Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 7, 2017

M. Boucadair
C. Jacquenet
Orange
S. Sivakumar
Cisco Systems
January 3, 2017

A YANG Data Model for the DS-Lite
draft-ietf-softwire-dslite-yang-02

Abstract

This document defines a YANG data model for the DS-Lite Address Family Transition Router (AFTR) and Basic Bridging BroadBand (B4) elements .

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."

This Internet-Draft will expire on July 7, 2017.

Table of Contents

1. Introduction

This document defines a data model for DS-Lite [RFC6333], using the YANG data modeling language [RFC6020]. Both the Address Family Transition Router (AFTR) and Basic Bridging BroadBand (B4) elements are covered by this specification.

DS-Lite deployment considerations are discussed in [RFC6908].

This document follows the guidelines of [RFC6087].

This document uses the common YANG types defined in [RFC6991].

1.1. Terminology

This document makes use of the terms defined in [RFC6333].

The terminology for describing YANG data models is defined in [RFC6020].

1.2. Tree Diagrams

The meaning of the symbols in these diagrams is as follows:

- Brackets "[" and "]" enclose list keys.
- Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.
- Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only).

- Symbols after data node names: "?" means an optional node, "!" a container with presence, and "*" denotes a "list" or "leaf-list".

- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").

- Ellipsis ("...") stands for contents of subtrees that are not shown.

2. DS-Lite YANG Data Model

Figure 1 depicts the YANG data model for the AFTR and B4 elements.

The model supports enabling one or more instances of the AFTR function on a device; each instance is responsible for serving a group of B4s. The data model assumes that each AFTR instance can: be enable/disabled, be provisioned with dedicated configuration data, and maintain its own mapping table. The data model assumes that pools of IPv4 addresses can be provisioned to the AFTR. These pools may be contiguous or non-contiguous. Also, it assumes that an AFTR can either assign individual port numbers or port sets.

This document assumes [RFC4787][RFC5382][RFC5508] are enabled by default. Also, the data model relies on the recommendations in [RFC6888] and [RFC7857]. In addition, the data model supports state migration as per [RFC7785].

