6Lo Working Group
Internet-Draft
Intended status: Experimental
Expires: 17 June 2022

G. Li
D. Lou
L. Iannone
Huawei
P. Liu
R. Long
China Mobile

14 December 2021

Native Short Addressing for Low power and Lossy Networks Expansion
draft-li-6lo-native-short-address-01

Abstract

This document specifies mechanisms of NSA (Native Short Address) that enables IP packet transmission over links where the transmission of a full length address may not be desirable. This document focuses on carrying IP packets across a LLN (Low power and Lossy Network), in which the topology is relatively static where nodes' location is fixed and the connection between nodes is rather stable. The changes in the logical topology are only caused by non-frequent disconnection in the link due to some reasons. The specifications details NSA address allocation, forwarding mechanism, header format design, including length-variable fields, and IPv6 interconnection support.

Status of This Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 17 June 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

There is an ongoing massive expansion of the network edge that is driven by the "Internet of Things" (IoT) technology, especially over low-power links which often, in the past, did not support IP packet transmission.

Particularly driven by the requirements stemming from Industry 4.0 and Smart City deployments, more and more devices/things are connected to the Internet. Sensors in plants/parking bays/mines, temperature/humidity/flash sensors in museums, normally are located in a fixed position and are networked by low power and lossy links even in hard wired networks. Comparing with traditional scenarios, scalability of the (edge) network along with lower power consumption are key technical requirements. Moreover, large-scale Low power

Lossy Networks (LLNs) are expected to be able to carry IPv6 packets over their links, together with an efficient access to native IPv6 domains.

The work in [SIXLOWPAN]/[SIXLO]/[LPWAN] Working Groups address many fundamental issues for those type of deployments. Those deployments can be considered an instantiation of what [RFC8799] defines as "limited domains". For instance, the 6lowpan compression technology ([RFC4944] and [RFC6282]) addresses the problem of IPv6 transmission over LLNs, making it possible to interconnect IPv6-based IoT networks and the Internet. [RFC8138] introduces a framework for implementing multi-hop routing on an LLN using a compressed routing header, which works also with RPL (Routing Protocol for LLNs [RFC6550]). This technique enables the ability to forward IPv6 packets within the domain without the need of decompression. In addition, SCHC (Generic Framework for Static Context Header Compression and Fragmentation [RFC8724]) enables even more compression by using a common static context.

Although aforementioned technologies are suitable in general for all IoT scenarios, there could be more simplified solutions for those scenarios and applications with static network topology and stable network connections leveraging on wired technologies [I-D.ietf-6lo-use-cases] (e.g. smart building, smart parking, etc.). In those kind of deployments, topologies are planned in advance and well provisioned, with sensor nodes usually fixed in specific locations. This draft presents a topology based addressing mechanism with shorter packet header and simpler forwarding rules for those static IoT networks.

The specifications in this document leverage on previous work, namely using the dispatch type field ([RFC4944], [RFC8025]) that allows to accommodate the proposed address format. This means that except the addresses (source and destination) the other fields of the header will be compressed mostly according to LPWAN_IPHC. The proposed addressing is independent of Unique Local Addresses [RFC4193], which has a dependency on specific link-layer conventions [RFC6282]. It is also different from stateful address allocation that requires all nodes to obtain addresses from a centralized DHCP server, which leads to long network startup time and consumption of extra bandwidth. Compared to RPL-based routing [RFC6550], NSA avoids the extra overhead of address assignment by integrating address assignment and tree forming together. Furthermore, NSA provides shorter packet header than non-storing mode RPL and much smaller routing table size than storing mode RPL.

