Address Validation Improvement (SAVI) Framework", RFC 7039, DOI 10.17487/RFC7039, October 2013, <<https://www.rfc-editor.org/info/rfc7039>>.
- [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>>.
- [RFC8704] Sriram, K., Montgomery, D., and J. Haas, "Enhanced Feasible-Path Unicast Reverse Path Forwarding", BCP 84, RFC 8704, DOI 10.17487/RFC8704, February 2020, <<https://www.rfc-editor.org/info/rfc8704>>.

Authors' Addresses

Dan Li
Tsinghua University
Beijing
China

Email: tolidan@tsinghua.edu.cn

Jianping Wu
Tsinghua University
Beijing
China

Email: jianping@cernet.edu.cn

Mingqing Huang
Huawei
Beijing
China

Email: huangmingqing@huawei.com

Lancheng Qin
Tsinghua University
Beijing
China

Email: qlc19@mails.tsinghua.edu.cn

Nan Geng
Huawei
Beijing
China

Email: gengnan@huawei.com

Internet Engineering Task Force	S.D. Schoen
Internet-Draft	J. Gilmore
Updates: 1122, 3704, 6890 (if approved)	D. Täht
Intended status: Standards Track	IPv4 Unicast Extensions Project
Expires: 8 September 2022	7 March 2022

Unicast Use of the Formerly Reserved 240/4
draft-schoen-intarea-unicast-240-02

Abstract

This document redesignates 240/4, the region of the IPv4 address space historically known as "Experimental," "Future Use," or "Class E" address space, so that this space is no longer reserved. It asks implementers to make addresses in this range fully usable for unicast use on the Internet.

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 8 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	2
1.1. Requirements Language	2
2. Background	2
2.1. History of IPv4 Address Types	3
2.2. Reserved IPv4 Addresses in the RFC Series	4
2.3. Attempts to Use the "Future Use" Addresses	4
2.4. Recent Use as Ordinary Unicast Addresses	5
3. Change in Status of 240/4	6
3.1. Continued Special Treatment for 255.255.255.255/32	7
4. Compatibility and Interoperability	7
5. IANA Considerations	9
6. Security Considerations	9
6.1. Existing Unofficial Uses of 240/4	11
7. Acknowledgements	11
8. References	11
8.1. Normative References	11
8.2. Informative References	12
Appendix A. Implementation Status	15
A.1. Operating systems	15
A.2. Other implementations	16
A.3. Internet of Things	16
Authors' Addresses	16

1. Introduction

With ever-increasing pressure to conserve IP address space on the Internet, it makes sense to consider where relatively minor changes can be made to fielded practice to improve numbering efficiency. One such change, proposed by this document, is to redefine the "Experimental" or "Future Use" 240/4 region (historically known as "Class E" addresses) as ordinary unicast addresses. These 268 million IPv4 addresses are already usable for unicast traffic in many popular implementations today. Standardization as unicast addresses will eventually allow them to be later deployed by Internet stewardship organizations to relieve address space scarcity.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Background

2.1. History of IPv4 Address Types

When the Internet Protocol was being designed, it was unclear whether it would be a success, or which of its features might be the key features that led to success. The bulk of its address space was dedicated to ordinary "host addresses". Other blocks and corners of the address space were reserved, either for particular protocol functions such as loopback, LAN broadcasting, or host bootstrapping, or for future definition. A major allocation of 268 million addresses was later made for multicasting [RFC0988], while leaving another 268 million reserved for "future use". After the invention of broadcast and multicast, the original ordinary host addresses were later described as unicast addresses, which is now the usual terminology.

With decades of hindsight, we can now see that unicast has been the success story of the Internet. Trillions of unicast packets now move around the world daily. By contrast, the non-unicast addresses are seldom used. The use of routable broadcast packets in denial of service attacks has now limited broadcast packets to local-area networks [RFC2644], and to critical but highly-specialized protocol functions such as DHCP [RFC2131], routing updates [RFC1256], or neighbor discovery.

Wide-area multicast packets had a brief research heyday, but never reached critical mass. Today, the overwhelming majority of multiply-replicated media streams (such as popular songs and videos, television programs, conference calls, and video meetings) are carried in unicast packets mediated by application-level replication rather than IP-protocol-level multicasting or broadcasting.

The Internet became a rapid worldwide success. Partly due to the reduction in experimentation that accompanied that success, little effort has been paid to looking back at the historical allocations of reserved addresses. The success of unicast traffic has led to a huge demand for unicast addresses. By contrast, there is far more supply of reserved, ignored, loopback, and multicast addresses than any foreseeable IPv4 Internet will demand. Most of these historical accidents were not carried forward into the IPv6 protocol [RFC4291]. We propose simple, compatible changes to existing IPv4 implementations that will increase the supply of unicast addresses by redesignating addresses that today are almost completely unused on the Internet. The best and easiest "future use" of many of today's formerly reserved IPv4 addresses is as ordinary unicast addresses.

2.2. Reserved IPv4 Addresses in the RFC Series

The Assigned Numbers RFC series reserved various IP addresses or assigned them special meanings, starting in 1977 and continuing through the early 1990s. The detailed behavioral requirements for IPv4 implementations based on these designations are set out in October 1989's RFC 1122 [RFC1122]. As other special cases continued to be introduced on occasion, RFC 3232 [RFC3232] announced that IANA would track such information in an online database; the present-day version of this mechanism is the IPv4 Special-Purpose Address Registry [IANA4], as provided for by RFC 6890 [RFC6890]. A wide range of host and network software follows these designations by treating these Internet addresses specially.

This document is concerned with the largest special case in RFC 1122: the designation of an entire /4 block for Future Use. In retrospect, the flexibility offered by keeping these addresses unused was insightful for its time, but since they ended up never being needed for any special purposes, they have become the least productive portion of the Internet address space.

The largest block of original addresses reserved for future use in 1983 was called "Class D" in RFC 870 [RFC0870], and contained what would now be called 224/3. This contained about 536 million addresses, about 12.5% of the total available address space. By 1986, RFC 988 [RFC0988] split the former Class D in half, designating a multicast Class D block, now called 224/4, and a future-use Class E block, now called 240/4. Following the 1993 implementation of CIDR [RFC1519] and its 2006 clarification [RFC4632], we no longer speak of any IPv4 address as having an "address class," but the reservations of these specific addresses that were made by RFC 1122, were unaffected by the CIDR change in terminology and routing technology.

2.3. Attempts to Use the "Future Use" Addresses

Through the 1980s, there were many reasons to suppose that new forms of Internet addressing could emerge, so reserving a substantial number of addresses for them was prudent.

One likely candidate for some time was protocol translation methods between IP and other protocols using special surrogate IP addresses. This possibility was particularly significant during the time frame when IP coexisted widely on heterogeneous networks with other protocols. Special number ranges could have been used to facilitate interoperability, protocol translation, or encapsulation between IP and non-IP protocols.

This prospect received new salience with the adoption of IPv6, where some deployed or proposed transition mechanisms use special-purpose IPv4 addresses with a distinctive meaning in the context of IPv6 transition, such as NAT64 [RFC7050] and the deprecated 6to4 [RFC3068]. While IPv6 transition mechanisms could conceivably have used portions of 240/4, they ended up instead using very small amounts of special address space from the IETF Protocol Assignments block 192.0.0.0/24 or elsewhere within the unicast space.

Another form of addressing that was novel in 1989 is anycast addressing, in which the same address is used to identify servers at physically distinct locations and connected to the Internet at different points. It would have been possible to designate a new "class" of addresses for anycast operations. RFC 1546 [RFC1546], which first defined anycast, concluded that this would be a possible and even desirable approach:

| There appear to be a number of ways to support anycast addresses,
| some of which use small pieces of the existing address space,
| others of which require that a special class of IP addresses be
| assigned. [...] In the balance it seems wiser to use a separate
| class of addresses.

But anycast services turned out to work fine in most respects by using existing unicast routing protocols, existing unicast datagram delivery protocols, and ordinary unicast addresses. They are now widely used for specific applications [RFC7094] such as the Internet's root nameservers.

2.4. Recent Use as Ordinary Unicast Addresses

Overall, 30 years of experience have demonstrated that no new addressing mechanism requires the use of 240/4; nor is any likely to require it in the future, particularly in light of the IPv6 transition. Other explicit reservations such as the IETF Protocol Assignments block at 192.0.0.0/24 have been sufficient. While it was reasonable to plan for an unknown future, the reserved block at 240/4 did not ultimately aid Internet innovation or functionality. The future has arrived, and it wants IPv4 unicast addresses far more than it wants permanently unusable IPv4 addresses.

The idea of making 240/4 addresses available for unicast addressing is not new. It was suggested by Lear on the influential TCP-IP mailing list in 1988 [Lear]. It was formally proposed to IETF more than a decade ago, both by Fuller, Lear, and Mayer [FLM], and by Wilson, Michaelson, and Huston [WMH]. While the idea of unicast use of 240/4 was merely being considered at IETF, the "running code" required was simple enough and compatible enough that this behavior

change was implemented at that time in several operating systems. Then, when the protocol change was ultimately not standardized, those implementations remained, but were largely forgotten. (They are summarized in the "Implementation Status" section of this document.)