PCP-related considerations are out of scope of the document. A YANG data model for PCP is documented in [I-D.boucadair-pcp-yang].

```
module: ietf-dslite
 +--rw dslite-config
 |  +--rw dslite-aftr-config {aftr}?
 |  |  +--rw enable?                  boolean
 |  |  +--rw dslite-aftr-instances
 |  |     +--rw dslite-aftr-instance* [id]
 |  |        +--rw id                                  uint32
 |  |        +--rw name?                               string
 |  |        +--rw dslite-aftr-ipv6-address* [address-id]
 |  |        |  +--rw address-id      uint32
 |  |        |  +--rw ipv6-address?   inet:ipv6-address
 |  |        +--rw ipv4-address?                       inet:ipv4-address
 |  |        +--rw tunnel-mtu?                         uint16
 |  |        +--rw external-ip-address-pool* [address-id]
 |  |        |  +--rw address-id          uint32
 |  |        |  +--rw external-ip-pool?   inet:ipv4-prefix
 |  |        +--rw subscriber-mask?                    uint8
 |  |        +--rw nat-mapping-type?                   enumeration
```

```
|  |        +--rw nat-filtering-type?                enumeration
|  |        +--rw port-quota                         uint16
|  |        +--rw exclude-ports* [id]
|  |        |  +--rw id                    uint16
|  |        |  +--rw (port-type)?
|  |        |     +--:(single-port-number)
|  |        |     |  +--rw single-port-number?   inet:port-number
|  |        |     +--:(port-range)
|  |        |        +--rw start-port-number?    inet:port-number
|  |        |        +--rw end-port-number?      inet:port-number
|  |        +--rw port-set
|  |        |  +--rw port-set-enable?    boolean
|  |        |  +--rw port-set-size?      uint16
|  |        |  +--rw port-set-timeout?   uint32
|  |        +--rw enable-app?                        boolean
|  |        +--rw max-softwire-per-subscriber?       uint8
|  |        +--rw transport-protocol* [transport-protocol-id]
|  |        |  +--rw transport-protocol-id    uint8
|  |        +--rw new-mappings-rate-limit?           uint32
|  |        +--rw mss-clamping
|  |        |  +--rw mss-clamping-enable?   boolean
|  |        |  +--rw mss-value?             uint16
|  |        +--rw port-randomization-enable?         boolean
|  |        +--rw port-preservation-enable?          boolean
|  |        +--rw port-parity-preservation-enable?   boolean
|  |        +--rw address-roundrobin-enable?         boolean
|  |        +--rw udp-lifetime?                      uint32
|  |        +--rw tcp-idle-timeout?                  uint32
|  |        +--rw tcp-trans-open-timeout?            uint32
|  |        +--rw tcp-trans-close-timeout?           uint32
|  |        +--rw tcp-in-syn-timeout?                uint32
|  |        +--rw fragment-min-timeout?              uint32
|  |        +--rw icmp-timeout?                      uint32
|  |        +--rw hold-down-timeout?                 uint32
|  |        +--rw v6-v4-dscp-preservation            boolean
|  |        +--rw logging-info
|  |        |  +--rw logging-enable?        boolean
|  |        |  +--rw destination-address?   inet:ip-prefix
|  |        |  +--rw destination-port?      inet:port-number
|  |        |  +--rw (protocol)?
|  |        |     +--:(syslog)
|  |        |     |  +--rw syslog?                boolean
|  |        |     +--:(ipfix)
|  |        |     |  +--rw ipfix?                 boolean
|  |        |     +--:(ftp)
|  |        |        +--rw ftp?                   boolean
|  |        +--rw notify-address-pool-usage
|  |        |  +--rw pool-id?                     uint32
```

```
|  |        |  +--rw notify-pool-hi-threshold     percent
|  |        |  +--rw notify-pool-low-threshold?   percent
|  |        +--rw ftp-alg-enable?                    boolean
|  |        +--rw tftp-alg-enable?                   boolean
|  |        +--rw sip-alg-enable?                    boolean
|  |        +--rw rtsp-alg-enable?                   boolean
|  |        +--rw h323-alg-enable?                   boolean
|  |        +--rw all-algs-enable?                   boolean
|  |        +--rw mapping-table
|  |           +--rw mapping-entry* [index]
|  |              +--rw index                  uint32
|  |              +--rw status?                enumeration
|  |              +--rw type?                  enumeration
|  |              +--rw b4-ip-address          inet:ipv6-address
|  |              +--rw internal-ip-address    inet:ipv4-prefix
|  |              +--rw internal-port
|  |              |  +--rw (port-type)?
|  |              |     +--:(single-port-number)
|  |              |     |  +--rw single-port-number?   inet:port-number
|  |              |     +--:(port-range)
|  |              |        +--rw start-port-number?    inet:port-number
|  |              |        +--rw end-port-number?      inet:port-number
|  |              +--rw external-ip-address    inet:ipv4-address
|  |              +--rw external-port
|  |              |  +--rw (port-type)?
|  |              |     +--:(single-port-number)
|  |              |     |  +--rw single-port-number?   inet:port-number
|  |              |     +--:(port-range)
|  |              |        +--rw start-port-number?    inet:port-number
|  |              |        +--rw end-port-number?      inet:port-number
|  |              +--rw transport-protocol     uint8
|  |              +--rw lifetime               uint32
|  |              +--rw v6-dscp?               uint8
|  |              +--rw internal-v4-dscp?      uint8
|  |              +--rw external-v4-dscp?      uint8
|  |              +--rw description?           string
|  +--rw dslite-b4-config {b4}?
|     +--rw enable?               boolean
|     +--rw dslite-b4-instances
|        +--rw dslite-b4-instance* [id]
|           +--rw id                          uint32
|           +--rw name?                       string
|           +--rw aftr-ipv6-addr              inet:ipv6-address
|           +--rw ipv4-address?               inet:ipv4-address
|           +--rw tunnel-mtu?                 uint16
|           +--rw v6-v4-dscp-preservation     boolean
+--rw dslite-state
   +--ro dslite-aftr-state {aftr}?
```

```
|  +--ro dslite-aftr-instances
|     +--ro dslite-aftr-instance* [id]
|        +--ro id                          int32
|        +--ro name?                       string
|        +--ro aftr-capabilities
|        |  +--ro eim-support?                          boolean
|        |  +--ro eif-support?                          boolean
|        |  +--ro edm-support?                          boolean
|        |  +--ro edf-support?                          boolean
|        |  +--ro adm-support?                          boolean
|        |  +--ro adf-support?                          boolean
|        |  +--ro pcp-support?                          boolean
|        |  +--ro subscriber-mask-support?              boolean
|        |  +--ro port-set-support?                     boolean
|        |  +--ro self-state-migration-support?         boolean
|        |  +--ro mss-clamping-support?                 boolean
|        |  +--ro port-randomization-support?           boolean
|        |  +--ro port-preservation-support?            boolean
|        |  +--ro port-parity-preservation-support?     boolean
|        |  +--ro transport-proto-capabilities* [transport-protocol-id]
|     |  |  |  +--ro transport-protocol-id    uint8
|        |  +--ro v6-v4-dscp-preservation-support?      boolean
|        |  +--ro logging-support?                      boolean