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

Native Short Address (NSA) is an efficient topology-based network layer address assignment mechanism that is performed in a decentralized fashion. The NSA nodes are aware of its own IPv6 address constructed by IPv6 prefix (by configuration) and NSA (see Section 5.2). Inside the NSA domain, nodes communicate with each other using only NSA addresses. It is a smaller address space compared to the huge IPv6 addressing space. The NSA enables stateless forwarding. When IPv6 communication occurs between nodes inside the NSA domain and external IPv6 nodes, the border router, which plays as well the role of "root" in the addressing tree, performs network layer translation (as per Section 5.2 and [RFC6282]). The architecture of NSA network is showed in Figure 1.

```
                  /|\               Internet (IPv6)
                   |               --------+--------
     IPv6 Domain   |                       |
                   |                       |
                   |               +-------+-------+
   ----------------------------    | Border Router |
                   |               +---------------+
                   |
                   |                         O
                   |
                   |                O         O   O
                   |                  O  O
                   |             O                    O
     NSA Domain    |                           O
                   |          O      O  O          O    O   O
                   |               O
                   |                      O   O
                   |                                  O
                   |               O
                   |
                   |        Up to several thousands of nodes
                   |
                  \|/
                                           LLN
```

Figure 1: The architecture of general NSA networks.

The overall design objective is centered on how to minimize the packet overhead and reduce the size (or completely avoid the usage) of routing/forwarding table to save communication energy in an IoT LLN network. NSA eliminates compression/decompression of the address and also reduces the amount of information synchronization messages, so it actually reduces computation complexity during packets parsing and forwarding. To this end, NSA uses a context-independent address encoding mechanism. It does not carry any field about address context in the packet. It carries source and destination addresses by variable length fields whose size can be reduced to one octet each in the best case. This allows the NSA packet header to be smaller than LOWPAN_IPHC's 7 octets (see Figure 2), down to 4 octets, representing around 40% reduction in the header size. Considering that devices in the target limited domain are strongly constrained in resources, while still requiring to use a global unicast IPv6 address to identify them, 7 octets is the smallest size that LOWPAN_IPHC can achieve in a multi-hop environment, higher than the 2-3 octets necessary in a link-local communication [RFC6282].

```
  0   1   2   3   4   5   6   7     0   1   2   3   4   5   6   7
+---+---+---+---+---+---+---+---+  +---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 1 |   TF  |NH |HLM|  | 0 | 1 | 1 |  TF   |NH | HLIM  |
+---+---+---+---+---+---+---+---+  +---+---+---+---+---+---+---+---+
|Payload Length(variable length)|  |CID|SAC|  SAM  | M |DAC|  DAM  |
+---+---+---+---+---+---+---+---+  +---+---+---+---+---+---+---+---+
|I/O|AM |     Src(var len)      |  |      SCI      |      DCI      |
+---+---+---+---+---+---+---+---+  +---+---+---+---+---+---+---+---+
|        Dst (var len)          |  |                               |
+---+---+---+---+---+---+---+---+  +         Source Address        +
                                   |                               |
                                   +---+---+---+---+---+---+---+---+
                                   |                               |
                                   +       Destination Address     +
                                   |                               |
                                   +---+---+---+---+---+---+---+---+

                                     b. IPHC best case header
   a. NSA best case header              with context-based encoding
                                        and global unicast address
```

Figure 2: Best case of NSA and LOWPAN_IPHC packet header.

There are three distinct NSA features that allow NSA to be efficient, namely:

1. Native Short Address allocation (see Section 4),

2. Stateless forwarding (see Section 5),

3. Compact header format design (see Section 7) that avoids context and compression.

4. NSA Allocation

In an NSA network, there are 3 types of roles, namely:

* Root,

* Forwarder,

* Leaf.

The basic rules of allocation include:

* Each node's address is prefixed by their parent's address.

* The forwarder runs an AF (Allocation Function) to generate its children's addresses.

* All nodes run the same AF in the same network instance.

Normally, the root role is assigned to the border router when the LLN bootstraps. An example of a possible result of an NSA deployment is shown in Figure 3.

```
                            root           +--------------------------+
                                 1         | append more bits to form |
                                 O ----+   | brother's address        |
                              /  |  \   \  +--------------------------+
                             /   |   \    \
                            /    |    \     \
         +---------+      /      |     \      \
         |forwarder| 10 /       11   110\       \  111
         |node     |  O -        O       O        O
         +---------+/ |\ \               | \
                  /   | \  \             |  \
                /     |  \    \          O   O
              /       |    \    \
          100/     101|   1010  1011   +--------------+
            O         O      O      O  |Prefix is '10'|
           /|        /|                +--------------+
          / |       / |
         O  O      O  O
```

Figure 3: An example of NSA addresses allocation.

