IETF 120 – 6lo

Path-Aware Semantic Addressing (PASA) for Low power and Lossy Networks

~~draft-ietf-6lo-path-aware-semantic-addressing-04~~
~~draft-ietf-6lo-path-aware-semantic-addressing-05~~
~~draft-ietf-6lo-path-aware-semantic-addressing-06~~
draft-ietf-6lo-path-aware-semantic-addressing-07

L. Iannone, G. Li, D. Lou, P. Liu, R. Long, K. Makhijani, P. Thubert

IETF 120 – Vancouver

Since IETF 119

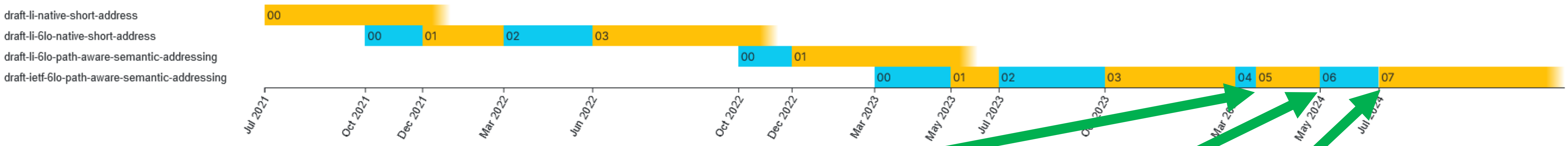
Path-Aware Semantic Addressing (PASA) for Low power and Lossy Networks

draft-ietf-6lo-path-aware-semantic-addressing-07

Status | IESG evaluation record | IESG writeups | Email expansions | History

Versions:

00 | 01 | 02 | 03 | 04 | 05 | 06 | 07



draft-ietf-6lo-path-aware-semantic-address-05.txt
March 2024
Main changes: Co-authors' affiliation update

draft-ietf-6lo-path-aware-semantic-address-06.txt
May 2024
Main changes: Address GENART review by Paul Kyzivat

draft-ietf-6lo-path-aware-semantic-address-07.txt
July 2024
Main changes: Address RTGDIR review by Joel Halpern

Main Content Changes 05 => 06 (I)

No changes in the document structure

```
3.  Definition of Terms

   PASA Root:  The PASA root node is the router responsible for the
      management of the whole PASA network and routing/forwarding both
      internal and external traffic.  It uses the Address Assignment
      Function (AAF) and performs the address assignment for its
      children.  The root node functions as gateway between the PASA
      domain and the Internet, acting as what [RFC8505] names 6LBR
      (6LowPAN Border Router).

   PASA Router:  A PASA Router is an internal node, different from the
      PASA Root, acting as a router, hence as what [RFC8505] names 6LR
      (6LowPAN Router).  Similar to the PASA Root, it uses the address
      Assignment Function (AAF) and performs the address assignment for
      its children.

   PASA Host:  A PASA Host is a node with no children (i.e., a leaf), it
      is what [RFC8505] names 6LN (6LowPAN Node).  This node does not
      perform the address Assignment Function.  It merely requests an
      address to its selected parent.

   Address Assignment Function (AAF):  As defined in

      [I-D.iannone-6lo-nd-gaao].  Used by PASA Root and PASA Routers to
      assign addresses to their children.
```

```
3.  Definition of Terms

   PASA Root:  The PASA root node is the router responsible for the
      management of the whole PASA network and routing/forwarding both
      internal and external traffic.  It uses the an Address Assignment
      Function (AAF) and performs the address assignment for its
      children.  The root node functions as gateway between the PASA
      domain and the Internet, acting as what [RFC8505] names 6LBR
      (6LowPAN Border Router).

   PASA Host:  A PASA Host is a node with no children (i.e., a leaf), it
      is what [RFC8505] names 6LN (6LowPAN Node).  This node does not
      perform the address Assignment Function.  It merely requests an
      address from its selected parent.

   PASA Router:  A PASA Router is an internal node, different from the
      PASA Root, acting as a router, hence as what [RFC8505] names 6LR
      (6LowPAN Router).  Before acting as a router it will act as a PASA
      Host by acquiring an address.  Then, similar to the PASA Root, it
      uses the Address Assignment Function (AAF) and performs the
      address assignment for its children.

   Address Assignment Function (AAF):  As defined in

      [I-D.iannone-6lo-nd-gaao].

   Tree Address Assignment Function (TAAF):  As defined in Section 6.
      Used by PASA Root and PASA Routers to assign addresses to their
      children.
```

- Inverted definition of PASA Host and PASA Router
  - A PASA Router at boot time first act as a PASA Host to obtain an address then switching role to PASA Router
- AAF => TAAF: **Tree** Address Allocation Function

Main Content Changes 05 => 06 (II)

* In an idealized PASA-based OT domain, a leaf-node could be a field device (sensor or actuator) that always connects to PLC serving as last node forwarding traffic to/from the leaves, i.e. sensors and actuators.