|        |  +--ro ftp-alg-support?                      boolean
|        |  +--ro tftp-support?                         boolean
|        |  +--ro sip-alg-support?                      boolean
|        |  +--ro rtsp-alg-support?                     boolean
|        |  +--ro h323-alg-support?                     boolean
|        +--ro aftr-current-config
|        |  +--ro dslite-aftr-ipv6-address* [address-id]
|        |  |  +--ro address-id      uint32
|        |  |  +--ro ipv6-address?   inet:ipv6-address
|        |  +--ro ipv4-address?                      inet:ipv4-address
|        |  +--ro tunnel-mtu?                        uint16
|        |  +--ro external-ip-address-pool* [address-id]
|        |  |  +--ro address-id          uint32
|        |  |  +--ro external-ip-pool?   inet:ipv4-prefix
|        |  +--ro subscriber-mask?                   uint8
|        |  +--ro nat-mapping-type?                  enumeration
|        |  +--ro nat-filtering-type?                enumeration
|        |  +--ro port-quota                         uint16
|        |  +--ro exclude-ports* [id]
|        |  |  +--ro id                    uint16
|        |  |  +--ro (port-type)?
|        |  |     +--:(single-port-number)
|        |  |     |  +--ro single-port-number?   inet:port-number
|        |  |     +--:(port-range)
|        |  |        +--ro start-port-number?    inet:port-number
```

```
     |        |  |        +--ro end-port-number?      inet:port-number
     |        |  +--ro port-set
     |        |  |  +--ro port-set-enable?    boolean
     |        |  |  +--ro port-set-size?      uint16
     |        |  |  +--ro port-set-timeout?   uint32
     |        |  +--ro enable-app?                          boolean
     |        |  +--ro max-softwire-per-subscriber?         uint8
     |        |  +--ro transport-protocol* [transport-protocol-id]
     |        |  |  +--ro transport-protocol-id    uint8
     |        |  +--ro new-mappings-rate-limit?             uint32
     |        |  +--ro mss-clamping
     |        |  |  +--ro mss-clamping-enable?   boolean
     |        |  |  +--ro mss-value?             uint16
     |        |  +--ro port-randomization-enable?           boolean
     |        |  +--ro port-preservation-enable?            boolean
     |        |  +--ro port-parity-preservation-enable?     boolean
     |        |  +--ro address-roundrobin-enable?           boolean
     |        |  +--ro udp-lifetime?                        uint32
     |        |  +--ro tcp-idle-timeout?                    uint32
     |        |  +--ro tcp-trans-open-timeout?              uint32
     |        |  +--ro tcp-trans-close-timeout?             uint32
     |        |  +--ro tcp-in-syn-timeout?                  uint32
     |        |  +--ro fragment-min-timeout?                uint32
     |        |  +--ro icmp-timeout?                        uint32
     |        |  +--ro hold-down-timeout?                   uint32
     |        |  +--ro v6-v4-dscp-preservation              boolean
     |        |  +--ro logging-info
     |        |  |  +--ro logging-enable?        boolean
     |        |  |  +--ro destination-address?   inet:ip-prefix
     |        |  |  +--ro destination-port?      inet:port-number
     |        |  |  +--ro (protocol)?
     |        |  |     +--:(syslog)
     |        |  |     |  +--ro syslog?                boolean
     |        |  |     +--:(ipfix)
     |        |  |     |  +--ro ipfix?                 boolean
     |        |  |     +--:(ftp)
     |        |  |        +--ro ftp?                   boolean
     |        |  +--ro notify-address-pool-usage
     |        |  |  +--ro pool-id?                     uint32
     |        |  |  +--ro notify-pool-hi-threshold     percent
     |        |  |  +--ro notify-pool-low-threshold?   percent
     |        |  +--ro ftp-alg-enable?                    boolean
     |        |  +--ro tftp-alg-enable?                   boolean
     |        |  +--ro sip-alg-enable?                    boolean
     |        |  +--ro rtsp-alg-enable?                   boolean
     |        |  +--ro h323-alg-enable?                   boolean
     |        |  +--ro all-algs-enable?                   boolean
     |        +--ro mapping-table
```

```
   |        |  +--ro mapping-entry* [index]
   |        |     +--ro index                  uint32
   |        |     +--ro status?                enumeration
   |        |     +--ro type?                  enumeration
   |        |     +--ro b4-ip-address          inet:ipv6-address
   |        |     +--ro internal-ip-address    inet:ipv4-prefix
   |        |     +--ro internal-port
   |        |     |  +--ro (port-type)?
   |        |     |     +--:(single-port-number)
   |        |     |     |  +--ro single-port-number?   inet:port-number
   |        |     |     +--:(port-range)
   |        |     |        +--ro start-port-number?    inet:port-number
   |        |     |        +--ro end-port-number?      inet:port-number
   |        |     +--ro external-ip-address    inet:ipv4-address
   |        |     +--ro external-port
   |        |     |  +--ro (port-type)?
   |        |     |     +--:(single-port-number)
   |        |     |     |  +--ro single-port-number?   inet:port-number
   |        |     |     +--:(port-range)
   |        |     |        +--ro start-port-number?    inet:port-number
   |        |     |        +--ro end-port-number?      inet:port-number
   |        |     +--ro transport-protocol     uint8
   |        |     +--ro lifetime               uint32
   |        |     +--ro v6-dscp?               uint8
   |        |     +--ro internal-v4-dscp?      uint8
   |        |     +--ro external-v4-dscp?      uint8
   |        |     +--ro description?           string
   |        +--ro statistics
   |        |  +--ro traffic-statistics
   |        |  |  +--ro sent-packet?      yang:zero-based-counter64
   |        |  |  +--ro sent-byte?        yang:zero-based-counter64
   |        |  |  +--ro rcvd-packet?      yang:zero-based-counter64
   |        |  |  +--ro rcvd-byte?        yang:zero-based-counter64
   |        |  |  +--ro dropped-packet?   yang:zero-based-counter64
   |        |  |  +--ro dropped-byte?     yang:zero-based-counter64
   |        |  +--ro mapping-table-stats
   |        |     +--ro current-mt-size?       yang:zero-based-counter64
   |        |     +--ro max-mt-size?           uint32
   |        |     +--ro total-tcp-mappings?    uint32
   |        |     +--ro total-udp-mappings?    uint32
   |        |     +--ro total-icmp-mappings?   uint32
   |        +--ro available-capacity-client?   percent
   |        +--ro available-capacity-ext?      percent
   |        +--ro address-pool-in-use?         percent
   |        +--ro port-in-use?                 percent
   +--ro dslite-b4-state {b4}?
      +--ro dslite-b4-instances
         +--ro dslite-b4-instance* [id]
```

```
            +--ro id                int32
            +--ro name?             string
            +--ro b4-capabilities
            |  +--ro ipv4-address-modify?              boolean
            |  +--ro tunnel-mtu-support?               boolean
            |  +--ro v6-v4-dscp-preservation-support   boolean
            +--ro b4-state
               +--ro status?                           boolean
               +--ro aftr-ipv6-addr                    inet:ipv6-address
               +--ro ipv4-address-configured?          inet:ipv4-address
               +--ro v6-v4-dscp-preservation-enabled?  boolean
```