Each node acquiring a native short address needs to send an Address Request (AR) message to its link layer neighbors and wait for the response. In the AR message, the node needs to designate a 'role' value (forwarder or leaf) and the 'nodeid'. Forwarder and Leaf roles can be assigned similarly to IEEE 802.15.4, which distinguishes between Full-Function Devices (FFD) and reduced function devices (RFD) (cf., [ZigBee]). If a neighbor is neither a forwarder nor the root, it will drop the message silently. Otherwise, the neighbor should calculate an address based on parameters in the AR message. After the neighbor node assigns an address to the node, using the Allocation function (AF), it stores the suffix of that address as the interface ID towards the node. Then, it generates and sends Address Assignment (AA) message back. Once a node receives a valid AA response, it uses that assigned address as its own network layer address, thus becomes a child of the address assigner. It will then ignore replies from other neighbors. If a node does not receive any response after an pre-defined interval, it will send the AR message again. It is RECOMMENDED that nodes re-send the AR message up to 3 times, if no answer is received they SHOULD stop.

The allocation function AF(role,i) used in this document is defined in Figure 4. Where every forwarder node should store and maintain two indexes, one for the children that are forwarders and one for the children that are leaves (starting at 0 for the first child in each role). Let's call the first index 'f', as of forwarder, and the second 'l' as for leaves. The '+' symbol indicates a concatenation operation. The b() operation indicates the binary string of '1' with length equal to its argument, for instance b(3) returns '111'.

```
AF(role, f, l) = 'address of the node performing the function'
               + (role == leaf? b(l++):b(f++))
               + (role == leaf?'1':'0'),
in which, f and l are the indexes of respectively the forwarders
and the leaves at this layer (starting at 0).
```

Figure 4: Definition of the Allocation Function (AF) of forwarder/root nodes.

Taking the example of the topology in Figure 3, the proposed AF works as follows.

At the top level, there are 4 children of root, two are forwarders and the other two are leaves. Starting from the left most node and moving to the right, the root node applies the AF as follows:

* For the first child, which is a forwarder:

- A('forwarder', 0, 0) = '1'(root address) + b(0) + '0' = '1' + '' + '0' = 10

- Index f is increased by one and is now equal 1 (f=1)

* For the second child, which is a leaf:

  - A('leaf', 1, 0) = '1'(root address) + b(0) + '1' = '1' + '' + '1' = 11

  - Index l is increased by one and is now equal 1 (l=1)

* For the third child, which is a forwarder:

  - A('forwarder', 1, 1) = '1'(root address) + b(1) + '0' = '1' + '1' + '0' = 110

  - Index f is increased by one and is now equal 2 (f=2)

* For the fourth child, which is a leaf:

  - A('leaf', 2, 1) = '1'(root address) + b(1) + '1' = '1' + '1' + '1' = 111

  - Index l is increased by one and is now equal 2 (l=2)

The first level addresses have now been assigned. Let's now have a look how the node 10 (the first forwarder child of the root) applies the same Allocation Function. Note that node 10 will use its own 'f' and 'l' indexes initialized to 0. Starting again from the left most node, node 10 applies the AF as follows:

* For the first child, which is a forwarder:

  - A('forwarder', 0, 0) = '10'(node address) + b(0) + '0' = '10' + '' + '0' = 100

  - Index f is increased by one and is now equal 1 (f=1)

* For the second child, which is a leaf:

  - A('leaf', 1, 0) = '10'(node address) + b(0) + '1' = '10' + '' + '1' = 101

  - Index l is increased by one and is now equal 1 (l=1)

* For the third child, which is a forwarder:

- A('forwarder', 1, 1) = '10'(node address) + b(1) + '0' = '10' + '1' + '0' = 1010

- Index f is increased by one and is now equal 2 (f=2)

* For the fourth child, which is a leaf:

  - A('leaf', 2, 1) = '10'(node address) + b(1) + '1' = '10' + '1' + '1' = 1011

  - Index l is increased by one and is now equal 2 (l=2)

Note how the children of the same parent all have the same prefix (10 in this example). The proposed AF algorithmically assigns addresses to the different nodes without the need to know the topology in advance. However, the largest address of the network will depend on the actual topology. Indeed, the maximum length of an address with the proposed AF grows linearly at each level of the tree with the number of siblings from the same parent. Let's take again the example in Figure 3 and let's assume that the children of node 10 are all leaves, for the largest address we need 2 bits to encode the parent node prefix (10 in this case) to which we need to add a number of '1' equal to the value of the l index which is the number of leaves minus one (because the first leaf has index 0), in this case since there are 4 leaves, the index value is 3 and we add the '111' string, hence the address length would be 6 (2 for the prefix, 3 to encode the 4th leaf address, and one for the final 1 the ends all leaves addresses). In a more formal way the maximum address length at each level can be calculated as (where Ceiling just returns the least integer greater or equal its argument):

```
Max_Length = length(Parent address)
             length(b(max(f,l)))
             + 1
```

Where f and l are the indexes counting respectively the forwarders and the leaves at this level.

The Allocation Function can be different from the one defined in Figure 4, but all nodes know which one to use by configuration. The use of one and only one AF is allowed in an NSA domain. It is RECOMMENDED that implementations support at least the AF proposed in this document (cf. Section 9).

Different allocation function may, for example, leverage on a priori knowledge of the topology in order to optimize the maximum address size and make it smaller. For instance, because the order of address allocation has an impact on the size, the address of children with

the largest sub-tree should be allocated in the first place so to reduce the average address length of the whole sub-tree. Also, knowing the traffic in advance, or being able to have an estimate, can help to minimize the size of addresses that have a lot of traffic. This kind of optimization can be an option, the specification of optimizations is out of the scope of this document and may be defined in new Allocation Functions to be added to the "Allocation Function Registry" (see Section 9).

4.1. NSA Addresses and IPv6 Addresses

Obtaining a full IPv6 address from a NSA address is pretty straightforward. First the NSA address is concatenated to the configured IPv6 prefix. Since the length of the NSA address is very likely smaller than 64 bits (the interface ID length in IPv6), the node needs to pad it with zeros ('0') used as most significative bits. The full IPv6 address will look like: IPv6 prefix + "000...000" + NSA (or in IPv6 notation <IPv6 Prefix>::<NSA>). The NSA is assigned by the root/forwarder as previously described.

In an IPv6 communication, the node will derive the NSA address as the short source address from its own IPv6 address by simply removing the IPv6 prefix and all leading zeros before the NSA part. The node will compare the destination IPv6 address with its own IPv6 address. If they have the same prefix, it means that the destination is in the local NSA domain and its corresponding NSA address will be extracted as the short destination address (and the I/O Flag can be set accordingly). Otherwise, it will be a communication towards the Internet. In that case, a mapping mechanism implemented on the root node will generate a short address to be mapped to the full IPv6 destination address. As previously stated, the mapping mechanism is out of the scope of this document.

Since the short mapped address is generated on the root, when the node first open the connection toward the external site, with a first packet, the destination address is set to the full, uncompressed, IPv6 address. Once the packet arrives to the root node, performing the destination address lookup the root will notice that a full IPv6 address is being used and will trigger the short address generation mechanism and create a new mapping. Such, mapping is communicated to the source node via a new dedicated ICMP message (see Section 8). Once the node originating the communication receives such a message it MUST use the mapped short address for any further communication.

4.2. Limitation of Number of Children Node

The maximum number of child nodes is determined by the specific AF used. IEEE 802.15.5 has explored the use of a per-branch setup, which, however, incurs scalability problems [LEE10]. NSA allocation design is more flexible and extensible than the one proposed in IEEE 802.15.5. The AF used as example in this document does not need any specific setup network by network, though it is still limited by the maximum length of addresses. For the special case of the parent connecting to huge amount of children, a variant of the proposed AF can be designed to fulfill the requirement and optimize the address allocation (as previously described).

