

Routing Area Working Group
INTERNET-DRAFT
Intended status: EXPERIMENTAL

Expires: June 30, 2022

A. Foglar, InnoRoute
M. Parker, Uni Essex
T. Rokkas, Incites
M. Khokhlov, IP Tek
November 22, 2021

IPv6 Source Routing for ultralow Latency
draft-foglar-ipv6-ull-routing-10

Abstract

This Internet-Draft describes a hierarchical addressing scheme for IPv6, intentionally very much simplified to allow for ultralow latency source routing experimentation using simple forwarding nodes. Research groups evaluate achievable latency reduction for special applications such as radio access networks, industrial networks or other networks requiring very low latency.

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 <http://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."

Copyright Notice

Copyright (c) 2019 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 (<http://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.

Revision Note for Version 02

Reference to experimental verification of the concept is added in the section "Acknowledgements".

Revision Note for Version 03

[Section 6](#) about Security Considerations has been inserted.

Revision Note for Version 04

[Section 7](#) about Redundancy has been inserted.

Revision Note for Version 05

[Section 8](#) about IANA Considerations added.

Revision Note for Version 06

[Section 8](#) about IANA Considerations updated.

Revision Note for Version 07

[Section 6](#) about Security Considerations improved.

Revision Note for Version 08

Soome typos corrected

Revision Note for Version 09

Improved address generation and ITU-T section added at the end of the document. An additional author is added.

Revision Note for Version 10

[Section 10](#) has been added describing a simple introduction scenario.

1. Introduction

To achieve minimum latency the forwarding nodes must support cut-through technology as opposed to the commonly used store-and-forward technology. Cut-through means, that the packet header already leaves a node at the egress port while the tail of the packet is still received at the ingress port. This short time does not allow complex routing decisions.

Therefore, a very simple routing address field structure is specified below. It should limit the complexity of the forwarding node used in the experiments. Therefore, in this text the term "forwarding node" is used instead of "router", although the device is operating in OSI Layer 3 and accordingly executes router functions such as decrementing the hop limit field.

2. IPv6 address prefix structure

The following proposal uses the 64-bit IPv6 address prefix.

Each forwarding node has up to 16 ports and hence needs 4 bits of the address field to decide to which port a packet should be forwarded. The 64-bit prefix is divided into 16 sub-fields of 4 bit, defining up to 16 hierarchy levels. A forwarding node is configured manually to which of the sub-fields it should evaluate for the forwarding decision.

A number n of leading 4-bit fields cannot be used for forwarding decisions, but must have a special value to indicate the 'escape prefix' of the experimental forwarding mode.

The 64-bit prefix of the IPv6 address has this structure:

| $n \times 4\text{-bit escape prefix}$ | $(16-n) \times 4\text{-bit address fields}$ |

The first 4-bit field following the escape prefix has the highest hierarchy level, the last 4-bit field has the lowest hierarchy level.

3. Forwarding node behavior

The forwarding node has up to 16 downlink ports and at least one uplink port. Typically, the forwarding nodes are arranged in a regular tree structure with one top node, up to 16 nodes in the second hierarchy, up to 256 nodes in the third hierarchy and so on for up to $16-n$ hierarchies.

A forwarding node must be configured to operate at a certain position in the hierarchical network. For example, at third hierarchy level, branch 4 of the first hierarchy and branch 12 of the second hierarchy.

The behavior of each forwarding node is depending on the position of a node in a hierarchical network. For all positions, the first step is to check the escape prefix. Only packets with matching escape prefix are forwarded.

The top forwarding node with the highest hierarchy level evaluates the first 4-bit field following the $n \times 4\text{-bit escape prefix}$. The value of the evaluation field determines the output port of the packet. The remaining fields are don't care:

| escape prefix | 4-bit | $(16-n-1) \times 4\text{-bit}$ |
< mandatory > <eval.> < don't care >

A forwarding node in a lower hierarchy first checks if the 4-bit fields preceding the evaluation field match the configured value. In case of match the value of the configured evaluation

field of the packet is used as downlink port number where the packet is forwarded. The remaining 4-bit fields are ignored. In case of mismatch the packet is forwarded to the uplink port(s).

```
| escape prefix | m x 4-bit | 4-bit | (16-n-m-1) x 4-bit |  
< mandatory   > < match   > <eval.> <   don't care   >
```

The parameter m indicates the hierarchy level with m=0 denoting the highest hierarchy.