The unicast support created in about 2008 in those implementations is now running in millions of nodes on the Internet, and has not caused any problems over the past decade. As a result, the 240/4 space has been attracting "wildcat" use in private networks; see [VPC].

Although software support for unicast use of 240/4 is widespread, it is not yet universal. The present document moves this process further along by confirming the consensus that unicast is the preferred use for 240/4, documenting the exact behavior changes required for maximum interoperability, and calling on all vendors and implementers to adopt this behavior. Doing so will prepare for a future in which use of these addresses is anticipated and unsurprising, so that their allocation can be considered.

Implementations generally treat public and private addresses identically, with the differences occurring only in how routes, firewalls, and DNS servers are configured. The earlier draft [WMH] suggested designating the unreserved 240/4 range as [RFC1918]-style private address space. Like the [FLM] draft, this document does not attempt to decide or designate whether future allocations from this address range will be public or private addresses. Both options require that both hosts and routers be able to use these addresses, so the next section fully defines both host and router behavior.

3. Change in Status of 240/4

The purpose of this document is to make addresses in the range 240/4 available for active unicast use on the public Internet. This includes supporting them for numbering and addressing networks and hosts, like any other unicast address.

Host and router software SHOULD treat addresses in the 240/4 range in the same way that they would treat other unicast IPv4 addresses. Software SHOULD be capable of accepting datagrams from, and generating datagrams to, addresses within this range.

Clients for autoconfiguration mechanisms such as DHCP [RFC2131] SHOULD accept a lease or assignment of an address within 240/4 whenever the underlying operating system is capable of accepting it.

Other interoperability details related to address-based filtering are discussed in a separate section, below.

3.1. Continued Special Treatment for 255.255.255.255/32

The address 255.255.255.255/32 was given a special meaning as a local segment limited broadcast address by numerous prior Internet standards, starting with RFC 919 [RFC0919] and continuing consistently up to the present day. For example, 255.255.255.255 is used as a network-layer destination address in BOOTP [RFC0951] and DHCP [RFC2131] for address autoconfiguration broadcasts by hosts that don't yet know anything about the networks to which they are connected. While some newer autoconfiguration or autodiscovery protocols use other addresses, the use of 255.255.255.255 remains widespread.

The special meaning of 255.255.255.255 was never restricted or affected by the reservation of 240/4. Accordingly, the existing distinctive meaning of 255.255.255.255 is unchanged by this document. This single address MUST NOT be assigned to an individual host, or interpreted as the address of an individual host, even if it would otherwise be part of an allocated or announced network block.

4. Compatibility and Interoperability

Older Internet standards counseled implementations in varying ways to reject packets from, and not to generate packets to, addresses within 240/4.

RFC 1122 [RFC1122], section 3.2.1.3, states that a "host MUST silently discard an incoming datagram containing an IP source address that is invalid by the rules of this section." The same section states that Class E addresses are "reserved" (which might be taken, in context, to imply that they are "invalid"); the section further treats Class A, B, and C as the only possibly relevant address ranges for unicast addressing.

RFC1812 [RFC1812], section 5.3.7, states that a "router SHOULD NOT forward" a packet with such a destination address. (If section 4.2.2.11's reference to these addresses as "reserved" is taken to imply that they are "special," section 5.3.7 would also imply that a "router SHOULD NOT forward" a packet with such a source address.)

RFC 3704 [RFC3704] (BCP 84) cites RFC 2827 [RFC2827] (BCP 38) in asking providers to filter based on source address:

RFC 2827 recommends that ISPs police their customers' traffic by dropping traffic entering their networks that is coming from a source address not legitimately in use by the customer network. The filtering includes but is in no way limited to the traffic whose source address is a so-called "Martian Address" - an address

that is reserved, including any address within 0.0.0.0/8, 10.0.0.0/8, 127.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 224.0.0.0/4, or 240.0.0.0/4.

In this context, RFC 3704 specifies filtering of these addresses as source (not destination) addresses at a network ingress point as a countermeasure against forged source addresses, limiting forwarded packets' source addresses to only the set which have been actually assigned to the customer's network. The RFC's mention of these "Martian Addresses" is based on the assumption that they could never be legitimately in use by the customer network.

Because the 240/4 address space is no longer reserved as a whole, an address within this space is no longer inherently a "Martian" address. Both hosts and routers MUST NOT hard-code a policy of always rejecting such addresses. Hosts and routers SHOULD NOT be configured to apply Martian address filtering to any packet solely on the basis of its reference to a source (or destination) address in 240/4. Maintainers of lists of "Martian addresses" MUST NOT designate addresses from this range as "Martian". As noted elsewhere, the address 255.255.255.255 retains its special meaning, but is also not a "Martian" address.

The filtering recommended by RFC 3704 is designed for border routers, not for hosts. To the extent that an ISP had allocated an address range from within 240/4 to its customer, RFC 3704 would already not require packets with those source addresses to be filtered out by the ISP's border router.

Since deployed implementations' willingness to accept 240/4 addresses as valid unicast addresses varies, a host to which an address from this range has been assigned may also have a varying ability to communicate with other hosts.

Such a host might be inaccessible by some devices either on its local network segment or elsewhere on the Internet, due to a combination of host software limitations or reachability limitations in the network. IPv4 unicast interoperability with 240/4 can be expected to improve over time following the publication of this document. Before or after allocations are eventually made within this range, "debogonization" efforts for allocated ranges can improve reachability to the whole address block. Similar efforts have already been done by Cloudflare on 1.1.1.1 [Cloudflare], and by RIPE Labs on 1/8 [RIPElabs18], 2a10::/12 [RIPElabs2a1012], and 128.0/16 [RIPElabs128016]. The Internet community can use network probing with any of several measurement-oriented platforms to investigate how usable these addresses are at any particular point in time, as well as to localize medium-to-large-scale routing problems. (Examples are

described in [Huston], [NLNOGRing], and [Atlas].) Any network operator to whom such addresses are made available by a future allocation will have to examine the situation in detail to determine how well its interoperability requirements will be met.

5. IANA Considerations

This memo unreserves the address block 240/4. It therefore requests IANA to update the IANA Special-Purpose Address Registry by removing the entry for 240/4, whose existing authority is RFC 1122, Section 4. Additionally, it requests IANA to update the IANA IPv4 Address Space Registry by changing the status of each /8 entry from 240/8 through 255/8 from "Future Use, 1981-09, RESERVED" to "Unallocated, [Date of this RFC], UNALLOCATED". Finally, IANA is requested to prepare for this address space to be addressed in the reverse DNS space in in-addr.arpa.

This memo does not effect a registration, transfer, allocation, or authorization for use of these addresses by any specific entity. This memo's scope is to require IPv4 software implementations to support the ordinary unicast use of addresses in the newly unallocated range 240.0.0.0 through 255.255.255.254. During a significant transition period, it would only be prudent for the global Internet to use those addresses for experimental purposes such as debogonization and testing. After that transition period, a responsible entity such as IETF or IANA could later consider whether, how and when to allocate those addresses to entities or to other protocol functions such as private addresses.

6. Security Considerations

The change specified by this document could create a period of ambiguity about historical and future interpretations of the meaning of host and network addresses in 240/4. Some networks and hosts currently discard all IPv4 packets bearing these addresses, pursuant to statements in prior standards that packets containing these addresses have no agreed-upon meaning. Such implementations have protected themselves from possible incompatible future packet formats that might have eventually used these addresses.

Disparate filtering processes and rules, both at present, and in response to the adoption of this document, could make it easier for rogue network operators to hijack or spoof portions of this address space in order to send malicious traffic.

Live traffic, accepted and processed by other devices, may legitimately originate from these addresses in the future. Network operators, firewalls, and intrusion-detection systems may need to take account of this change in various regards, to avoid permitting either more or less traffic from such addresses than they expected.

Automated systems generating reports, and human beings reading those reports, SHOULD NOT assume that the use of a 240/4 source address indicates spoofing, an attack, or a new incompatible packet format. At the same time, they SHOULD NOT assume that the use of 240/4 is impossible or will be precluded by other systems' behavior.

An important concern about the [FLM] and [WMH] drafts was that discrepant behavior between systems could create security problems, as when a middlebox fails to detect or report an attack or policy violation because it believes that an address involved cannot be used or cannot be relevant. Similarly, a logging system could fail to log traffic related to 240/4 addresses because it incorporates an assumption that no such traffic can ever occur. Such discrepancies between multiple systems' views of communication semantics are a common security antipattern. (Compare [Sherr], exploiting discrepancies in telephony equipment's recognition and interpretation of DTMF signals.) Any change to the meaning or status of a group of addresses can introduce such a discrepancy.

In this case, because 240/4 is already commonly supported by several widely-used implementations, and is already used for private network communications, such discrepancies are already a reality. If routers follow this document's request to cease filtering this address range, they will increase the variety of contexts in which implementations may receive ordinary unicast packets containing these addresses. (Such packets are still unlikely to arrive from distant hosts until some of these addresses are eventually allocated for experimental or production use, and until the global routing table receives announcements for subnets in this range.)