Figure 1: YANG Data Model for DS-Lite AFTR

The following notifications are supported. These notifications are triggered by configurable parameters.

```
notifications:
  +---n dslite-aftr-event
     +--ro id?           -> /dslite-state/dslite-aftr-state/
     |                   dslite-aftr-instances/dslite-aftr-instance/id
     +--ro notify-pool-threshold  percent
```

3. DS-Lite YANG Module

```
<CODE BEGINS> file "ietf-dslite@2017-01-03"
module ietf-dslite {
    namespace "urn:ietf:params:xml:ns:yang:ietf-dslite";
    prefix dslite;

    import ietf-inet-types { prefix inet; }
    import ietf-yang-types { prefix yang; }

    organization "Softwire Working Group";
    contact
     "Mohamed Boucadair <mohamed.boucadair@orange.com>
      Christian Jacquenet <christian.jacquenet@orange.com>
      Senthil Sivakumar <ssenthil@cisco.com>";

   description
      "This module is a YANG module for DS-Lite AFTR
      implementations.

      Copyright (c) 2016 IETF Trust and the persons identified as
      authors of the code.  All rights reserved.

      Redistribution and use in source and binary forms, with or
      without modification, is permitted pursuant to, and subject
```

```
        to the license terms contained in, the Simplified BSD License
        set forth in Section 4.c of the IETF Trust's Legal Provisions
        Relating to IETF Documents
        (http://trustee.ietf.org/license-info).

        This version of this YANG module is part of RFC XXXX; see
        the RFC itself for full legal notices.";

      revision 2017-01-03 {
        description "Fixed a compilation error:
          https://github.com/mbj4668/pyang/issues/296.";
         reference "-ietf-02";
      }

      revision 2016-11-14 {
        description "Integrates the comments from Ian:
        add B4 module, add an MSS leaf, add more details about
        logging protocols, and other edits.";
         reference "-ietf-01";
      }

      revision 2016-07-27 {
        description "-00 IETF version.";
         reference "-ietf-00";
      }

      revision 2016-06-13 {
        description "Update the module.";
         reference "-04";
      }

      revision 2015-12-16 {
        description "Fix an error.";
         reference "-03";
      }

      revision 2015-09-01 {
        description "Add port threshold notifications.";
         reference "-02";
      }

      revision 2015-08-31 {
        description "Fix a timeout issue.";
         reference "-01";
      }

      revision 2015-08-17 {
        description "First spec.";
```

```
        reference "-00";
      }

  // Typedef

  typedef percent {
      type uint8 {
              range "0 .. 100";
      }
      description
          "Percentage";
  }

  // FEATURES

    feature aftr {
      description
      "An AFTR element is the combination of
      an IPv4-in-IPv6 tunnel endpoint and an
      IPv4-IPv4 NAT implemented on the same node.";

     reference
        "RFC6333";
    }

    feature b4 {
      description
      "The B4 element is a function implemented
      on a dual-stack-capable node, either a directly
      connected device or a CPE, that creates
      a tunnel to an AFTR.";

     reference
        "RFC6333";
    }

  /*
   * Grouping
   */

  // port numbers: single or port range

  grouping port-number {
      description
          "Individual port or a range of ports.";

      choice port-type {
          default single-port-number;
```

```
          description
              "Port type: single or port-range.";

          case single-port-number {
              leaf single-port-number {
                  type inet:port-number;
                  description
                      "Used for single port numbers.";
              }
          }

          case port-range {
              leaf start-port-number {
                  type inet:port-number;
                  description
                      "Beginning of the port range.";
              }

              leaf end-port-number {
                  type inet:port-number;
                  description
                      "End of the port range.";
              }
          }
      }
  }

  // Timeout variables

  grouping lifetime {
     description
      "Configure values of various timeouts.";

      leaf udp-lifetime {
          type uint32;
          units "seconds";
          default 120;
          description
              "UDP inactivity timeout [RFC4787].";
      }

      leaf tcp-idle-timeout {
          type uint32;
          units "seconds";
          default 7440;
          description
              "TCP Idle timeout as per [RFC5382] should be no
               more than 2 hours and 4 minutes.";
```

```
      }

      leaf tcp-trans-open-timeout {
          type uint32;
          units "seconds";
          default 240;
          description
            "The value of the transitory open connection
            idle-timeout.
            Section 2.1 of [RFC7857] clarifies that a NAT
            should provide different configurable
            parameters for configuring the open and
            closing idle timeouts.
            To accommodate deployments that consider
            a partially open timeout of 4 minutes as being
            excessive from a security standpoint, a NAT may
            allow the configured timeout to be less than
            4 minutes.
            However, a minimum default transitory connection
            idle-timeout of 4 minutes is recommended.";
      }

      leaf tcp-trans-close-timeout {
          type uint32;
          units "seconds";
          default 240;
          description
            "The value of the transitory close connection
            idle-timeout.
            Section 2.1 of [RFC7857] clarifies that a NAT
            should provide different configurable
            parameters for configuring the open and
            closing idle timeouts.";
      }

      leaf tcp-in-syn-timeout {
          type uint32;
          units "seconds";
          default 6;
          description
            "6 seconds, as defined in [RFC5382].";
      }

      leaf fragment-min-timeout {
          type uint32;
          units "seconds";
          default 2;
          description
```

```
            "As long as the AFTR has available resources,
            the AFTR allows the fragments to arrive
            over fragment-min-timeout interval.
            The default value is inspired from RFC6146.";
        }

        leaf icmp-timeout {
            type uint32;
            units "seconds";
            default 60;
            description
                "60 seconds, as defined in [RFC5508].";
        }

        leaf hold-down-timeout {
            type uint32;
            units "seconds";
            default 120;
            description
                "Hold down timer. Ports in the
                hold down pool are not reassigned until
                this timer expires.
                The length of time and the maximum
                number of ports in this state must be
                configurable by the administrator
                [RFC6888]. This is necessary in order
                to prevent collisions between old
                and new mappings and sessions. It ensures
                that all established sessions are broken
                instead of redirected to a different peer.
                The default value is defined in REQ#8
                from [RFC6888].";
        }
    }
    // AFTR Parameters

    grouping aftr-parameters {

        description
           "A set of AFTR parameters";

        list dslite-aftr-ipv6-address {

            key address-id;

            description
              "set one or multiple IP addresses for
               the dslite-aftr";
```

```
          leaf address-id {
             type uint32;
             description
                "The identifier of the address";
          }

           leaf ipv6-address {
             type inet:ipv6-address;
             description
               "IPv6 address of the dslite-aftr.";
          }
      }
      leaf ipv4-address {
          type inet:ipv4-address;
          default "192.0.0.1";
          description
            "IPv4 address of the DS-Lite AFTR.
            192.0.0.1 is reserved for the AFTR element
            [RFC6333].
            This address can be used to report ICMP
            problems and will appear in traceroute
            outputs.";
      }

      leaf tunnel-mtu {
          type uint16;
          description
              "Configures a tunnel MTU.
              [RFC6908] specifies that since
              fragmentation and reassembly is not
              optimal, the operator should do
              everything possible to eliminate
              the need for it.  If the operator uses
              simple IPv4-in-IPv6 softwire, it is
              recommended that the MTU size of the IPv6
              network between the B4 and the AFTR
              accounts for the additional overhead
              (40 bytes).";
      }

      list external-ip-address-pool {

          key address-id;

          description
           "Pool of external IP addresses used to service
            internal hosts.
            Both contiguous and non-contiguous pools
```

```
            can be configured to an AFTR.
            Refer to REQ-3 of [RFC6888].";

        leaf address-id {
            type uint32;
            description
              "An identifier of the address.";
        }

        leaf external-ip-pool {
            type inet:ipv4-prefix;
            description
              "An IPv4 prefix used by the AFTR
               for NAT purposes.";
        }
    }

    leaf subscriber-mask {
        type uint8 {
            range "0 .. 128";
            }
        default "56";
        description
            "The subscriber-mask is an integer that indicates
             the length of significant bits to be applied on
             the source IPv6 address (internal side) to
             unambiguously identify a CPE.

             Subscriber-mask is a system-wide configuration
             parameter that is used to enforce generic
             per-subscriber policies (e.g., port-quota).

             The enforcement of these generic policies does not
             require the configuration of every subscriber's prefix.

             Example: suppose the 2001:db8:100:100::/56 prefix is
             assigned to a DS-Lite enabled CPE. Suppose also that the
             2001:db8:100:100::1 is the IPv6 address used by the