Hence, when a packet enters a hierarchical network at the lowest layer node it is forwarded in uplink direction until it reaches a node where the m x 4-bit prefix matches the configured value of the node. At latest, the highest-level node will always match and forward the packet in the desired downlink direction.

4. Numerical values

As mentioned, one pre-requisite of the simple forwarding concept is to keep the complexity of the forwarding nodes low. Also, the configuration of the nodes should be kept simple. In particular, industrial networks are operated by persons who are not experts in communication. Configurations should be intuitively understandable by all without long explication. Therefore, for the first experimental forwarding node the number of downlink ports is limited to 10 with numbers 0...9. 16 digits at the front panel of the forwarding device show the configuration. Use of classical 7-segment digits make the limits of the configuration obvious.

As escape code, the first two digits are fixed to the value "AF" (binary '10101111'). These two characters contrast with the following numerical digits, so that the escape code can be clearly differentiated from the following configuration. The display uses the 'H' character instead of the 'X' the usual term for a variable. It can be interpreted as 'hierarchy'.

The H specifies the digit of the packet prefix which is evaluated for forwarding. When the H is selected all lower digits are automatically set to '-' to indicate the don't care nature.

To make the configuration still more obvious it is recommended to configure the local telephone number. With that measure, every local experimentation has unique numbers and can potentially be interconnected via tunnels (IP, MPLS, VPN etc.) with other experimental setups.

The length of 14 digits allows sufficient in-house

hierarchies, even for industrial applications where forwarding nodes interconnect large numbers of sensor controllers. Inhouse installations would be structured for example in building, floor, fabrication unit, machine - with one sensor controller per machine. For the sake of simplicity numbers are deliberately wasted, for example if the building has only 3 stories the digits 4...9 are unused.

5. Example configuration

A company office in Munich with the telephone number +49-89-45241990 configures its local top-level forwarding node to:

AF49.8945.2419.90H-

Note that for the sake of simplicity this simplified notation is introduced here as alternative to the usual notation AF49:8945:2419:90:0/56. With the new notation, the cabling staff people can immediately check the hierarchy location of the forwarding node and connect the cables to the floors at ports 0...3.

The next hierarchy level is related to the floor. In case of a 3-story building only three next level forwarding nodes are used with these configured values:

AF49.8945.2419.900H at the ground level

AF49.8945.2419.901H at the first floor

AF49.8945.2419.902H at the second floor

AF49.8945.2419.903H at the third floor.

In each floor, up to 10 sensor nodes can be connected.

Each of the sensor nodes can address several sensors/actuators addressed via the interface identifier contained in the second part of the 128-bit IPv6 address.

In the following a connection between sensors in this office to other IoT equipment located in Essex University is described. The connection is realized with one additional forwarding node installed at Essex University premises with the second level address

AF4H.----.----.----.

This high level forwarding node can be used although the phone number of the researcher is +44 1206 872413, as long as there is no further node in UK.

At downlink port 9 the 13th level forwarding node in Munich is connected via a Layer 2 link such as VLAN or SDH pipe or MPLS tunnel. The levels in between must not be populated by forwarding nodes as long as no other branch is needed at one of the two sides. If for example another site in Munich center must be connected one additional forwarding node must be installed with the 5th level address

that packets received from downlink 1 have source addresses AF49.891x.xxxx.xxxx with x is don't care. To that aim the node checks if the leading digits of the packet source address match with AF49.89 and if the digit at the 'H' position matches with the receiving downlink port number.

The lower the hierarchy level of a node the more digits are checked. In particular, the lowest hierarchy node checks the complete prefix.

For example, the Munich IoT node in the Figure above must send packets with the source address AF49.8945.5419.9014 to the higher level node. It will discard packets with any other source address.

Hence in upstream direction every higher level node will check a shorter part of the prefix. At the highest level the node AFH-.----.----.---- will check if the source address digit at the 'H' position matches with the receiving downlink port number.

As packets with non-matching source address are discarded a receiver can rely on the correctness of the source address. This feature provides an orthogonal level of security to existing security measures such as password authentication and encryption. Anonymous hackers are not possible in such hierarchical networks. Receivers may use white-listing for address filtering.