The adoption of this document will converge on an explicitly shared understanding that implementations should prepare for this possibility. Since unofficial private use of 240/4 addresses is a reality today, while any public allocations from this range are still distant and contingent on further study, implementers are receiving considerable advance notice of this issue.

6.1. Existing Unofficial Uses of 240/4

Some organizations are reportedly using portions of 240/4 internally as RFC 1918-type private-use address space, for example for internal communications within datacenters. Google has advised hosting customers [VPC] that they may use this address space this way. Future allocations of 240/4 could result in use of this space on the public Internet in ways that overlap these unofficial private-use addresses, creating ambiguity about whether a particular host intended to use such an address to refer to a private or public network. Among other unintended outcomes, hosts or firewalls that have extended greater trust to other hosts based on their use of a certain unofficial network number (that was considered to imply presence on a LAN or within an organization) may eventually receive legitimate traffic from an external network to which this address space has been allocated.

Operators of networks that are making unofficial uses of portions of 240/4 may wish to plan to discontinue these uses and renumber their internal networks, or to request that IANA formally designate certain ranges as additional Private-Use areas.

7. Acknowledgements

This document directly builds on prior work by Dave Täht and John Gilmore as part of the IPv4 Unicast Extensions Project.

8. References

8.1. Normative References

- [IANA4] Internet Assigned Numbers Authority, "IANA IPv4 Special-Purpose Address Registry",
<<https://www.iana.org/assignments/iana-ipv4-special-registry/iana-ipv4-special-registry.xhtml>>.
- [RFC0870] Reynolds, J. and J. Postel, "Assigned numbers", RFC 870, DOI 10.17487/RFC0870, October 1983,
<<https://www.rfc-editor.org/info/rfc870>>.
- [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>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995,
<<https://www.rfc-editor.org/info/rfc1812>>.

- [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>>.
- [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>>.
- [RFC2827] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, DOI 10.17487/RFC2827, May 2000, <<https://www.rfc-editor.org/info/rfc2827>>.
- [RFC3704] Baker, F. and P. Savola, "Ingress Filtering for Multihomed Networks", BCP 84, RFC 3704, DOI 10.17487/RFC3704, March 2004, <<https://www.rfc-editor.org/info/rfc3704>>.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <<https://www.rfc-editor.org/info/rfc4632>>.
- [RFC6890] Cotton, M., Vegoda, L., Bonica, R., Ed., and B. Haberman, "Special-Purpose IP Address Registries", BCP 153, RFC 6890, DOI 10.17487/RFC6890, April 2013, <<https://www.rfc-editor.org/info/rfc6890>>.
- [RFC7050] Savolainen, T., Korhonen, J., and D. Wing, "Discovery of the IPv6 Prefix Used for IPv6 Address Synthesis", RFC 7050, DOI 10.17487/RFC7050, November 2013, <<https://www.rfc-editor.org/info/rfc7050>>.

8.2. Informative References

- [Atlas] RIPE Network Coordination Centre, "RIPE Atlas", <<https://atlas.ripe.net/>>.
- [Cloudflare] Strong, M., "Fixing reachability to 1.1.1.1, GLOBALLY!", 4 April 2018, <<https://blog.cloudflare.com/fixing-reachability-to-1-1-1-1-globally/>>.

- [FLM] Fuller, V., Lear, E., and D. Meyer, "Reclassifying 240/4 as usable unicast address space", Work in Progress, Internet-Draft, draft-fuller-240space-02, 25 March 2008, <<https://datatracker.ietf.org/doc/html/draft-fuller-240space-02>>.
- [Huston] Huston, G., "Detecting IP Address Filters", 13 January 2012, <<https://labs.ripe.net/author/gih/detecting-ip-address-filters/>>.
- [Lear] Lear, E., "Re: Running out of Internet addresses?", TCP-IP mailing list, 27 November 1988, <https://web.archive.org/web/20120514082839/http://www-mice.cs.ucl.ac.uk/multimedia/misc/tcp_ip/8813.mm.www/0146.html>.
- [NLNOGRing] NLNOG RING, "10 Years of NLNOG RING", <<https://ring.nlnog.net/post/10-years-of-nlnog-ring/>>.
- [RFC0919] Mogul, J., "Broadcasting Internet Datagrams", STD 5, RFC 919, DOI 10.17487/RFC0919, October 1984, <<https://www.rfc-editor.org/info/rfc919>>.
- [RFC0951] Croft, W. and J. Gilmore, "Bootstrap Protocol", RFC 951, DOI 10.17487/RFC0951, September 1985, <<https://www.rfc-editor.org/info/rfc951>>.
- [RFC0988] Deering, S., "Host extensions for IP multicasting", RFC 988, DOI 10.17487/RFC0988, July 1986, <<https://www.rfc-editor.org/info/rfc988>>.
- [RFC1256] Deering, S., Ed., "ICMP Router Discovery Messages", RFC 1256, DOI 10.17487/RFC1256, September 1991, <<https://www.rfc-editor.org/info/rfc1256>>.
- [RFC1519] Fuller, V., Li, T., Yu, J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, DOI 10.17487/RFC1519, September 1993, <<https://www.rfc-editor.org/info/rfc1519>>.
- [RFC1546] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, DOI 10.17487/RFC1546, November 1993, <<https://www.rfc-editor.org/info/rfc1546>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<https://www.rfc-editor.org/info/rfc2131>>.

- [RFC2644] Senie, D., "Changing the Default for Directed Broadcasts in Routers", BCP 34, RFC 2644, DOI 10.17487/RFC2644, August 1999, <<https://www.rfc-editor.org/info/rfc2644>>.
- [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>>.
- [RFC3232] Reynolds, J., Ed., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, DOI 10.17487/RFC3232, January 2002, <<https://www.rfc-editor.org/info/rfc3232>>.
- [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>>.
- [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>>.
- [RIPElabs128016] Aben, E., "The Curious Case of 128.0/16", 6 December 2011, <<https://labs.ripe.net/author/emileaben/the-curious-case-of-128016/>>.
- [RIPElabs18] Schwarzsinger, F., "Pollution in 1/8", 3 February 2010, <<https://labs.ripe.net/author/franz/pollution-in-18/>>.
- [RIPElabs2a1012] Aben, E., "The Debogonisation of 2a10::/12", 17 January 2020, <<https://labs.ripe.net/author/emileaben/the-debogonisation-of-2a1012/>>.
- [Sherr] Sherr, M., Cronin, E., Clark, S., and M. Blaze, "Signaling vulnerabilities in wiretapping systems", IEEE Security & Privacy November-December 2005, <<https://www.matthblaze.org/papers/wiretap.pdf>>.
- [VPC] Google Inc., "VPC Network Overview: Valid Ranges", <<https://cloud.google.com/vpc/docs/vpc#valid-ranges>>.
- [WMH] Wilson, P., Michaelson, G., and G. Huston, "Redesignation of 240/4 from "Future Use" to "Private Use"", Work in Progress, Internet-Draft, draft-wilson-class-e-02, 29 September 2008, <<https://datatracker.ietf.org/doc/html/draft-wilson-class-e-02>>.

Appendix A. Implementation Status

The IPv4 protocol update proposed by this document has already been implemented in a variety of widely-used software platforms. In many cases, implementers were persuaded of the value of the suggestions contained in [FLM] and [WMH].

All known TCP/IP implementations either interoperate properly with packets with sources or destinations in the 240/4 range, or ignore these packets entirely, except FreeBSD, which has support for 240/4 for some purposes while blocking it for others.

A.1. Operating systems

240/4 has been supported for transmitting and receiving ordinary unicast packets in Linux kernels since linux-2.6.25 was released in January 2008. Creating interfaces in the 240/4 range also worked fine using the iproute2 api (as used by the "ip" command) in that release. A kernel patch that allows properly configuring interfaces in the 240/4 range using the busybox ifconfig command was released in linux-4.20 and linux-5.0 in December 2018.

240/4 has been supported as ordinary unicast in the Android mobile operating system since Android 1.5 Cupcake (April 2009, using linux-2.6.27).

240/4 has been supported as ordinary unicast in the OpenWRT router OS since OpenWRT 8.09 (September 2008, using linux-2.6.26). A December 2018 kernel patch that allows properly configuring interfaces in the 240/4 range using the ifconfig command was merged into OpenWRT 19.01, along with two other patches to netifd and BCP38 that improve support for 240/4.

240/4 has been supported as ordinary unicast in Apple's macOS (formerly OS X) operating system and iOS mobile operating system since about 2008.

240/4 has been supported as ordinary unicast in Sun's Solaris operating system since about 2008.

240/4 has been tested to interoperate as ordinary unicast in 2019 in a Cisco router using IOS release 6.5.2.28I, which was also released in 2019. Older and newer releases are also likely to work.

240/4 traffic is blocked by default in Juniper's JUNOS router operating system, but can be enabled with a simple configuration switch.

240/4 traffic is partly supported for local interface assignment in the FreeBSD operating system. However, ICMP and packet forwarding are not supported. Small patches that fully enable FreeBSD support for 240/4 have been tested and are fully interoperable.

240/4 traffic is blocked by default in all versions of the Microsoft Windows operating system. Windows will not assign an interface address in this range, if one is offered by DHCP.