             B4 that resides in that CPE. When the AFTR
             receives a packet from this client,
             it applies the subscriber-mask (e.g., 56) on
             the source IPv6 address to compute the associated prefix
             for this client (that is 2001:db8:100:100::/56).  Then,
             the AFTR enforces policies based on that prefix
             (2001:db8:100:100::/56), not on the exact
             source IPv6 address [RFC7785].";
    }
```

```
     leaf nat-mapping-type {
             type enumeration {
                     enum "eim"  {
                         description
                             "Endpoint-Independent-Mapping.
                             Refer to Section 4 of [RFC4787].";
                           }

                      enum "adm"  {
                          description
                             "Address-Dependent-Mapping.
                              Refer to Section 4 of [RFC4787].";
                            }

                        enum "edm"  {
                            description
                              "address-and-port-Dependent-Mapping.
                               Refer to Section 4 of [RFC4787].";
                            }
                         }
                     description
                        "Indicates the type of the NAT mapping.";
      }

      leaf nat-filtering-type {
               type enumeration {
                     enum "eif"  {
                        description
                           "Endpoint-Independent-Filtering.
                           Refer to Section 5 of [RFC4787].";
                         }

                      enum "adf"  {
                           description
                              "Address-Dependent-Filtering.
                              Refer to Section 5 of [RFC4787].";
                      }

                      enum "edf"  {
                        description
                            "address-and-port-Dependent-Filtering.
                             Refer to Section 5 of [RFC4787].";
                            }
                       }
            description
              "Indicates the type of the NAT filtering.";
      }
```

```
      leaf port-quota {
          type uint16;
          mandatory true;
          description
            "Configures a port quota to be assigned per
            subscriber.
            According to [RFC6888], per-subscriber limits
            must be configurable by the administrator.";
      }

      list exclude-ports {
          key "id";
          description
             "The set of ports not to be assigned
              by the AFTR.";

          leaf id {
              type uint16;
              description
                "An identifier";
          }

          uses port-number;
      }

      container port-set {
          description
            "Manages port-set assignments.";

          leaf port-set-enable {
            type boolean;
             description
              "Enable/Disable port set assignment.";
          }

          leaf port-set-size {
            type uint16;
            description
              "Indicates the size of assigned port sets.";
          }

          leaf port-set-timeout {
            type uint32;
            description
            "Inactivity timeout for port sets.";
          }
      }
```

```
      leaf enable-app {
          type boolean;
          default true;
          description
          "Enable/disable the IP address
          pooling behavior of Paired (APP).
          APP is recommended in REQ-2 from
          [RFC4787].";
      }

      leaf max-softwire-per-subscriber {
          type uint8;
          default 1;
          description
          "Configures the maximum softwire per subscriber
          feature as per Section 4 of [RFC7785].

          A subscriber is uniquely identified by means
          of subscriber-mask.

          This policy aims to prevent a misbehaving
          subscriber from mounting several DS-Lite
          softwires that would consume additional AFTR
          resources (e.g., get more external ports if
          the quota were enforced on a per-softwire basis,
          consume extra processing due to a large number
          of active softwires).";
      }

      list transport-protocol {
          key "transport-protocol-id";
          description
            "Set of (transport) protocols supported by
            the AFTR. Default must be set to
            TCP and UDP.";

          leaf transport-protocol-id {
              type uint8;
              description
                 "Identifier of the transport protocol.
                 IANA Protocol Numbers maintained in
                 http://www.iana.org/assignments/
                 protocol-numbers are used.";
          }
      }

      leaf new-mappings-rate-limit {
          type uint32;
```

```
          description
           "Rate-limit sessions per subscriber.
           The goal is to prevent a single subscriber
           from consuming excessive CPU resources from
           the AFTR.";
      }

      container mss-clamping {
          description
            "Manages port-set assignments.";

          leaf mss-clamping-enable {
             type boolean;
             description
               "Enable/disable MSS rewriting feature.";
          }

          leaf mss-value {
            type uint16;
            units "octets";
            description
              "Sets the MSS value to be used for
               MSS rewriting.";
          }
       }


      leaf port-randomization-enable {
          type boolean;
          description
          "Enable/disable port randomization feature.
          Section 9 of [RFC7857] specifies that a NAT
          should follow the recommendations in
          Section 4 of RFC6056.";
      }

      leaf port-preservation-enable {
          type boolean;
          description
           "Indicates whether the AFTR should
           preserve the internal port number.";
      }

      leaf port-parity-preservation-enable {
          type boolean;
          description
            "Indicates whether the AFTR should
            preserve the port parity of the
```

```
            internal port number.
            Section 8 of [RFC7857] indicates that
            a NAT may disable port parity preservation
            for all dynamic mappings.
            It also specifies that a NAT should support
            means to explicitly request to preserve
            port parity (e.g., [RFC7753]).";
        }

        leaf address-roundrobin-enable {
           type boolean;
           description
              "Enable/disable address allocation
              round robin.";
        }

        uses lifetime;

        leaf v6-v4-dscp-preservation {
            type boolean;
            mandatory true;
            description
             "Copies the DSCP value from the IPv6 header
             and vice versa.
             According to Section 2.10 of [RFC6908],
             operators should use this model
             by provisioning the network such that
             the AFTR copies the DSCP value in the IPv4
             header to the Traffic Class field in
             the IPv6 header, after the encapsulation
             for the downstream traffic.";
        }

        container logging-info {
             description
                "Information about AFTR logging events.";

            leaf logging-enable {
                 type boolean;
                 description
                   "Enable logging features as per Section 2.3
                   of [RFC6908].";
            }

            leaf destination-address {
                 type inet:ip-prefix;
                 description
                     "Address of the collector that receives
```

```
                 the logs.";
          }

          leaf destination-port {
               type inet:port-number;
               description
                  "Destination port of the collector.";
          }

      choice protocol {

          description
              "Enable the protocol to be used for
               the retrieval of logging entries.";

          case syslog {
              leaf syslog {
                  type boolean;
                  description
                      "Used if SYSLOG is in use.";
              }
          }

          case ipfix {
              leaf ipfix {
                  type boolean;
                  description
                      "Used if IPFIX is in use.";
              }
          }

          case ftp {
              leaf ftp {
                  type boolean;
                  description
                      "Used if FTP is in use.";
              }
          }
       }
      }

      container notify-address-pool-usage {
            description
               "Notification of Pool usage when certain criteria
                is met.";

            leaf pool-id {
                 type uint32;
```

```
                description
                    "Pool-ID for which the notification criteria is
                     defined.";
            }

            leaf notify-pool-hi-threshold {
                type percent;
                mandatory true;
                description
                  "Notification must be generated when the defined
                  high threshold is reached. For example, if a
                  notification is required when the pool utilization
                  reaches 90%, this configuration parameter must be
                  set to 90%.";
            }

            leaf notify-pool-low-threshold {
                type percent;
                description
                  "Notification must be generated when the defined
                  low threshold is reached. For example, if a
                  notification is required when the pool utilization
                  reaches below 10%, this configuration parameter
                  must be set to 10%.";
            }
        }

            leaf ftp-alg-enable {
                type boolean;
                description
                   "Enable/Disable FTP ALG.";
            }

            leaf tftp-alg-enable {
                type boolean;
                description
                   "Enable/Disable TFTP ALG.";
            }

            leaf sip-alg-enable {
                type boolean;
                description
                   "Enable/Disable SIP ALG.";
            }