* In an idealized PASA-based OT domain, a leaf-node could be a field device (sensor or actuator) that always connects to PLC serving as last node forwarding traffic to/from the leaves, i.e. sensors and actuators. Hence, the PLC will work as a PASA Router only for field devices supporting IPv6. For field devices not supporting IPv6, the PLC will assign PASA addresses for each of them, and then translate between IPv6 packets and the device protocol, making the devices appear as PASA Hosts within the enclosing PASA Domain.

- Clarification about PLC functioning

A node that, for any reason, reboots does not need to restart the whole procedure. According to [I-D.iannone-6lo-nd-gaao] and [RFC8505] address registration state has to be stored in non-volatile memory, hence, when the nodes is up again there is no need to go through parent selection and address request, it can just re-register the previously obtained address.

- Clarification about nodes reboot
  - Usage of non-volatile memory
  - On reboot if state in non-volatile memory just re-register
  - Text amended in a few places in the document, including a revised "Reliability Considerations" section

12. Reliability Considerations

Because PASA uses algorithmically generated addresses based on the network topology, nodes do not generate and store forwarding 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 PASA, 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, which is strictly related to the underlying technology, the network topology, the deployed redundancy, and the expected reliability.

More complex reliability scenarios and alternative solutions are beyond the scope of this document, which is focused only on the address allocation framework and stateless forwarding. Furthermore, specific reliability solutions can depend as well on the specific Address Assignment Function used (different from the one presented in this document). Reliability is discussed in more details in [I-D.li-6lo-pasa-reliability].

12. Reliability Considerations

Because PASA uses algorithmically generated addresses based on the network topology, nodes do not generate and store forwarding table entries in the normal case. They are limited to have a default gateway and the ND table. One of the potential issues is the risk of renumbering of addresses in case of topology changes. Because of the applicability domain of PASA, the common case of topology change is known in advance and can be planned, so to reduce disruption due to renumbering (see Section 4).

Another case is temporary link failures or node temporary failures, where the network is still able to provide connectivity through alternative links, which is strictly related to the underlying technology, the network topology, the deployed redundancy, and the expected reliability. Failures may raise the issue of topology changes and re-numbering. Such a issues can be avoided or at least mitigated following the procedures in [RFC8505] and [I-D.iannone-6lo-nd-gaao] keeping state in non-volatile memory.

More complex reliability scenarios, including the case for multiple root nodes and alternative solutions, are beyond the scope of this document, which is focused only on the address allocation framework and stateless forwarding. Furthermore, specific reliability solutions depend as well on the specific Address Assignment Function used (different from the one presented in this document). A more in-depth discussion about reliability can be found in [I-D.li-6lo-pasa-reliability].

Main Content Changes 05 => 06 (III)

```
*  For the first child, which is a router:

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

```
*  For the first child, which is a router:

      TAAF('router', 0, 0) = '1'(root address) + b(0) + '0'
                           = '1' + '' + '0'
                           = '10'
```

- Adjusted example in Section 6.1 for readability

```
The converse operation, from full IPv6 format to PASA format is very
simple.  Firstly the configured IPv6 prefix is cut out.  Secondly, on
the remaining 64 bits, all leading zeros can be trimmed.  Indeed,
because all TAAF addresses are derived from the PASA Root, and since
the latter has address '1', all TAAF generated addresses always start
with '1' (See Section 6.1).  As such, trimming the leading zeros is a
safe operation returning a PASA TAAF address.
```

- Added text on how to go from IPv6 address format to short PASA address format

```
                         root          +--------------------------+
                             1         | append more bits to form |
                             o ----+   | brother's address        |
                            /|  \   \  +--------------------------+
                           / |   \   \
                          /  |    \   \
+-------------+          /   |     \    \
| PASA Router | 10 /      11   110\      \  111
+-------------+ o -          o       o        o
               / |\ \                | \
              /  | \  \              |  \
             /   |  \    \           o   o
            /    |   \    \
      100/    1010|  101   1011   +-------------+
         o        o     o      o  |Prefix is '10'|
        /|       /|               +-------------+
       / |      / |
      o  o     o  o          +-----------+
  1001 10011 10101 101011    | PASA Host |
                             +-----------+
```