A.2. Other implementations

Routing of subnets in the 240/4 range is fully supported by the Babel routing protocol and by its main implementation, as of 2020 (or earlier).

Routing of subnets in the 240/4 range is supported by the Gobgp routing daemon, as of release 3.0.0 in 2022-03 (or earlier).

A.3. Internet of Things

Popular embedded Internet-of-Things environments such as RIOT and FreeRTOS already support 240/4 as unicast.

Authors' Addresses

Seth David Schoen
IPv4 Unicast Extensions Project
San Francisco, CA
United States of America
Email: schoen@loyalty.org

John Gilmore
IPv4 Unicast Extensions Project
PO Box 170640-rfc
San Francisco, CA 94117-0640
United States of America
Email: gnu@rfc.toad.com

David M. Täht
IPv4 Unicast Extensions Project
Half Moon Bay, CA
United States of America
Email: dave@taht.net

Internet Engineering Task Force	S.D. Schoen
Internet-Draft	J. Gilmore
Updates: 1122, 1812, 3021 (if approved)	D. Täht
Intended status: Standards Track	IPv4 Unicast Extensions Project
Expires: 26 May 2022	M. Karels
	22 November 2021

Unicast Use of the Lowest Address in an IPv4 Subnet
draft-schoen-intarea-unicast-lowest-address-01

Abstract

With ever-increasing pressure to conserve IP address space on the Internet, it makes sense to consider where relatively minor changes can be made to fielded practice to improve numbering efficiency. One such change, proposed by this document, is to increase the number of unicast addresses in each existing subnet, by redefining the use of the lowest-numbered (zeroth) host address in each IPv4 subnet as an ordinary unicast host identifier, instead of as a duplicate segment-directed broadcast address.

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 26 May 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 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	2
1.1. Requirements Language	3
2. Background and Current Standards	3
2.1. Assumptions About the Lowest Host Address by Remote Systems	5
2.2. Multicast Addresses; Point-to-Point Links	6
2.3. Current Limitations on Subnet-Directed Broadcast Addresses	6
2.4. Comparison to IPv6 Behavior	6
3. Change to Interpretation of the Lowest Address	7
3.1. Link-Layer Interaction	7
3.2. Recommendations	8
3.2.1. "Requirements for Internet Hosts -- Communication Layers" [RFC1122]	8
3.2.2. "Requirements for IP Version 4 Routers" [RFC1812]	8
3.3. Example	10
3.4. Compatibility and Interoperability	11
4. IANA Considerations	11
5. Security Considerations	11
6. Acknowledgements	12
7. References	12
7.1. Normative References	12
7.2. Informative References	12
Appendix A. Implementation Status	14
Authors' Addresses	14

1. Introduction

This document provides history and rationale to change the interpretation of the lowest address in each IPv4 subnet from an alternative broadcast address to an ordinary assignable host address, and updates requirements for hosts and routers accordingly. The decision taken in 1989 to reserve two forms instead of one for local IPv4 segment broadcasts is no longer necessary because of the obsolescence and disappearance of the software that motivated it. Unreserving the lowest address provides an optional extra IPv4 host address in every subnet, Internet-wide, alleviating some of the pressure of IPv4 address exhaustion.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Background and Current Standards

IPv4 has long supported several mechanisms for broadcasting communications to every station on a network. One form of broadcast in IPv4 is "segment-directed broadcast" in which a broadcast is addressed to every station on a particular network (identified by its network number). Current standards reserve a huge number of addresses for this: not just one, but two, addresses per subnet, Internet-wide.

The standard broadcast address on a subnet is the address whose "host part" consists of all ones in binary. For example, in a 24-bit subnet that starts at 192.168.3.0, the address 192.168.3.255 is the standard broadcast address. [RFC1122][RFC0894]

In addition to the standard broadcast address, RFC 1122 (October 1989) acknowledged a then-current implementation behavior in "4.2BSD Unix and its derivatives, but not 4.3BSD", whereby those operating systems "use non-standard broadcast address forms, substituting 0 for -1". (Note that there was no standard for IP broadcast when 4.2BSD was released in August 1983, more than a year prior to [RFC0919]. The -1 form was first proposed in [IEN212] in 1982 but was not yet a standard.) According to RFC 1122 and its successors, Internet hosts are expected to "recognize and accept [...] these non-standard broadcast addresses as the destination address of an incoming datagram", and not otherwise use them to identify Internet hosts. RFCs 1812 [RFC1812] (sections 4.2.3.1 and 5.3.5), and 3021 [RFC3021] (sections 2.2, 3.1, and 3.3) reiterate and further expand on this requirement.

This non-standard broadcast address is the address whose "host part" consists of all zeroes. (The quantity of zeroes depends, in present-day terminology, on the applicable subnet mask.) This address is the lowest expressible address within any particular numbered network. Following computer science tradition, it may also be referred to as the "zeroth address" of its respective subnet.

The address triple syntax used in RFC 1122 looks unusual to modern eyes. These triples included the "{ network number, subnet number, host number }". The notation also used two's complement binary notation in referring to a host number "-1" as containing all binary 1-bits. After the widespread adoption of CIDR [RFC4632], network

numbers no longer have an "address class" definition based on their high-order bits, and there is no distinction between a network number and a subnet number (except locally at the router where individual subnets are being routed). Instead, following RFC 4632, IPv4 network addresses are denoted with a dotted-decimal format containing one to four positive 8-bit integers, a slash, and a whole count of the bits in the network number portion, the so-called CIDR notation, reportedly devised by Phil Karn. So for example 192.1.2.0/28 has a 28-bit network number and a 4-bit host number (32 address bits total, minus those 28 bits). All of the bits of that particular host number are zero (because the whole fourth dotted number is 0), and thus the interpretation of this address would be affected by this document. We will use both notations as convenient. Where RFC 1122 and its successors use the terms "0 form" and "-1 form", we may refer respectively to "all-zeros" and "all-ones" host numbers, since every bit in the binary representation of these two host numbers has the value 0 or 1, respectively.

4.2BSD, the operating system to whose behavior RFC 1122 required deference, was the first BSD operating system full release to include TCP/IP support; it was released in 1983. Its successor, 4.3BSD, was released in 1986, which is why RFC 1122 could already confirm that the broadcast behavior had been changed. See [BSDHIST]. RFC 1812 calls the old behavior "obsolete" in 1995, and RFC 3021 reiterates that it is "obsolete" in 2000, although both express the idea that the lowest address must continue to be reserved for historical reasons.

All subsequent operating systems used on the Internet implement the standard all-ones form of the broadcast address and use it by default. Continuing to reserve the non-standard all-zeroes form wastes one IPv4 address per subnet. No known operating system generates IP broadcasts in this format today, and documentation consistently encourages network administrators and software developers to use the standard form. The IPv4 protocol does not benefit from having two different broadcast addresses with the same functionality in every subnet, and the non-standard form has always been reserved primarily for backwards compatibility with systems that have not existed for decades on the Internet.

As IPv4 addresses were not perceived as particularly scarce through the 1980s, the prospect of wasting tens of millions of otherwise assignable addresses in order to achieve backwards compatibility with a particular operating system appeared reasonable. Today, those addresses are clearly valuable and could be put to good use as identifiers of Internet hosts in a time of IPv4 numbering resource exhaustion.

2.1. Assumptions About the Lowest Host Address by Remote Systems

In general, under CIDR [RFC4632], only hosts and routers on a network segment know that segment's netmask with certainty. Remote parts of the Internet are already expected to not make assumptions about whether or not a particular address is a broadcast address, since that determination is already only meaningful for devices connected to the segment containing that address. This document does not change any of these things. Thus, if the behavior of devices on a particular network segment has been updated in accordance with this memo, the lowest address on that segment can already be addressed by hosts elsewhere on the Internet without any changes to their own software.

[RFC1812] noted in section 4.2.3.1 that whether a reserved address is treated specially at all depends on one's vantage point on the network:

[a] router obviously cannot recognize addresses of the form { <Network-prefix>, 0 } if the router has no interface to that network prefix. In that case, the rules of the second bullet [requiring a packet to be discarded] do not apply because, from the point of view of the router, the packet is not an IP broadcast packet.

It also noted in section 5.3.5.2 that

in view of CIDR, such [packets addressed to broadcast addresses of distant networks] appear to be host addresses within the network prefix; we preclude inspection of the host part of such network prefixes.

To the extent that software continues to make assumptions about IP network classes today, it is out of compliance with RFC 4632. Ever since the adoption of CIDR in RFC 1519, it has been unknowable whether or to what extent the remote network would internally aggregate or deaggregate routes that were visible elsewhere on the Internet. Therefore, Internet hosts and routers MUST NOT assume that an IPv4 address on a remote network, other than 0.0.0.0, is invalid, unroutable, or inaccessible merely because it ends with a particular number of zeroes. In keeping with the Internet's end-to-end principle, decisions about possible invalidity of otherwise routable addresses belong as close to the endpoints as possible.

2.2. Multicast Addresses; Point-to-Point Links

Multicast addresses, as defined by RFC 1112 [RFC1112], do not have a network part and host part, nor do they have a netmask or CIDR prefix length. IPv4 multicast addresses, except 224.0.0.0 (which is "guaranteed not to be assigned to any group" by RFC 1112), could always end with any number of zeroes, and have never had any form of directed broadcast address.