5. Forwarding in a NSA Network

Internal and external communications in an NSA network work slightly different. For internal communications, among NSA endpoints, packets carry native short addresses and no special operation is needed. For external communications, the root is responsible to perform the translation between native short addresses and IPv6 addresses. For instance, for a packet entering into the NSA domain, the root will extract the native short address of the destination from the suffix of the IPv6 address, by removing all leading '0's. It will also map the source IPv6 address to a mapped native short address, in order to make it more efficient for communication inside the NSA domain.

The root has to store the mapping between external IPv6 addresses and their assigned mapped Native Short Addresses. The method of generating those mapping is out of scope of this document, however, the addressing space for the external NSA has to be maintained separate from the internal NSA address space. Overlap are allowed since the two addressing space are distinguishable in the packets by the use of the I/O field, as explained later on.

The following paragraphs will detail the forwarding operations for both internal and external communication. The intra-network forwarding procedure depends on the specific AF used. Here we will use the AF previously introduced (see Figure 4) to illustrate the forwarding procedure.

5.1. Forwarding toward an NSA endpoint

To perform forwarding operations, NSA nodes access the I/O field in the NSA header (see Section 7). When its value is 1, the packet is destined to an internal NSA node, so it is an inner-domain packet. Otherwise, the packet is destined to an external IPv6 node, so it is called an outer-domain packet. Intra-domain packets carry a native short addresses in the source and the destination address fields.

More specifically the destination address field is the address of another node in the same NSA domain. As such an NSA node performs the following sequence of actions (also see Figure 5):

1. Get destination address from packet (abbreviated to DA) and the current node's address (abbreviated to CA). Go to step 2.

2. If length of DA is smaller than length of CA, send the packet to parent node, exit. Otherwise, go to step 3.

3. If length of DA equals to length of CA, go to step 4. Otherwise, go to step 5.

4. If DA and CA are the same, the packet arrived at destination, exit. Otherwise, send the packet to parent node, exit.

5. Check whether CA is equal to the prefix of DA. If yes, go to step 6. Otherwise, send the packet to parent node, exit.

6. Calculate which child is the next hop address and forward packet to it. With the AF propose in this document such operation reduces to reading the DA's bits starting from the position equals to the length of CA, then skip all '1' until the first '0' or the last bit of DA. The sub-string obtained in such a way is the address of direct child of current node.

7. If any exception happens in the above steps, drop the packet and send error notification.

```
              /*\       DA:Destination Address
             |***|      CA:Current Node's Address
              \*/
               |
      +--------+--------+
      |Parse DA from pkt|
      +--------+--------+
               |
              \|/
       +-------+------+
      /                \  yes
     | Len(DA)<Len(CA)? |-------------------------------+
      \                /                                |
       +-------+------+                                 |
               | no                                     |
              \|/                                       |
       +-------+------+           +--------------+      |
      /                \  yes    /                \  no |
     | Len(DA)=Len(CA)? |------>|     CA == DA ?   |--->+
      \                /         \                /     |
       +-------+------+           +-------+------+      |
               | no                       | yes         |
              \|/                        /*\            |
       +-------+------+                 |***|           |
      /                \  no             \*/            |
     | CA==PrefixOf(DA)?|------------------------------>+
      \                /                                |
       +-------+------+                                 |
               | yes                                    |
              \|/                                      \|/
     +---------+---------+                    +---------+---------+
     | Calculate next-hop|                    | Forward to Parent |
     |         &         |                    +---------+---------+
     |      Forward      |                              |
     +---------+---------+                              |
               |<---------------------------------------+
              \|/
              /*\
             |***|
              \*/
```

Figure 5: Flow Chart of Internal Forwarding Procedure

In the case of packet arriving from the Internet (external IPv6 domain toward the local NSA domain) header adaptation operation is performed by the root node. Concerning the destination address, the root builds the native short address of the destination by removing the prefix and the leading '0's of the suffix of the destination

address. Meanwhile, it checks whether it exists already a mapping between the source address and a mapped NSA address to be used as source address in the NSA packet. If not it creates one. Then the root creates the inner-domain packet. It uses the NSA address as destination setting the I/O field to 1 so to route the packet to as described above to the destination node. The mapped NSA address is used as source address and the fact that is a Mapped Address is signaled by setting to 1 the MA field.