            leaf rtsp-alg-enable {
                type boolean;
                description
```

```
                   "Enable/Disable RTSP ALG.";
            }

            leaf h323-alg-enable {
                 type boolean;
                 description
                    "Enable/Disable H323 ALG.";
            }

            leaf all-algs-enable {
                 type boolean;
                 description
                    "Enable/Disable all the ALGs.";
            }
  }

  // Mapping Entry (Extended NAT44 mapping Entry)

  grouping mapping-entry {
     description
       "A DS-Lite AFTR mapping entry.";

     leaf index {
         type uint32;
         description
           "A unique identifier of a mapping entry.";
     }

     leaf status {
          type enumeration {

          enum "disabled"  {
             description
                "The mapping entry is not in use (Disabled).";
            }

          enum "assigned" {
             description
                "This mapping has been granted by the server.";
          }

          enum "stale" {
             description
                "This is a stale mapping (case of reboot).";
          }
         }
         description
            "Indicates the status of a mapping entry.";
```

```
      }

     leaf type {
          type enumeration {

          enum "static"  {
             description
                "The mapping entry is manually configured.";
            }

          enum "implicit" {
             description
                "This mapping is created by an outgoing packet.";
          }

          enum "explicit" {
             description
              "This is a dynamic explicit mapping created as a result
              of a PCP operation.";
          }
         }
         description
            "Indicates the type of a mapping entry. E.g.,
            a mapping can be: static, dynamic implicit, or
            dynamic explicit.";
      }

      leaf b4-ip-address {
          type inet:ipv6-address;
          mandatory true;
          description
              "Corresponds to the IPv6 address
              used by the B4 element.";
       }

      leaf internal-ip-address {
          type inet:ipv4-prefix;
          mandatory true;
          description
           "Corresponds to the source IPv4 address
            of the IPv4 packet conveyed over the softwire.";
       }

      container internal-port {
          description
             "Corresponds to the source port of the
             IPv4 packet conveyed over the softwire.";
           uses port-number;
```

```
      }

     leaf external-ip-address {
          type inet:ipv4-address;
          mandatory true;
          description
           "External IPv4 address assigned by the AFTR.";
     }

     container external-port {
          description
              "External port number assigned by the AFTR.";
          uses port-number;
     }

     leaf transport-protocol {
         type uint8;
         mandatory true;
         description
           "Upper-layer protocol associated with this mapping.
            Values are taken from the IANA protocol registry.
            For example, this field contains 6 (TCP) for a TCP
            mapping or 17 (UDP) for a UDP mapping.";
     }

     leaf lifetime {
          type uint32;
          units "seconds";
          mandatory true;
          description
            "Lifetime of the mapping.";
     }

     leaf v6-dscp {
          type uint8;
          description
            "DSCP value used at the softwire level
            (i.e., IPv6 header).";
     }

     leaf internal-v4-dscp {
          type uint8;
          description
            "DSCP value of the encapsulated IPv4 packet.";
     }

     leaf external-v4-dscp {
          type uint8;
```

```
          description
            "DSCP value of the translated IPv4 packet
             as marked by the AFTR.";
     }

     leaf description {
          type string;
          description
           "A description string associated with the mapping.";
     }
  }

  /*
    * DS-Lite AFTR Configuration
    */

  container dslite-config {

      description
       "AFTR and B4 configuration.";

  container dslite-aftr-config {
      if-feature aftr;

      description
       "dslite-aftr";

      leaf enable {
          type boolean;
          description
           "Enable/Disable dslite-aftr function.";
      }

      container dslite-aftr-instances {
         description
           "dslite-aftr instances";

          list dslite-aftr-instance {
              key "id";
                description
                  "A dslite-aftr instance.";

             leaf id {
                type uint32;
                description
                  "dslite-aftr instance identifier.";
             }
```

```
            leaf name {
               type string;
               description
                 "A name associated with the dslite-aftr instance.";
            }

            uses aftr-parameters;

            container mapping-table {
               description
                 "dslite-aftr mapping table maintained by
                   the dslite-aftr server.";

                 list mapping-entry {
                   key "index";
                   description
                     "dslite-aftr mapping entry.";
                   uses mapping-entry;
                 }
             }
         }
     }
  }

  /*
    * DS-Lite B4 Configuration
    */
  container dslite-b4-config {
      if-feature b4;
      description
       "dslite-b4";

      leaf enable {
          type boolean;
          description
           "Enable/Disable dslite-b4 function.";
      }

      container dslite-b4-instances {
         description
           "dslite-b4 instances";

          list dslite-b4-instance {
              key "id";
                description
                  "a dslite-b4 instance.";

             leaf id {
```

```
               type uint32;
               description
                 "dslite-b4 instance identifier.";
             }

             leaf name {
               type string;
               description
                 "A name associated with the dslite-b4 instance.";
             }


           leaf aftr-ipv6-addr {
               type inet:ipv6-address;
               mandatory true;
               description
                  "The AFTR's IPv6 address.";
             }


      leaf ipv4-address {
          type inet:ipv4-address;
          default "192.0.0.2";
          description
            "IPv4 address of the DS-Lite B4.
            192.0.0.0/29 is reserved for the B4 element
            [RFC6333].
            This address can be used to report ICMP
            problems and will appear in traceroute
            outputs.";
      }

      leaf tunnel-mtu {
          type uint16;
          description
              "Configures a tunnel MTU.
              [RFC6908] specifies that since
              fragmentation and reassembly is not
              optimal, the operator should do
              everything possible to eliminate
              the need for it.  If the operator uses
              simple IPv4-in-IPv6 softwire, it is
              recommended that the MTU size of the IPv6
              network between the B4 and the AFTR
              accounts for the additional overhead
              (40 bytes).";
      }
```

```
      leaf v6-v4-dscp-preservation {
          type boolean;
          mandatory true;
          description
           "Copies the DSCP value from the IPv6 header
           and vice versa.
           According to Section 2.10 of [RFC6908],
           operators should use this model
           by provisioning the network such that
           the AFTR copies the DSCP value in the IPv4
           header to the Traffic Class field in
           the IPv6 header, after the encapsulation
           for the downstream traffic.";
      }
      }
    }
   }
  }

  /*
    * DS-Lite State
    */

  container dslite-state {

      description
         "dslite-aftr and b4 state.";

  /*
   * DS-Lite AFTR State
   */

  container dslite-aftr-state {

      if-feature aftr;

      config false;

      description
         "dslite-aftr";

      container dslite-aftr-instances {
          description
              "dslite-aftr instances";

          list dslite-aftr-instance {
              key "id";
```

```
            description
             "dslite-aftr instance";

            leaf id {
                type int32;
                 description
                  "The identifier of the dslite-aftr instance.";
            }

            leaf name {
                type string;
                description
                 "The name of the dslite-aftr instance.";
            }

            container aftr-capabilities {
               description
                  "AFTR capabilities";

               leaf eim-support {
                      type boolean;
                      description
                        "Indicates whether EIM is enabled.";
               }

               leaf eif-support {
                      type boolean;
                      description
                        "Indicates whether EIF is enabled.";
               }

               leaf edm-support {
                      type boolean;
                      description
                        "Indicates whether EDM is enabled.";
               }

               leaf edf-support {
                      type boolean;
                      description
                        "Indicates whether EDF is enabled.";
               }

               leaf adm-support {
                      type boolean;
                      description
                        "Indicates whether ADM is enabled.";
               }
```

```
                leaf adf-support {
                       type boolean;
                       description
                         "Indicates whether ADF is enabled.";
                }

                leaf pcp-support {
                       type boolean;
                       description
                         "Indicates whether a PCP server is enabled.";
                }

                leaf subscriber-mask-support{
                    type boolean;
                    description
                      "Indicates whether the subscriber-mask feature
                       is supported.";
                 }

                leaf port-set-support {
                   type boolean;
                   description