[RFC3021], section 2.1, standardized that, in a point-to-point link using a 31-bit netmask, the all-zero and all-one forms of the host-part of the address MUST both be treated as unicast ("host") addresses.

The present document does not change the interpretation of multicast addresses or 31-bit subnet addresses in any way.

2.3. Current Limitations on Subnet-Directed Broadcast Addresses

Sending packets to a subnet-directed broadcast address is still generally useful in today's Internet, but only for nodes attached directly to that subnet. [RFC2644] discouraged routers from forwarding such packets, to reduce their use in amplifying denial-of-service attacks, so they often cannot be received when sent from distant hosts. Many hosts today ignore ICMP packets sent as broadcasts, so a directed broadcast ping is no longer a reliable means of enumerating all hosts attached to a network. As Informational [RFC6250] notes, "broadcast can only be relied on within a link".

2.4. Comparison to IPv6 Behavior

In IPv6 there are no reserved per-segment broadcast addresses (or, indeed, any broadcast addresses whatsoever). Instead, IPv6 hosts can address all hosts on a network segment by using the link-local multicast group address ff02::1 [RFC4291], which, for example, produces a multicast Ethernet frame when transmitted over an Ethernet-like link [RFC2464]. The lowest address on a subnet is, however, reserved in IPv6 by Section 2.6.1 of RFC 4291 [RFC4291] for the Subnet-Router address (a means of addressing "any router" on an indicated subnet).

3. Change to Interpretation of the Lowest Address

The purpose of this document is to designate the all-zeros address in each subnet as a unicast address. All such addresses are now available for general non-broadcast use, treated identically to all host addresses in the subnet besides the "all-ones" broadcast address. This document therefore eliminates an element of the IPv4 protocol's historical adaptation to 4.2BSD's behavior. All hosts SHOULD continue to recognize and accept only the all-ones form of the IPv4 subnet broadcast address.

Host software that intends to transmit a segment-directed broadcast packet in an IPv4 network MUST use only the all-ones form as the destination address of the packet.

An IPv4 datagram containing a source or destination that is equal to the all-zeroes form of the local broadcast address SHOULD be treated, by both hosts and routers, as a normal unicast datagram; it SHOULD NOT be treated as a local broadcast datagram.

Host software SHOULD allow a network interface to be configured with the lowest address on a subnet. A host with such an address configured MUST use this assigned address as a source address for datagrams just as it would with any other assigned interface address, and MUST recognize a datagram sent to that address as addressed to itself. Host software SHOULD be capable of generating unicast packets to the lowest address on a subnet when so requested by an application, and MUST encapsulate such packets into link-layer unicast frames when transmitted on a link layer that distinguishes unicast and broadcast.

Clients for autoconfiguration mechanisms such as DHCP [RFC2131] SHOULD accept a lease or assignment of the lowest address whenever the underlying operating system is capable of accepting it. Servers for these mechanisms SHOULD assign this address when so configured. The network operator of each subnet retains the discretion to number hosts on that subnet with, or without, the use of the lowest address, based on local conditions.

3.1. Link-Layer Interaction

The link layer always indicates to the IP layer whether or not a datagram was transmitted as a broadcast at the link layer. Hosts MUST continue to follow the RFC 1122 rule about link-layer broadcast indications:

A host SHOULD silently discard a datagram that is received via a link-layer broadcast [...] but does not specify an IP multicast or broadcast destination address.

This rule is, among other things, intended to avoid broadcast storms. This document now defines the lowest address as a non-broadcast address. Therefore, a host SHOULD silently discard a datagram received via a link-layer broadcast whose destination address is the lowest IPv4 address in a subnet. This is true even if the interface on which the host received that datagram uses the lowest address as a unicast IPv4 address.

3.2. Recommendations

The considerations presented in this document affect other published work. This section details the updates made to other documents.

3.2.1. "Requirements for Internet Hosts -- Communication Layers" [RFC1122]

The new section numbered 3.2.1.3 (h) which was added by RFC 3021 is replaced with:

(h) { <Network-number>, <Subnet-number>, 0 }

An ordinary unicast ("host") address in the subnet. May be used as either a source or destination address. If a link-level broadcast packet is received with this address (or any other unicast address) as its destination, it MUST be silently discarded. Such a packet may be sent by long-obsolete hosts on the local network.

In applications using CIDR notation [RFC4632], this address, or any other address in the subnet, may also be used together with a prefix length to refer to the entire subnet.

3.2.2. "Requirements for IP Version 4 Routers" [RFC1812]

The new section (numbered 4.2.2.11 (f)) added by RFC 3021 is replaced by:

(f) { <Network-prefix>, 0 }

An ordinary unicast ("host") address in the subnet. May be used as either a source or destination address. If a link-layer broadcast packet is received with this address (or any other unicast address) as its destination, it MUST be silently discarded. Such a packet may be sent by long-obsolete hosts on the local network.

In applications using CIDR notation [RFC4632], this address, or any other address in the subnet, may also be used together with a prefix length to refer to the entire subnet.

The first paragraph on page 49 (which appears after section 4.2.2.11 (e) in the original RFC 1812, or after section 4.2.2.11 (f) in RFC 1812 as modified by RFC 3021) is changed from this original text

IP addresses are not permitted to have the value 0 or -1 for the <Host-number> or <Network-prefix> fields except in the special cases listed above. This implies that each of these fields will be at least two bits long.

DISCUSSION

Previous versions of this document also noted that subnet numbers must be neither 0 nor -1, and must be at least two bits in length. In a CIDR world, the subnet number is clearly an extension of the network prefix and cannot be interpreted without the remainder of the prefix. This restriction of subnet numbers is therefore meaningless in view of CIDR and may be safely ignored.

to this new text

Unicast IP addresses are permitted to have the value 0 for the <Host-number> field, and may have the value -1 in the special cases listed above. There is no requirement that the <Host-number> field be any particular length. In some cases using CIDR notation, a host may be designated with a /32 suffix (e.g. 192.0.2.34/32), indicating that the specific host rather than its subnet is being described.

DISCUSSION

Previous versions of this document also noted that subnet numbers must be neither 0 nor -1, and must be at least two bits in length. Other versions required that <Network-prefix> fields must be neither 0 nor -1, and must be at least two bits long.

Now that the Internet has fully transitioned to CIDR routing, there are no original classful <Network-number>s to be distinguished from <Subnet-numbers>. Each address only has a <Network-prefix> based on its network mask (or equivalently, the CIDR suffix specifying how many bits are in the <Network-prefix>). The former restrictions on subnet numbers and their sizes are meaningless in view of CIDR and are hereby repealed. For example, a route to 0.0.0.0/6 or even 0.0.0.0/1 is a viable CIDR route (for the aggregation of the blocks 0/8, 1/8, 2/8, and 3/8; or for the entire lower half of the IPv4 address space) and should not be considered invalid. 0.0.0.0/0 is standardized to mean "all unicast IPv4 addresses", e.g. in a default route, by section 5.1 of [RFC4632], which MUST also continue to work.

Sections 4.2.3.1 (2) and (4) are replaced with:

(2) SHOULD silently discard on receipt (i.e., do not even deliver to applications in the router) any packet addressed to 0.0.0.0. If these packets are not silently discarded, they MUST be treated as IP broadcasts (see Section [5.3.5]). There MAY be a configuration option to allow receipt of these packets. This option SHOULD default to discarding them.

A packet addressed to { <Network-prefix>, 0 } is an ordinary unicast packet, and MUST be treated as such.

(4) SHOULD NOT originate datagrams addressed to 0.0.0.0. SHOULD allow for the generation of datagrams addressed to {<Network-prefix>, 0 } since that is now defined as an ordinary unicast address.

3.3. Example

The only IPv4 broadcast address for 192.168.42.0/24 is 192.168.42.255 (the all-ones or "-1" host number). 192.168.42.0 (the all-zeroes or "0" host number) was formerly a second broadcast address on that subnet, but is now a unicast address.

The fact that the address identifier 192.168.42.0 can refer to both a network and a specific host 192.168.42.0 is not unusual. Similarly, referring to a subnet as 192.168.42.0/24 and configuring a particular interface on that subnet as 192.168.42.0/24 is also not unusual. Computer scientists normally count all sorts of things starting at the zeroth (lowest) element in a sequence.[EWD831] For example, the initial element in an array is likely to be stored at a memory address equal to the memory address of the array itself.[ARRAY] Similarly, IPv4 hosts in a subnet MAY be enumerated starting with an address that matches the address of the subnet itself.

Similarly, the only IPv4 broadcast address for the subnet 192.168.42.96/28 is 192.168.42.111. The address 192.168.42.96 MAY be assigned to an individual host on this network.

3.4. Compatibility and Interoperability

Many deployed systems follow older Internet standards in not allowing the lowest address in a network to be assigned or used as a source or destination address. Assigning this address to a host may thus make it inaccessible by some devices on its local network segment. Network operators considering assigning this address to a host should investigate their own network environments to determine whether their interoperability requirements will be met. Interoperability with these addresses is likely to improve over time, following the publication of this document.