5.2. Forwarding toward an external IPv6 node

In the case that the I/O field (cf. Section 7) is set to 0, the packet is destined to an external IPv6 node, it is an outer-domain packet. As such the destination address is either a full IPv6 address (for the first packet of a communication) or a mapped short address generated by the root node and not belonging to any node inside the NSA domain.

All NSA nodes (except root) just send packets that are destined outside the local domain (I/O field equal 0) to their parent, not even looking at the actual destination address. Eventually all packets will reach the root node, which acts as gateway. The root node is able to map the destination NSA address to the corresponding full IPv6 address. Also, the root node is able to rebuild the full source IPv6 address by concatenating the IPv6 prefix and the NSA address as explained in Section 5.2. Other fields of the header are also decompressed as described in Section 7. A full IPv6 header replaces the original NSA header in the packet, which is then forwarded according to traditional IPv6 protocol.

6. Benefits of Native Short Addressing

The NSA use a single set of messages for address assignment and tree forming. It is not more complex than RPL tree forming. So, NSA saves the overhead of address assignment of RPL.

Comparing to RPL with storing mode (see [RFC6550]), there is no need for a NSA node to generate and store routing table entries in the normal case. One of the potential issues is the risk of renumbering of addresses in case of topology changes. Because of the applicability domain of NSA, the common case of topology change is known in advance and can be planned, so to reduce disruption due to renumbering. Another case is temporary link failures where the underlying technology is still able to provide connectivity through alternative links.

In this scenario a node can temporarily "move" along with its whole sub-tree. Instead of performing a renumbering of the whole sub-tree, which may cause disruption, the sub-tree rooted on the "moving" node, its address and the addresses of all its children and grand-children can be kept unchanged, by creating a temporary entry in the forwarding tables of the original and new parent nodes, in order to make the sub-tree still reachable.

Herewith one example, also depicted in Figure 6, node A with the address of 1000 somehow moves from node B (original parent) to node C (new parent). In this case, the forwarding tables in B, C and their parents' nodes should be updated by adding a new entry to "1000", the address of node A. Meanwhile, the original parent (node B) should keep its original address assignment. Comparing with renumbering the addresses of node A and its children, the cost of adding one new route to their parent nodes is much lower, although in this case, the NSA does not implement complete stateless forwarding. However, this solution SHOULD be used only in case of temporary topology changes, where the entries will be deleted once the original topology is re-established. For permanent topology changes it is RECOMMENDED to re-run the NSA Allocation Function.

```
          +------------------+
          |     Add route:   |
          |to 1000, ia 1010  |     1
          +------------------+     O
                           *     /
     +-----------------+    *   /
     |     Add route:  |     * /
     | to 1000, via 10 |     O 10
     +-----------------+     / \
                        *   /    \
                         * /       \ 1010 (node C)
              (node B)100 O         O
                          .       /  *
                          .     /     *-----------------+
                          .   /       |     Add route:  |
                          O /         | to 1000, direct |
                        1000(node A)  +-----------------+
```