                    "Indicates whether port set assignment is
                    supported.";
                }

                leaf self-state-migration-support {
                   type boolean;
                   description
                    "Indicates whether mappings migration
                    to the new IPv6 address used by the B4 is
                    supported [RFC7785].";
                }

                leaf mss-clamping-support {
                       type boolean;
                       description
                         "Indicates whether the MSS clamping
                         feature is supported.";
                }

                leaf port-randomization-support {
                   type boolean;
                   description
                    "Indicates whether port randomization is
                    supported.";
                }
```

```
                leaf port-preservation-support {
                   type boolean;
                   description
                    "Indicates whether port preservation
                    is supported.";
                }

                leaf port-parity-preservation-support {
                   type boolean;
                   description
                    "Indicates whether port parity preservation is
                     supported.";
                }

                list transport-proto-capabilities {
                  key "transport-protocol-id";
                  description
                    "A set of supported transport protocols.";

                  leaf transport-protocol-id {
                     type uint8;
                     description
                       "ID of the transport protocol.";
                  }
                }

                leaf v6-v4-dscp-preservation-support {
                       type boolean;
                       description
                         "Copy the DSCP value from the IPv6 header
                         and vice versa.";
                }

                leaf logging-support {
                    type boolean;
                    description
                      "Indicates whether a logging feature is
                      supported.";
                }

           leaf ftp-alg-support {
                type boolean;
                description
                   "Indicates whether FTP ALG is supported.";
           }

           leaf tftp-support {
                type boolean;
```

```
                description
                   "Indicates whether TFTP ALG is supported.";
             }

             leaf sip-alg-support {
                type boolean;
                description
                   "Indicates whether SIP ALG is supported.";
             }

             leaf rtsp-alg-support {
                type boolean;
                description
                   "Indicates whether RTSP ALG is supported.";
             }

             leaf h323-alg-support {
                type boolean;
                description
                   "Indicates whether H323 ALG is supported.";
             }
               }

              container aftr-current-config {
                 description
                      "current config";

                 uses aftr-parameters;
              }

              container mapping-table {
                  description
                      "Mapping table";
                  list mapping-entry {
                      key "index";
                      description
                          "mapping entry";
                      uses mapping-entry;
                      }
              }

              container statistics {
                  description
                       "traffic statistics";

                  container traffic-statistics {
                              description
                                "Generic traffic statistics.";
```

```
                leaf sent-packet {
                  type yang:zero-based-counter64;
                   description
                    "Number of packets sent.";
                }

                leaf sent-byte {
                  type yang:zero-based-counter64;
                  description
                    "Counter for sent traffic in bytes.";
                }

                leaf rcvd-packet {
                  type yang:zero-based-counter64;
                  description
                     "Number of received packets.";
                }

                leaf rcvd-byte {
                  type yang:zero-based-counter64;
                  description
                   "Counter for received traffic
                   in bytes.";
                }

                leaf dropped-packet {
                  type yang:zero-based-counter64;
                  description
                   "Number of dropped packets.";
                }

                leaf dropped-byte {
                  type yang:zero-based-counter64;
                  description
                    "Counter for dropped traffic in
                    bytes.";
                }
              }

              container mapping-table-stats {
                description
                 "Mapping table statistics.";

            leaf current-mt-size {
               type yang:zero-based-counter64;
               description
                 "Size of the mapping table.";
               }
```

```
        leaf max-mt-size {
          type uint32;
               description
                 "Maximum configured size of the
                 mapping table.";
           }

          leaf total-tcp-mappings {
               type uint32;
               description
                "Total number of TCP Mappings present
                at the time.";
          }
          leaf total-udp-mappings {
               type uint32;
               description
                "Total number of UDP Mappings present
                at the time.";
          }

          leaf total-icmp-mappings {
               type uint32;
               description
                "Total number of ICMP Mappings present
                at the time.";
          }
      }
    }

    leaf available-capacity-client {
         type percent;
         description
          "Ratio of available capacity in the
          customer-facing interfaces.";
    }

    leaf available-capacity-ext {
         type percent;
         description
          "Ratio of available capacity in the
          Internet-facing interfaces.";
    }

    leaf address-pool-in-use {
         type percent;
         description
          "Ratio of the shared address pool.";
    }
```

```
                    leaf port-in-use {
                         type percent;
                         description
                          "Ratio of the port usage.";
                    }
               }
          }

     }

     /*
      * DS-Lite B4 State
      */

     container dslite-b4-state {

         if-feature b4;

         config false;

         description
            "dslite-b4";

         container dslite-b4-instances {
             description
                 "dslite-b4 instances";

             list dslite-b4-instance {
                 key "id";

                 description
                  "dslite-b4 instance";

                 leaf id {
                     type int32;
                      description
                       "The identifier of the dslite-b4 instance.";
                 }

                 leaf name {
                     type string;
                     description
                      "The name of the dslite-b4 instance.";
                 }

                 // B4 Capabilities

                 container b4-capabilities {
```

```
                description
                    "B4 capabilities";

                leaf ipv4-address-modify {
                    type boolean;
                    description
                        "Indicates whether it is possible
                         to configure an IPv4 address
                         for the B4 element.";
                }

                leaf tunnel-mtu-support {
                    type boolean;
                    description
                        "Indicates whether it is possible to
                    configure a tunnel MTU.
                    ";
                }

                leaf v6-v4-dscp-preservation-support {
                    type boolean;
                    mandatory true;
                    description
                        "Indicates whether it supports
                    DSCP preservation.
                    ";
                }
            }

          // B4 State

           container b4-state {
               description
                   "B4 capabilities";

               leaf status {
                   type boolean;
                   description
                       "Indicates whether the instance is
                   enabled or disabled.
                   ";
               }

               leaf aftr-ipv6-addr {
                   type inet:ipv6-address;
                   mandatory true;
                   description
                       "The AFTR's IPv6 address.";
```

```
                }

                leaf ipv4-address-configured {
                    type inet:ipv4-address;
                    default "192.0.0.2";
                    description
                        "The B4's IPv4 address.";
                }

                leaf v6-v4-dscp-preservation-enabled {
                    type boolean;
                    description
                        "Indicates whether this feature is
                    enabled/disabled.
                    ";
                }
            }
        }
       }
    }
   }

   /*
    * Notifications
    */

     notification dslite-aftr-event {

        description
          "Notifications must be generated when the defined
          high/low threshold is reached. Related configuration
          parameters must be provided to trigger
          the notifications.";

        leaf id {
            type leafref {
             path
               "/dslite-state/dslite-aftr-state/dslite-aftr-instances/"
               + "dslite-aftr-instance/id";
            }
                  description
                    "AFTR instance ID.";
         }

        leaf notify-pool-threshold {
                  type percent;
                  mandatory true;
                  description
```

```
                    "A threshold has been fired.";
        }
    }
  }
  <CODE ENDS>
```