Prior standards required hosts and routers to ignore, and to refrain from generating, non-broadcast datagrams from or to the lowest address. So when a single network contains a device that has been assigned the lowest address as specified by this document, along with one or more devices that follow the traditional behavior, the traditional devices will not be able to communicate with the lowest-address device at all. Other sorts of malfunctions are unlikely, because the former standards (RFC 1122) required traditional hosts to drop any unicast packet addressed to the secondary broadcast address that they implemented at the lowest address.

4. IANA Considerations

This memo includes no request to IANA.

5. Security Considerations

The behavior change specified by this document could produce security concerns where two devices, or two different pieces of software on a single host, or a software application and a human user, follow divergent interpretations of the lowest address on a network. For example, this could lead to errors in the specification or enforcement of rules about Internet hosts' connectivity to one another, or their right to access resources.

Firewall rules that assume that the lowest address on a subnet cannot be addressed SHOULD be updated to take into account that it can be addressed, so as to avoid either unintentionally allowing or unintentionally forbidding connections involving it. Other security, monitoring, or logging systems that treat the lowest address as an inaccessible bogon address SHOULD likewise be updated.

Host software SHOULD make the distinction between lowest-address (considered individually) and subnet (considered as a group) clear to users, where this distinction is relevant and could be a subject of confusion.

6. Acknowledgements

This document directly builds on prior work by Dave Täht and John Gilmore as part of the IPv4 Unicast Extensions Project.

7. References

7.1. Normative References

- [RFC0919] Mogul, J., "Broadcasting Internet Datagrams", STD 5, RFC 919, DOI 10.17487/RFC0919, October 1984, <<https://www.rfc-editor.org/info/rfc919>>.
- [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>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<https://www.rfc-editor.org/info/rfc1812>>.
- [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>>.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <<https://www.rfc-editor.org/info/rfc4632>>.

7.2. Informative References

- [ARRAY] Wikipedia, "C Programming Language: Array-pointer interchangeability", <[https://en.wikipedia.org/wiki/C_\(programming_language\)#Array%E2%80%93pointer_interchangeability](https://en.wikipedia.org/wiki/C_(programming_language)#Array%E2%80%93pointer_interchangeability)>.
- [BSDHIST] Wikipedia, "History of the Berkeley Software Distribution", <https://en.wikipedia.org/wiki/History_of_the_Berkeley_Software_Distribution>.

- [EWD831] Dijkstra, E.W., "Why Numbering Should Start at Zero", August 1982, <<https://www.cs.utexas.edu/users/EWD/transcriptions/EWD08xx/EWD831.html>>.
- [IEN212] Gurwitz, R. and R. Hinden, "IP - Local Area Network Addressing Issues", IEN 212, September 1982, <<https://www.postel.org/ien/pdf/ien212.pdf>>.
- [RFC0894] Hornig, C., "A Standard for the Transmission of IP Datagrams over Ethernet Networks", STD 41, RFC 894, DOI 10.17487/RFC0894, April 1984, <<https://www.rfc-editor.org/info/rfc894>>.
- [RFC1112] Deering, S., "Host extensions for IP multicasting", STD 5, RFC 1112, DOI 10.17487/RFC1112, August 1989, <<https://www.rfc-editor.org/info/rfc1112>>.
- [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>>.
- [RFC2644] Senie, D., "Changing the Default for Directed Broadcasts in Routers", BCP 34, RFC 2644, DOI 10.17487/RFC2644, August 1999, <<https://www.rfc-editor.org/info/rfc2644>>.
- [RFC3021] Retana, A., White, R., Fuller, V., and D. McPherson, "Using 31-Bit Prefixes on IPv4 Point-to-Point Links", RFC 3021, DOI 10.17487/RFC3021, December 2000, <<https://www.rfc-editor.org/info/rfc3021>>.
- [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>>.
- [RFC6250] Thaler, D., "Evolution of the IP Model", RFC 6250, DOI 10.17487/RFC6250, May 2011, <<https://www.rfc-editor.org/info/rfc6250>>.

Appendix A. Implementation Status

The behavior specified in this document has been implemented by the Linux kernel since version 5.14, released in August 2021. It is also implemented in a patch currently present in the development version of the FreeBSD kernel. These two implementations interoperate successfully.

To our knowledge, the behavior specified by this document is not currently the default in any other TCP/IP implementation.

Authors' Addresses

Seth David Schoen
IPv4 Unicast Extensions Project
San Francisco, CA
United States of America

Email: schoen@loyalty.org

John Gilmore
IPv4 Unicast Extensions Project
PO Box 170640-rfc
San Francisco, CA 94117-0640
United States of America

Email: gnu@rfc.toad.com

David M. Täht
IPv4 Unicast Extensions Project
Half Moon Bay, CA
United States of America

Email: dave@taht.net

Michael J. Karels
Eden Prairie, MN
United States of America

Email: rfc@karels.net

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: 13 October 2022

H. Song
Futurewei Technologies
11 April 2022

Short Hierarchical IP Addresses for Edge Networks
draft-song-ship-edge-03

Abstract

To mitigate the IPv6 header overhead and improve the scalability and performance in edge networks, this draft proposes to use short hierarchical IP addresses excluding the network prefix within edge networks. An edge network can be further organized into a hierarchical architecture containing one or more levels of networks. While each end node only needs to keep a short address suffix as its identifier, the border routers for each hierarchical level are responsible for address augmenting and pruning when a packet leaves or enter a lower level network. Specifically, the top-level border routers of an edge network convert the internal IP header to and from the standard IPv6 header. This draft presents an incrementally deployable scheme allowing packet header to be effectively compressed in edge networks without affecting the network interoperability. Simplifying both network data plane and control plane, the SHIP architecture is suitable for any types of edge networks, especially when low latency, high performance, and high bandwidth efficiency are required.

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 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

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 13 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	3
2. Short Hierarchical IP Address (SHIP) in Edge Networks	4
2.1. Edge Network Hierarchy	4
2.2. Address Fields	5
2.3. Router Roles and Function	6
3. Deployment and Interoperability Consideration	9
3.1. Control Plane	9
3.2. Data Plane	11
3.3. Using NAT for the edge network	11
3.4. Extension Beyond IPv6	12
4. Comparison with Existing IPv6 Header Compression Schemes	12
5. Use Cases	12
6. Evaluation	13
7. Security Considerations	13
8. IANA Considerations	13
9. Acknowledgments	13
10. References	13
10.1. Normative References	13
10.2. Informative References	13
Author's Address	14

1. Introduction

Internet of Things (IoT) and 5G introduce to the Internet a huge number of addressable entities (e.g., sensors, machines, vehicles, and robots). The transition to IPv6 is inevitable. While the 128-bit address of IPv6 was considered large enough and future-proof, the long IP addresses inflate the packet header size. 80% of a basic IPv6 header is consumed by addresses.

In IoT networks, thing-to-thing communication through wireless connections is dominant, which presents several distinct characteristics. (1) The communication pattern is often frequent short-message exchanges (e.g., industry robots and networked vehicles). (2) The communication is usually energy sensitive (e.g., battery-powered sensors). (3) The communication often requires low latency (e.g., industry control). (4) The precious wireless channels demand high bandwidth utilization (e.g., ZigBee, Bluetooth, Wi-Fi, and 5G). These characteristics render a large header overhead unfavorable and even prohibitive.

The address overhead also takes its toll on Data Center Networks (DCN), especially when large scale containers are deployed, the east-west traffic is dominant, and the prevailing communications are comprised of short messages (e.g., key-value pairs) and conducted through virtual switches.

In IoT and DCN, since most communications happen between adjacent and related entities, it is a good practice to locally confine communication, computing, and storage due to performance, efficiency, and security considerations, as advocated by Edge Computing. Such a communication pattern provides an opportunity to mitigate the IPv6 header overhead problem due to the long addresses.

When an IPv6 address block is allocated to an edge network, all the entities in the edge network share the same address prefix. When these entities communicate with each other, they can ignore the common prefix. In fact, they do not even need to know the common prefix. Only when they need to communicate with entities outside of the edge network, the full addresses are needed. Even in this case, the entities in the edge network still do not need to know the prefix. It is sufficient for the gateway routers at the network border to manipulate the addresses (i.e., augmenting or pruning the address) to meet the addressing requirement.

Following this line of thought, an edge network can be further partitioned into multiple hierarchical levels, which support flexible sub-networking. The result is that an end entity needs to maintain an even shorter address as its identifier. For communication crossing network levels, the address manipulation is done at each gateway router on the path recursively.

2. Short Hierarchical IP Address (SHIP) in Edge Networks

2.1. Edge Network Hierarchy

In this draft, we define an edge network as a stub network which does not support traffic transit service. The stub network is assigned an IPv6 address block. In this sense, a data center network in cloud can also be considered as an edge network. An edge network usually falls under a single network administration domain.

The address block assigned to an edge network is identified by a prefix P with the length of $L < 128$ bits. The remaining $S = 128 - L$ bits can be used to assign addresses to the entities in this network. A key observation is: the entities in this network do not need to be aware of P 's length and value at all. We can further partition the edge network into multiple hierarchical levels, making a tree architecture. The root represents the entire edge network. Each other node represents a lower level network occupying a sub address space owned by its parent node. A leaf node represents a lowest level network. We name the root level network the L_0 network. Its children are all L_1 networks, and so on so forth. In other words, the network level is the depth of the corresponding node in the tree.