To circumvent the source address check a hacker must break into the network and insert packets in downstream direction. At the highest level node the network is most vulnerable, as any address can be reached from there. However, the higher a network node level the more sophisticated are the security means to avoid intrusion.

At lower level nodes an additional source address check in downstream direction may be implemented: at the uplink ports packets with source address from the own hierarchy branch are not expected. These packets should have been forwarded within the hierarchy branch. At the uplink ports these packets are discarded silently.

For example the node AF49.89H-.----.---- in the Figure above would not expect a packet with the source address AF49.8945.5419.9014 at an uplink port. Hence this packet will be discarded.

7. Redundancy

The hierarchical structure implied by the addressing leads to the fact that node failures have more implications the higher the hierarchy of a node. Therefore, a node should be equipped with at least two redundant uplink ports. Each of them is connected to a next higher hierarchy node, each of them having again at least two redundant uplinks.

In the case of nodes with ten downlinks and two uplinks the number of nodes grows with the power of two and the number of terminals grows with the power of ten. A three-dimensional network is constructed

with up to n hierarchies and up to 2^n redundancy planes. With 14 hierarchies the number of redundancy planes becomes 16384. This number of top hierarchy nodes sounds very high, but distributed around the world would lead to well-balanced redundancy.

With two or more uplinks a routing feature emerges in the network. In other words, each node has to take a routing decision in upstream direction, when forwarding packets to one the uplinks. This decision should be based on node-local information (autarkic) to avoid routing protocols. One option is learning prefixes from packets received in downstream direction.

8. IANA Considerations

In Q2/2021 a local field trial with ultra-low latency routing starts in Germany. A temporary /16 prefix "AF49" will be requested from the national registry or RIR. Later, extension of the field trial to other countries is planned. The other countries will apply for "AF33" for France, "AF44" for UK, "AF43" for Austria and so on.

9. Numbering Considerations

The international telephone number format and the country prefixes are standardized by Study Group 2 of ITU-T in the Recommendation E.164. This numbering, however, specifies several exceptions such as 800 or 900 special calling codes. The numbering used for ultra-low according to this document shall have no exception at all. Hence, in future the Study Group 2 could open a new Recommendation.

When mapping a telephone number to IPv6 prefix one problem is the different length of numbers. At the one side, telephone numbers according to E.164 can have up to 15 digits and would not fit into the remaining 14 digits in case of a 2-digit escape prefix. A future ITU-T numbering recommendation could deal with that problem. At the other side, some private phone numbers are very short. For example, the city of Munich has numbers as short as +49-89-886757. Still, the private subscriber would get a /64 prefix. To solve this problem the solution is to fill the remaining part of the IPv6 prefix with 'F' digits:

AF49:8988:6757:FFFF::/64

This rule has the advantage that the reverse process of converting an IPv6 prefix back to a telephone number always works.

10. Introduction Scenario

A possible introduction scenario is explored in Germany. It gives up the ultra-low routing feature thus avoiding to build up the network with dedicated hardware routers. Instead, networked processors are used as forwarding nodes. These can be rented at low monthly costs in

data centers.

From the subscriber to the first node a WireGuard tunnel is set up. The tunnel encryption includes the source prefix of the subscriber, so that false prefixes are automatically discarded. The service can be booked at <https://innoroute.com/save> in Germany only i.e. for prefixes starting with AF49.

11. Acknowledgements

The authors would like to thank the consortium of the European research project CHARISMA for the possibility to experiment. The description of the final demonstration is available for download: <http://www.charisma5g.eu/wp-content/uploads/2015/08/D4.3-Demonstrators-Evaluation-and-Validation-vFinal.pdf>

12. Authors' Addresses

Andreas Foglar
InnoRoute GmbH
Marsstr. 14a
80335 Munich
Germany
Email: foglar@innoroute.de

Mike Parker
Wivenhoe Park, Colchester
Essex, CO3 4HG
United Kingdom
Email: mcpark@essex.ac.uk

Theodoros Rokkas
Incites S.A.R.L.
130, Route d' Arlon
Strassen L-8008
Luxembourg
Email: trokkas@incites.eu

Mikhail Khokhlov
IP Tek UG
Dircksenstr. 50
10178 Berlin
Germany
Email: info@ip-tek.net