```
                         root          +--------------------------+
                             1         | append more bits to form |
                             ()----+   | brother's address        |
                            /|  \    \ +--------------------------+
                           / |   \    \
                          /  |    \    \
+-------------+          /   |     \     \
| PASA Router | 10 /      11   110\      \  111
+-------------+ ()           ()      ()       ()
               / |\ \                | \
              /  | \  \              |  \
             /   |  \    \           ()   ()
            /    |   \    \
      100/    1010|  101   1011   +-------------+
        ()       ()    ()     ()  |Prefix is '10'|
        / |      /|               +-------------+
       /  |     / |
      ()  ()   () ()         +-----------+
  1001 10011 10101 101011    | PASA Host |
                             +-----------+
```

- Updated misleading picture

Main Content Changes 06 => 07 (I)

No changes in the document structure

```
The overall design objective is centered on reducing the size of
routing/forwarding tables by using a topological addressing scheme.
PASA reduces the amount of information synchronization messages, so
it actually reduces computation complexity during packets parsing and
forwarding.  As such, PASA may save communication energy in an IoT
LLN network.
```

```
The overall design objective is centered on reducing the size of
routing/forwarding tables by using a topological addressing scheme.
PASA reduces the amount of information synchronization messages, so
it actually reduces computation complexity during packets parsing and
forwarding.  As such, PASA may save communication energy in an IoT
LLN network [LI22].  Compared to RPL-based routing [RFC6550], PASA
avoids the extra overhead of address assignment by integrating
address assignment and tree forming together.  Compared to RPL
storing mode, PASA uses smaller forwarding tables, hence less memory,
since there is no need to store topology information.  Compared to
RPL non-storing mode, PASA does has lower overhead in terms of
bandwidth since, instead of an explicit list of addresses, it encodes
the path directly in the address itself.  The overhead in both modes,
while smaller compared to routing protocols not designed for 6LowPAN
environments, is still not negligible [CHING21].  PASA addressing has
also lower overhead compared to BABEL, since there is no need to
share a routing table among neighbors ([NEUMANN15], [RFC8966]).
```

- Added some text in Section 5 on the why we are exploring PASA
  - Reduce overhead in specific deployments described in the use cases section.

Main Content Changes 06 => 07 (II)

```
7.1.  Forwarding toward a local PASA endpoint

   Intra-domain packets carry a PASA destination address in the PASA-
   6LoRH header.  More specifically the destination address field is the
   address of another node in the same PASA domain.  As such a PASA node
   receiving a packet performs the following sequence of actions (also
   see Figure 7):
```

```
           /-\       DA:Destination Address
          |***|      CA:Current Node's Address
           \_/
            |
   +--------+--------+
   |Parse DA from pkt|
   +--------+--------+
            |
           \|/
```

```
7.1.  Forwarding toward a local PASA endpoint

   Intra-domain packets carry a PASA destination address in the PASA-
   6LoRH header.  More specifically the destination address field is the
   address of another node in the same PASA domain.
   In the proposed TAAF algorithm the length of the addresses increases
   with the distance from the root, whose address has length 1.  The
   length operation, indicated by Len(X), of a PASA address X, returns
   the number of bits between the most significant bit of the IID that
   is set to 1 and the least significant bit.  For instance, for and
   address encoded on two bytes, len(0x00010010) = 5.

   Such property can be used to quickly take forwarding decisions based
   the length of the destination address.  Indeed, when a PASA router
   receives a packet destined to local PASA domain, the following three
   cases arise:

   *  The length of the destination address is shorter than the address
      of the PASA Router.  This means the destination is closer to the
      root and just forwarding to its parent is enough.

   *  The length of the destination address is equal to the address of
      the PASA Router.  In this case the destination is either the PASA
      Router itself, or another node in a different branch of the tree.
      The PASA Router compares its address and the destination address.
      If they are equal, then the packet has reached its destination.
      Otherwise the packet has to be sent to the parent in order to
      reach the right branch.

   *  The length of the destination address is greater than the address
      of the PASA Router.  In this case the destination is either in a
      sub-branch rooted in PASA router itself or in a totally different
      branch of the tree.  In the former case the packet has to be
      forwarded toward the correct child, in the latter just sent to the
      parent.  In order to decide which operation to do, the router
      compares its own address with the most significant bits of the
      destination address, in other words whether its own address is a
      prefix of the destination address.  If there is a match, then a
      child is selected as next hop based on the remaining bits of the
      destination address, otherwise, the destination is in totally
      different branch and the packet is sent to the parent.