The network hierarchy partitions the S -bit address into multiple sections. Assume an entity is in an L_n network. The S -bit address is partitioned into $n+1$ sections. The entity only needs to keep the last section of the S -bit address as its ID. The gateway routers for each level of network maintain one section of the S -bit address. Specifically, the gateway routers of L_i ($i > 0$) keep the i -th section of the S -bit address, and the gateway routers of L_0 keep the assigned IPv6 address block prefix P .

Figure 1 shows an edge network example, in which are three network levels. The edge network A is assigned a 96-bit IPv6 address prefix (2001:0db8:ac10:fe01::0001), which means it owns a 32-bit address space. In this space, two L_1 networks are created: B with a 16-bit prefix (0xaaaa) and C with a 24-bit prefix (0xcccccc). Note that the prefixes at the same level must not overlap in order to guarantee entities in the edge network are uniquely addressable. Network B contains two entities x and y , and Network C contains one entity z . In network B, an L_2 network C is further created with a 8-bit prefix

(0xbb). In this example, an entity in C or D (e.g., m and z) only need to own a 8-bit address, an entity in B but not in D (e.g., x and y) needs to own a 16-bit address, and an entity in A but not in B and C needs to own a 32-bit address. In this way, each entity in A still logically owns a unique IPv6 address (e.g., the IPv6 address of the entity m in D with ID of 5 is 2001:0db8:ac10:fe01::0001:aaaa:bb05), although the entity m is only aware of its local ID (0x05).

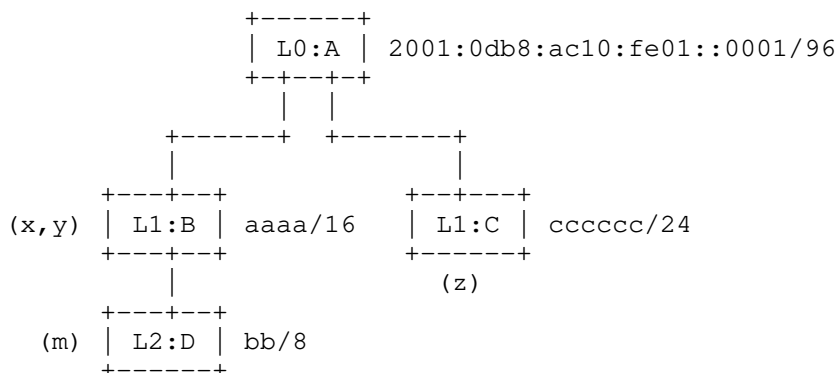


Figure 1: A Hierarchical Edge Network Example

2.2. Address Fields

The edge networks adopting the short and variable size address scheme need a new type of IP header, which is referred as IPvn in this draft. Apart from the IP version, the major difference between IPvn and IPv6 headers is the address fields. IPvn replaces IPv6's 128-bit source address field and 128-bit destination address field with the four fields shown in Figure 2.

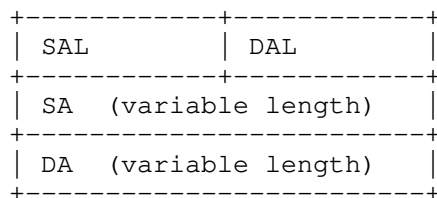


Figure 2: IPvn Address Fields

The Source Address Length (SAL) and the Destination Address Length (DAL) fields have fixed length, while the Source Address (SA) and the Destination Address (DA) fields are of variable length. To simplify

the implementation, SA and DA are preferred to be byte-aligned. It is possible to define the length of address in the unit of byte, nibble, or bit. Each has its own pros and cons. The unit of byte can help reduce the size of the SAL/DAL but results in coarse network granularity which might be inefficient in address allocation. For example, a 3-bit SAL/DAL is enough to encode 8 possible address lengths (one to eight bytes) for networks. In this design, each higher level network's address space expands 256 times. On the other extreme, the unit of bit allows fine network granularity but requires more space for SAL/DAL. For example, 6-bit SAL and DAL can support an address length up to 64 bits (8 bytes) and each higher level network is only twice larger.

With a few bits, it is also possible to design a more sophisticated encoding scheme that supports variable address length steps and adapts to the ideal network sizes at different levels.

Assuming SA and DA are 2 bytes each, and SAL and DAL are 4 bits each, the address fields are only 5 bytes in total. Comparing to IPv6, the size of the address fields is reduced by 84%.

2.3. Router Roles and Function

In the edge network hierarchy, each network has one or more Level Gateway Routers (LGR) which are responsible for forwarding packets in or out of this network. The LGRs are the only interface between a network and its parent network.

A network can be in a single L2 domain, which means all the entities in this network (excluding those in its child networks) and all the network devices (including the LGRs to the parent network and the child networks) are L2 reachable. A network can also be a pure L3 network in which no L2 device is allowed. Each entity in a network is directly connected to either an LGR or some internal routers named Intra-Level Router (ILR) which is solely responsible for packet forwarding within the network. In this case, the entities need to partially participate in the routing process (e.g., advertising its address).

The scale of an intra-level network can be used to guide the L2/L3 selection. Small networks prefer the L2-based solution and large networks prefer the L3-based solution. In the higher level networks (e.g., closer to the top level network or the tree root), since the number of entities is usually small, it is free to choose between L2 or L3-based solution. The leaf level networks are usually L2-based for simplicity.

Unlike in IPv4 and IPv6 networks, the address related fields in IPvn header can be modified by LGRs. An LGR of a network keeps a prefix that can augment the SAs from this network to an address outside of this network. If an LGR needs to forward an internal packet outside (i.e., $DAL > SAL$), it augments the packet's SA and updates its SAL accordingly. Reversely, if an LGR receives a packet from the parent network destined for the child network for which it serves as a gateway (i.e., the parent network prefix matches the DA's prefix), it strips off the parent network prefix from the packet's DA and updates its DAL accordingly.

In contrast, within an L3-based level network, ILRs do not modify the address fields. An ILR can decide the packet forwarding direction by examining the DAL. If $DAL > SAL$, the packet needs to be forwarded to an LGR of this network; otherwise, the packet needs to be forwarded within the current network, and possibly into a lower-level child network.

An LGR of the top-level network (i.e., the L0 network) is special. In addition to the address manipulation, it is also responsible for converting the IPvn header to and from the standard IPv6 header to support the Internet interoperability. We name such a router IP Translator (IPT).

We use the edge network shown in Figure 1 to illustrate some packet forwarding examples. The details for the involved entities are summarized in Figure 3. In the IPvn packet header, we use 4 bits to encode the address length. In particular, 0b0000 is used to indicate the address is 16 bytes long (i.e., a complete IPv6 address).

Entity	ID	Length	Level	Network	Prefix
x	0x0001	2bytes	1	B	0xaaaa/16
y	0x0002				
z	0x01	1byte	1	C	0xcccccc/24
m	0x08	1byte	2	D	0xbb/8

Figure 3: Entity Address Configuration

The first example in Figure 4 shows how packets are forwarded from x to y within the same network B. In this case, the source address and destination address have the same length. The packets only pass through an ILR which does not change the address fields.

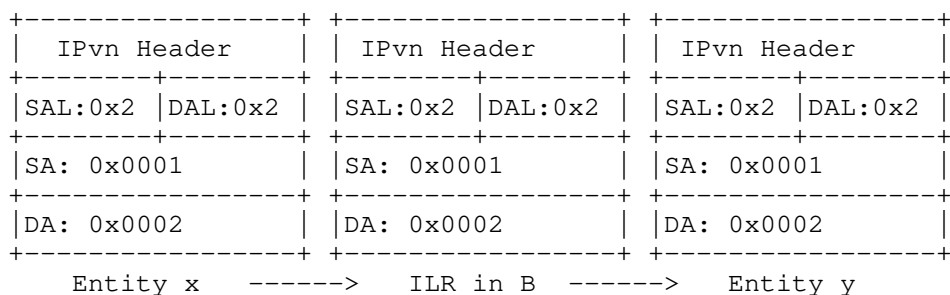


Figure 4: Forward within a network level in the edge

The second example in Figure 5 shows how packets are forwarded from x in B to z in C. At LGR of B, the source address is augmented, and at the LGR of C, the destination address is pruned. Since x and z's nearest common ancestor network is A, so the packets never need to leave network A, so A's prefix is oblivious throughout the communication.