Figure 6: Add extra routes for "logic topology change

7. NSA Header Format

As explained in Section 4, the addresses in NSA are of variable length, in this section, we outline the design of the header format partially based on the format of 6lowPAN, accommodating the variable length property in the packet. The header format is shown in Figure 7.

```
  0                                       1
  0   1   2   3   4   5   6   7   8   9   0   1   2   3   4   5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| 0 | 1 | 0 | 1 |  TF   |NH |HL | Payload Length(Variable Len)  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|I/O|MA |    SA(Variable Len)   |         DA(Variable Len)      |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                        In-line fields ...                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

Figure 7: Header format of NSA packets

The first 4 bits are new dispatch types that will be introduced in Section 9.

* TF: The same definition as in [RFC6282] Section 3.1.1.

* NH: The same definition as in [RFC6282] Section 3.1.1.

* HL: This field indicates the hop limit. When HL is 1, a hop limit field defined in [RFC2460] locates in in-line fields, while HL is 0 means no hop limit header in packet.

* Payload length is a variable length field. It normally occupies an octet assuming most packets are smaller than 252 octets. For larger packets, payload length may expand to 2 to 3 octets. The encoding method is defined as follows. When the first octet has value of:

  - 0~252: Indicates how many octets the payload consist of.

  - 253: Indicates that there is an extra octet for payload length, with the actual length value equal to the last octet value plus 252.

  - 254: Indicates that there is an extra two octets for payload length, with the actual length value obtained from the second and third octets interpreted as a 16 bits unsigned integer plus 252 (from the first octet).

- 255: Reserved.

* I/O: Indicates whether this packet is destined to a inner-domain node (value '1') or an outer-domain node (value '0'), where the former means from an NSA or IPv6 node to a NSA destination, while the latter means to an external IPv6 node.

* MA: Indicates the source address is actually a Mapped Address generated by the root. When it is '1', the source address of the packet is a mapped address of an external IPv6 address, while if it is '0', the source address of the packet is an NSA address.

For length variable native short address encoding, for both Source Address (SA) and Destination Address (DA), the definition is:

* 0~252: if the address value locates in this interval, one octet is used to encode the value

* 253: indicates that the following 2 octets encode the address.

* 254: indicates that the following 4 octets encode the address.

* 255: indicates that the following octet defines the length of address in octets, followed by the address octets.

The sequence of in-line fields is as per [RFC8200] section 3.

8. NSA Control Message

This documents specifies only one NSA Control Message, namely the NSA Mapped Address Advertisement described in Section 4. The purpose of such a message is advertise the mapping of an IPv6 address into a NSA address. The map is performed by the root node and sent to the node originating the communication. The root keeps a copy of the mapping to be used for future packets. The format is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Type         | Code = 0x00   | Reserved      | NSA Length    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|          Target IPv6 Address (Fixed length 128 bist)          |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Target NSA Address (Variable length) .... |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

* Type: Type value identifying NSA Control Message. Value to be assigned by IANA (cf. Section 9)

* Code: This field identifies the specific control message. In this case it is set to the value 0x00 "NSA Mapped Address for External IPv6 Address".

* Reserved: Set as 0 on transmission and ignored on reception.

* NSA Length: This field indicates the length of the Target NSA Address at the end of the message, expressed in octets.

The "NSA Mapped Address for External IPv6 Address" is a variable length message, however, the first five fields of the message, namely Type, Code Reserved, NSA Length, and Target IPv6 address, have a fixed length of 160 bits (20 octets), hence the length of the NSA address is sufficient to calculate the length of the entire packet: 20 octets + "NSA length".

9. IANA Considerations

9.1. Dispatch Type Field

This document requires IANA to assign the range 01010000 to 01011111 in page 10 of the "Dispatch Type Field" registry as follows:

Bit Pattern	Page	Header Type	Reference
0101TTNH	10	LOWPAN NSA IP(LOWPAN_NIP)	[This Document]

Figure 8: LOWPAN Dispatch Type Field requested allocation

9.2. Allocation Function Registry

This section provides guidance to the Internet Assigned Numbers Authority (IANA) regarding registration of values related to the NSA specification, in accordance with BCP 26 [RFC8126].

IANA is asked to create a registry named "Native Short Addresses (NSA) Parameters".

Such registry should be populated with a one octet sub registry named "Allocation Function" and used to identify the AF used in a NSA deployment. The sub registry is populated as follows:

Value	AF Name	Reference
0x00	Native Allocation Function	[This Document]
0x01-0xFF	Un-assigned	

Values can be assigned by IANA on a "First Come, First Served" basis according to [RFC8126].

9.3. ICMP NSA Control Message

IANA is requested to allocate an ICMPv6 type value from the "ICMPv6 Parameters" registry to be used by "NSA Control Message".

Also IANA is requested to create an "NSA Control Codes" sub registry, for the Code field of the ICMPv6 NSA Control Message.

New codes may be allocated through the "Specification Required" procedure as defined in [RFC8126]. The following code is currently defined (the others are to be marked as un-assigned):

Code	Description	Reference
0x00	NSA Mapped Address for External IPv6 Address	[This Document]

10. Security Considerations

An extended security analysis will be provided in future revision of this document. As of this point we consider that the security considerations of [RFC4944], [RFC6282] apply.

11. References

11.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, DOI 10.17487/RFC2460, December 1998, <https://www.rfc-editor.org/info/rfc2460>.

[RFC4944] Montenegro, G., Kushalnagar, N., Hui, J., and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944, DOI 10.17487/RFC4944, September 2007, <https://www.rfc-editor.org/info/rfc4944>.

[RFC6282] Hui, J., Ed. and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, DOI 10.17487/RFC6282, September 2011, <https://www.rfc-editor.org/info/rfc6282>.

[RFC6550] Winter, T., Ed., Thubert, P., Ed., Brandt, A., Hui, J., Kelsey, R., Levis, P., Pister, K., Struik, R., Vasseur, JP., and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks", RFC 6550, DOI 10.17487/RFC6550, March 2012, <https://www.rfc-editor.org/info/rfc6550>.

[RFC8025] Thubert, P., Ed. and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Paging Dispatch", RFC 8025, DOI 10.17487/RFC8025, November 2016, <https://www.rfc-editor.org/info/rfc8025>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

11.2. Informative References

[I-D.ietf-6lo-use-cases] Hong, Y., Gomez, C., Sangi, A. R., and S. Chakrabarti, "IPv6 over Constrained Node Networks (6lo) Applicability & Use cases", Work in Progress, Internet-Draft, draft-ietf-6lo-use-cases-11, 12 July 2021, <https://www.ietf.org/archive/id/draft-ietf-6lo-use-cases-11.txt>.

[LEE10] Lee, M., Zhang, R., Zheng, J., Ahn, G., Zhu, C., Park, T., Cho, S., Shin, C., and J. Ryu, "IEEE 802.15.5 WPAN mesh standard-low rate part: Meshing the wireless sensor networks", IEEE Journal on Selected Areas in Communications Vol. 28, pp. 973-983, DOI 10.1109/jsac.2010.100902, September 2010, <https://doi.org/10.1109/jsac.2010.100902>.

[LPWAN] "IPv6 over Low Power Wide-Area Networks (lpwan) WG", n.d., <https://datatracker.ietf.org/wg/lpwan/about/>.

[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>.

[RFC8138] Thubert, P., Ed., Bormann, C., Toutain, L., and R. Cragie, "IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) Routing Header", RFC 8138, DOI 10.17487/RFC8138, April 2017, <https://www.rfc-editor.org/info/rfc8138>.

[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>.

[SIXLO] "IPv6 over Networks of Resource-constrained Nodes (6lo) WG", n.d., <https://datatracker.ietf.org/wg/6lo/about/>.

[SIXLOWPAN] "IPv6 over Low power WPAN (6lowpan) - Concluded WG", n.d., <https://datatracker.ietf.org/wg/6lowpan/about/>.

[ZigBee] "ZigBee Wireless Networks and Transceivers", Elsevier book, DOI 10.1016/b978-0-7506-8393-7.x0001-5, 2008, <https://doi.org/10.1016/b978-0-7506-8393-7.x0001-5>.

Authors' Addresses

Guangpeng Li
Huawei Technologies
Beiqing Road, Haidian District
Beijing

100095
China

Email: liguangpeng@huawei.com

David Lou
Huawei Technologies Duesseldorf GmbH
Riesstrasse 25
80992 Munich
Germany

Email: zhe.lou@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

Peng Liu
China Mobile
No. 53, Xibianmen Inner Street, Xicheng District
Beijing
100053
China

Email: liupengyjy@chinamobile.com

Rong Long
China Mobile
No. 53, Xibianmen Inner Street, Xicheng District
Beijing
100053
China

Email: longrong@chinamobile.com