   More formally, when a PASA node receiving a packet, it performs the
   following sequence of actions (also see Figure 7):
```

```
           /-\       DA:Destination Address
          |***|      CA:Current Node's Address
           \_/       Len(x): number of bits forming the PASA address
            |            y==PrefixOf(x): True if the most significant
   +--------+--------+                   bits of x equals y
   |Parse DA from pkt|
   +--------+--------+
            |
           \|/
```

- Added text in the forwarding algorithm section to better highlight the main concept and the operations
- Added legend on figure to define:
  - Len()
  - PrefixOf()

Main Content Changes 06 => 07 (III)

```
+================+=============+==================+
| Value          | Description | Reference        |
+================+=============+==================+
| 8 (suggested)  | PASA-6LoRH  | [This Document]  |
+----------------+-------------+------------------+

        Table 1: Critical 6LoWPAN Routing Header Type
                          for PASA
```

```
+=======+==============+==================+
| Value | Description  | Reference        |
+=======+==============+==================+
| TBD1  | PASA-6LoRH   | [This Document]  |
+-------+--------------+------------------+

        Table 1: Critical 6LoWPAN Routing
             Header Type for PASA
```

```
+=================+=====================+===========+
| Value           | AAF Name            | Reference |
+=================+=====================+===========+
| 0x01 (suggested)| PASA Tree Address   | [This     |
|                 | Allocation Function | Document] |
+-----------------+---------------------+-----------+

                 Table 2: PASA TAAF.
```

```
+=======+========================================+=================+
| Value | AAF Name                               | Reference       |
+=======+========================================+=================+
| TBD2  | PASA Tree Address Allocation Function  | [This Document] |
+-------+----------------------------------------+-----------------+

                 Table 2: PASA TAAF.
```

```
12.  Reliability Considerations

   Because PASA uses algorithmically generated addresses based on the
   network topology, nodes do not generate and store forwarding table
   entries in the normal case.  They are limited to have a default
   gateway and the ND table.  One of the potential issues is the risk of
```

skipping to change at *page 27, line 40*

```
   alternative links, which is strictly related to the underlying
   technology, the network topology, the deployed redundancy, and the
   expected reliability.  Failures may raise the issue of topology
   changes and re-numbering.  Such a issues can be avoided or at least
   mitigated following the procedures in [RFC8505] and
   [I-D.iannone-6lo-nd-gaao] keeping state in non-volatile memory.

   More complex reliability scenarios, including the case for multiple
   root nodes and alternative solutions, are beyond the scope of this
   document, which is focused only on the address allocation framework
   and stateless forwarding.  Furthermore, specific reliability

   solutions depend as well on the specific Address Assignment Function
   used (different from the one presented in this document).  A more in-
   depth discussion about reliability can be found in
   [I-D.li-6lo-pasa-reliability].
```

```
12.  Reliability Considerations

   Because PASA uses algorithmically generated addresses based on the
   network topology, nodes do not generate and store forwarding table
   entries in the normal case.  They are limited to have a default
   gateway and the ND table.  One of the potential issues is the risk of
```

skipping to change at *page 28, line 39*

```
   alternative links, which is strictly related to the underlying
   technology, the network topology, the deployed redundancy, and the
   expected reliability.  Failures may raise the issue of topology
   changes and re-numbering.  Such a issues can be avoided or at least
   mitigated following the procedures in [RFC8505] and
   [I-D.iannone-6lo-nd-gaao] keeping state in non-volatile memory.

   More complex reliability scenarios, including the case for multiple
   root nodes and alternative solutions, are beyond the scope of this
   document, which is focused only on the address allocation framework
   and stateless forwarding.  A more in-depth discussion about
   reliability, including the case of multiple roots, can be found in
   [I-D.li-6lo-pasa-reliability].  Furthermore, specific reliability
   solutions depend as well on the specific Address Assignment Function
   used (different from the one presented in this document).
```

- Modified IANA section (and text in the document) so that no codepoint is suggested
- Replaced with
  - TBD1
  - TBD2
- Re-phrased section 12 reliability to point out that the use of multiple roots is described in the reliability document

Next Steps

- Good progress with RTGDIR & GENART Reviews
  - Getting closer to WGLC

THANKS!