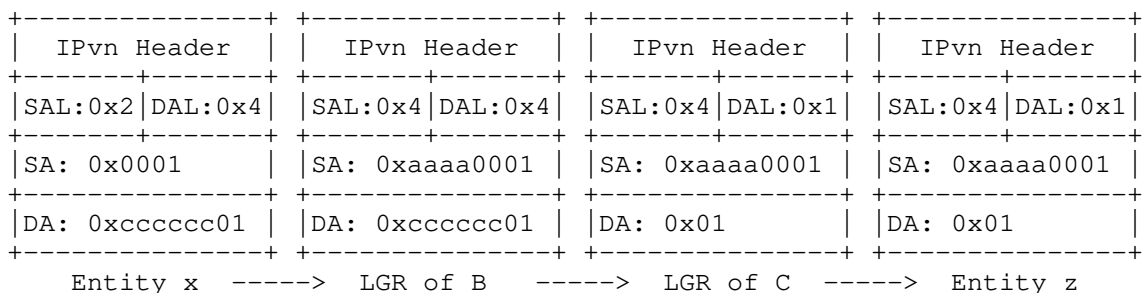


Figure 5: Forward to another network in the edge

The last example in Figure 6 shows how packets are forwarded from x in B to a host in IPv6 domain. In the IPT of A, the IPvn header is converted to an IPv6 header.

IPvn Header	IPvn Header	IPv6 Header	IPv6 Header
SAL:0x2 DAL:0x0	SAL:0x4 DAL:0x0	SA: 2001:0db8 ac10:fe01: 0000:0001: aaaa:0001	SA: 2001:0db8 ac10:fe01: 0000:0001: aaaa:0001
SA: 0x0001	SA: 0xaaaa0001	DA: 2001:0db8: 85a3:0000: 0000:8a2e: 0370:7334	DA: 2001:0db8: 85a3:0000: 0000:8a2e: 0370:7334

Entity x -----> LGR of B -----> IPT of A -----> Entity n

Figure 6: Forward out of the edge network

3. Deployment and Interoperability Consideration

3.1. Control Plane

Within the edge networks where IPvn is applied, all the control plane functions and protocols need to be modified or redesigned due to the hierarchical network architecture of IPvn. Fortunately, the updates are often incremental and the results are usually simpler than their counterparts in IPv4 and IPv6. We briefly discuss a few essential protocols that enable the operation of IPvn.

DHCP: An entity can be manually configured or dynamically acquire its address when booting up. Each network in the edge network hierarchy may contain a Dynamic Host Configuration Protocol (DHCP) server responsible for assigning addresses (i.e., IDs) to the entities in the same network. The protocol is almost identical to that for IPv4 and IPv6, except that the assigned address length is adaptive to the allocated network size.

DNS: For an entity to acquire the address of a peer entity in order to initiate a communication, Domain Name System (DNS) is the prominent approach but with a new service model. Any network in the hierarchy can provide name service. Each entity is configured with the address of the closest DNS server on the path to the root network. The hierarchical network architecture allows a scoped domain name service. That is, a name registered in a network is only valid in this network and its child networks. It is possible that a same name is registered in two networks and one network is the other's ancestor. Such name conflict is not a bug but a

feature for name reuse, which is transparent to the name query process. In this case, the name resolved from the closer DNS server is used.

Each network may contain a DNS server (the notation is only logical. The actual implementation may follow the same hierarchical and distributed architecture of today's DNS). Each DNS server knows the nearest DNS server in a higher level network and the nearest DNS servers in lower level networks. This essentially organizes the DNS servers in the same tree structure as the hierarchical network. Each named entity in a network is registered with the DNS server that covers its scope, which is basically a subtree.

We have several methods to populate the name to support the scoped name queries, each with different storage and performance trade-off: 1) register the name in all the DNS servers in its scope (i.e., all the subtree nodes); 2) recursively register the name in every parent DNS server until the scope root; and 3) register the name only in the DNS server in its scope root. The address for a name returned by a DNS server is on a "need-to-know" basis. In a network, if the address's prefix matches the query's address prefix, the prefix is removed. This can be easily done by the original or the relay DNS servers. If a query cannot be resolved by the DNS server in the L0 network, the query, after the IP protocol translation is done, exits the IPvn domain and enters into the IPv4/IPv6 domain to a public DNS server. When the response comes back and enters the edge network, the result can be cached by the DNS servers on the path.

ARP: In a L2-based network, the operation of Address Resolution Protocol (ARP) or Neighbor Discovery Protocol (NDP) is almost identical to that for IPv4 and IPv6. In an L2-based network, each immediate entity should be configured with a default gateway address to its parent network. If no default gateway is configured, a network LGR should be configured as an ARP proxy to respond to all internal ARP requests for addresses out of the network. Similarly, the LGRs to any child network of this network are also needed to be configured as ARP proxy to response all ARP requests for addresses in that network. Due to the multi-homing gateway routers, an ARP request may receive multiple responses. It is up to the requester to determine which one to cache.

Routing Protocol: The entire edge network may belong to a single AS, so the interior gateway routing protocols (IGP) such as OSPF and IS-IS can be used. Other child networks in this network can be considered OSPF stub areas or IS-IS levels. A simpler way is that each network run an independent instance of OSPF or IS-IS.

Specially, an LGR at a network border runs two OSPF/IS-IS instances: one for the upper-level network and the other for the lower-level network. The hierarchical architecture solves the routing protocol scalability issue, and simplifies the protocol implementation by removing unnecessary features. The clean routing scope helps to secure the infrastructure and troubleshoot the networks.

3.2. Data Plane

IPvn Socket for End Entities: To enable IPvn as a new network layer protocol in end entities, we need to add the protocol implementation in the OS Kernel and allow applications to invoke the socket API using the address family parameter AF_INETN. The L4-L7 protocol stack and the application logic remains the same, allowing direct communication between entities in IPvn domain and in IPv4/IPv6 domain.

Forwarding Table Lookups in Networks: The short hierarchical address simplifies the router forwarding table structure in L3-based networks. A forwarding table only contains the addresses to local entities and the prefixes to the child networks. Since there is no nested prefixes, the Longest Prefix Matching (LPM) is not necessary. The small number of unique prefix lengths allows the prefixes to be grouped on lengths and each group to be implemented as a hash table. A lookup can search all the hash tables in parallel, and at most one table can result a positive match. This design avoids the use of expensive TCAM or other complex trie-based algorithms.

An LGR between an L_i network and an $L_{(i+1)}$ network has two types of interfaces: one faces the L_i network and the other faces the $L_{(i+1)}$ network. One LGR may serve more than one $L_{(i+1)}$ network. Hence, an LGR may contain multiple logical forwarding tables, with each for a network. For a packet in LGR, once its target network is determined and the address related fields are processed, the proper forwarding table can be searched.

3.3. Using NAT for the edge network

To expand the address space of the edge network, the IPT of the edge network can also support functions similar to NAT. In this case, the edge network is assigned one or more public IPv4/IPv6 addresses. The entities in IPvn domain use private addresses. The IPT maintains the mapping table between the private address and public address.

3.4. Extension Beyond IPv6

Although the motivation of this draft is to support shorter address (i.e., smaller L3 header overhead) in edge networks, it is worth noting that the scheme allows the addresses to be extended to arbitrary length, even longer than 128bits. In that case, the address space of the IPv6 network can be greater than that of IPv4 and the entire IPv4 network can be considered an edge network of the IPv6 network. This scenario should be considered when specifying the address fields of IPv6.

4. Comparison with Existing IPv6 Header Compression Schemes

IPv6 header compression schemes have been specified for some particular low power IoT networks such as 6LoWPAN ([RFC6282]) and LPWAN ([RFC8724]). These networks feature low data rate and are insensitive to latency, however, due to the low power constraint, they are extremely sensitive to bandwidth efficiency. Therefore, they adopt the context-based compression schemes which, while needing extra storage and computation, can reduce the header overhead to the utmost extent.

In contrast, SHIP is context-less and independent to the edge network type. Hence, SHIP is free from the packet-based compression/decompression process and the context maintenance, making it suitable for high bandwidth and low latency communications. Also, SHIP provides a hierarchical network architecture which allows better network manageability and isolation.

The current proposal only concerns the address part of the IPv6 packet header. In edge networks and for particular applications, the context-less field eliding and reduction on the other non-essential IPv6 header fields are possible to further reduce the header overhead while maintaining the high performance.

5. Use Cases

Below is a list of potential use cases in addition to the DCN discussed in Section 1 which can appreciate the unique property of SHIP.

- * A subset of mMTC UEs needs low latency and high bandwidth and are sensitive to power consumption. For example, the V2X UEs and AR/VR UEs (e.g., advanced handset or 5G enabled headset) are constrained by battery power but demand for high bandwidth and low latency.

- * LEO satellite constellations and communication also require high bandwidth efficiency and low latency.

6. Evaluation

TBD

7. Security Considerations

The SHIP addressing scheme and architecture allow a securer edge network. The IPTs and LGRs naturally support the access control.

8. IANA Considerations

The proposal requires to use a new IP version and define a new IP header which can be converted to/from an equivalent IPv6 header.

9. Acknowledgments

We acknowledge the technical contributions, suggestions and comments from Yingzhe Qu, Zhaobo Zhang, James Guichard, Toerless Eckert, Stewart Bryant, Michael McBride, Adnan Rashid, Alexander Pelov, Michael Richardson, Pascal Thubert, Uma Chunduri, Kerry Lynn, and many others.

10. References

10.1. 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>>.
- [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>>.

10.2. Informative References

- [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>>.

[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>>.

Author's Address

Haoyu Song
Futurewei Technologies
Santa Clara,
United States of America
Email: haoyu.song@futurewei.com