4. Security Considerations

The YANG module defined in this memo is designed to be accessed via the NETCONF protocol [RFC6241]. The lowest NETCONF layer is the secure transport layer and the support of SSH is mandatory to implement secure transport [RFC6242]. The NETCONF access control model [RFC6536] provides means to restrict access for particular NETCONF users to a pre-configured subset of all available NETCONF protocol operations and contents.

All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default). These data nodes are considered sensitive. Write operations (e.g., edit-config) applied to these data nodes without proper protection can negatively affect network operations.

5. IANA Considerations

This document requests IANA to register the following URI in the "IETF XML Registry" [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-dslite
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
```

This document requests IANA to register the following YANG module in the "YANG Module Names" registry [RFC6020].

```
name: ietf-dslite
namespace: urn:ietf:params:xml:ns:yang:ietf-dslite
prefix: dslite
reference: RFC XXXX
```

6. Acknowledgements

Thanks to Q. Wu for identifying a compiling error.

Many thanks to Ian Farrer for the review and comments.

7. References

7.1. Normative references

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <http://www.rfc-editor.org/info/rfc3688>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <http://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <http://www.rfc-editor.org/info/rfc6333>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <http://www.rfc-editor.org/info/rfc6991>.

7.2. Informative references

[I-D.boucadair-pcp-yang] Boucadair, M., Jacquenet, C., Sivakumar, S., and S. Vinapamula, "YANG Data Models for the Port Control Protocol (PCP)", draft-boucadair-pcp-yang-03 (work in progress), November 2016.

[RFC4787] Audet, F., Ed. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, DOI 10.17487/RFC4787, January 2007, <http://www.rfc-editor.org/info/rfc4787>.

[RFC5382] Guha, S., Ed., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, DOI 10.17487/RFC5382, October 2008, <http://www.rfc-editor.org/info/rfc5382>.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, DOI 10.17487/RFC5508, April 2009, <http://www.rfc-editor.org/info/rfc5508>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <http://www.rfc-editor.org/info/rfc6087>.

[RFC6888] Perreault, S., Ed., Yamagata, I., Miyakawa, S., Nakagawa, A., and H. Ashida, "Common Requirements for Carrier-Grade NATs (CGNs)", BCP 127, RFC 6888, DOI 10.17487/RFC6888, April 2013, <http://www.rfc-editor.org/info/rfc6888>.

[RFC6908] Lee, Y., Maglione, R., Williams, C., Jacquenet, C., and M. Boucadair, "Deployment Considerations for Dual-Stack Lite", RFC 6908, DOI 10.17487/RFC6908, March 2013, <http://www.rfc-editor.org/info/rfc6908>.

[RFC7753] Sun, Q., Boucadair, M., Sivakumar, S., Zhou, C., Tsou, T., and S. Perreault, "Port Control Protocol (PCP) Extension for Port-Set Allocation", RFC 7753, DOI 10.17487/RFC7753, February 2016, <http://www.rfc-editor.org/info/rfc7753>.

[RFC7785] Vinapamula, S. and M. Boucadair, "Recommendations for Prefix Binding in the Context of Softwire Dual-Stack Lite", RFC 7785, DOI 10.17487/RFC7785, February 2016, <http://www.rfc-editor.org/info/rfc7785>.

[RFC7857] Penno, R., Perreault, S., Boucadair, M., Ed., Sivakumar, S., and K. Naito, "Updates to Network Address Translation (NAT) Behavioral Requirements", BCP 127, RFC 7857, DOI 10.17487/RFC7857, April 2016, <http://www.rfc-editor.org/info/rfc7857>.

Authors' Addresses

Mohamed Boucadair
Orange
Rennes 35000
France

EMail: mohamed.boucadair@orange.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Senthil Sivakumar
Cisco Systems
7100-8 Kit Creek Road
Research Triangle Park, North Carolina 27709
USA

Phone: +1 919 392 5158
EMail: ssenthil@cisco.com