Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2016

M. Boucadair
C. Jacquenet
Orange
S. Sivakumar
Cisco Systems
June 13, 2016

YANG Data Model for the DS-Lite Address Family Transition Router (AFTR)

draft-boucadair-softwire-dslite-yang-04

Abstract

This document defines a YANG data model for the DS-Lite Address Family Transition Router (AFTR).

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 December 15, 2016.

Copyright Notice

Copyright (c) 2016 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.

Table of Contents

1. Introduction 2
1.1. Terminology 2
1.2. Tree Diagrams 2
2. DS-Lite AFTR YANG Data Model 3
3. AFTR YANG Module 8
4. Security Considerations 28
5. IANA Considerations 28
6. Acknowledgements 28
7. References 28
7.1. Normative references 28
7.2. Informative references 29
Authors' Addresses 30

1. Introduction

This document defines a data model for DS-Lite [RFC6333] architectures, using the YANG data modeling language [RFC6020]. Only the Address Family Transition Router (AFTR) element is covered by this specification. As a reminder, [RFC6334] can be used to configure the name of the AFTR to a B4 element.

This document assumes [RFC4787][RFC5382][RFC5508] are enabled by default.

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:

- o Brackets "[" and "]" enclose list keys.

- o Curly braces "{" and "}" contain names of optional features that make the corresponding node conditional.

- o Abbreviations before data node names: "rw" means configuration (read-write), "ro" state data (read-only).

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

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

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

2. DS-Lite AFTR YANG Data Model

Figure 1 depicts the YANG data model for the AFTR element.

A device can enable multiple AFTR instances; each responsible to service a group of B4s. Also, it assumes that each AFTR instance can: be enable/disabled, be provisioned with a dedicated configuration data, and maintain its own mapping table. This 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. Both features are supported in the YANG data model.

This model supports state migration as per [RFC7785].

PCP-related considerations are out of scope of the document.

```
module: ietf-dslite-aftr
 +--rw dslite-aftr-config
 |  +--rw enable?                  boolean
 |  +--rw dslite-aftr-instances
 |     +--rw dslite-aftr-instance* [id]
 |        +--rw id                                uint32
 |        +--rw name?                             string
 |        +--rw dslite-aftr-ip-address* [address-id]
 |        |  +--rw address-id       uint32
 |        |  +--rw ip-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 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-enable?               boolean
 |        +--rw port-randomization-enable?         boolean
 |        +--rw port-preservation-enable?          boolean
 |        +--rw port-parity-preservation-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-enable?                    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
 +--ro dslite-aftr-state
    +--ro dslite-aftr-instances
       +--ro dslite-aftr-instance* [id]
          +--ro id                     int32
          +--ro name?                  string
          +--ro aftr-capabilities
          |  +--ro pcp-support?                       boolean
          |  +--ro subscriber-mask-support?           boolean
          |  +--ro port-set-support?                  boolean
          |  +--ro self-state-migration?              boolean
          |  +--ro mss-clamping-support?              boolean
          |  +--ro port-randomization-support?        boolean
          |  +--ro port-preservation-suport?          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-ip-address* [address-id]
          |  |  +--ro address-id          uint32
          |  |  +--ro ip-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 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-enable?               boolean
          |  +--ro port-randomization-enable?         boolean
          |  +--ro port-preservation-enable?          boolean
          |  +--ro port-parity-preservation-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-enable?                    boolean
          |  +--ro notify-address-pool-usage
          |     +--ro pool-id?                     uint32
          |     +--ro notify-pool-hi-threshold     percent
          |     +--ro notify-pool-low-threshold?   percent
          +--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 available-capacity-client?   percent
          +--ro available-capacity-ext?      percent
          +--ro address-pool-in-use?         percent
          +--ro port-in-use?                 percent
```

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-aftr-state/dslite-aftr-instances/
     |                      dslite-aftr-instance/id
     +--ro notify-pool-threshold  percent
```

3. AFTR YANG Module

```
<CODE BEGINS> file "ietf-dslite-aftr@2016-06-13.yang"
module ietf-dslite-aftr {
    namespace "urn:ietf:params:xml:ns:yang:ietf-dslite-aftr";
    prefix dslite-aftr;

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

    organization "xxxx Working Group";
    contact
     "Mohamed Boucadair <mohamed.boucadair@orange.com>
      Christian Jacquenet <christian.jacquenet@orange.com>
      Senthil Sivakumar (ssenthil) <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 2015-12-16 {
      description "Update the module.";
       reference "-04";
    }
    revision 2015-12-16 {
      description "Fix an error.";
       reference "-03";
    }

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

    revision 2015-08-31 {
```

```
         description "Fix a timeout issue.";
          reference "-01";
       }

       revision 2015-08-17 {
         description "Changes xxxx.";
          reference "-00";
       }

   // Typedef

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

   /*
    * 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
                       "Begining 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;
            default 120;
            description
                "UDP inactivity timeout.";
        }

        leaf tcp-idle-timeout {
            type uint32;
            default 7440;
            description
                "TCP Idle timeout as per RFC 5382 should be no
                 more than 2 hours and 4 minutes.";
        }

        leaf tcp-trans-open-timeout {
            type uint32;
            default 240;
            description
                "The value of the transitory open connection
                idle-timeout.";
        }

        leaf tcp-trans-close-timeout {
            type uint32;
            default 240;
            description
              "The value of the transitory close connection
               idle-timeout.";
        }

        leaf tcp-in-syn-timeout {
            type uint32;
```

```
            default 6;
            description
              "6 seconds, as defined in [RFC5382].";
        }

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

        leaf hold-down-timeout {
            type uint32;
            default 120;
            description
                "Hold down timer. Ports in the
                hold down pool are not reassigned till
                this timer expires.";
        }
    }
    // 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;
            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.";
        }

        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.";

            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-subscriberpolicies (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 port-quota {
            type uint16;
            mandatory true;
            description
              "Configures a port quota to be assigned per
              subscriber.";
        }

        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
              "Inactivty timeout for port sets.";
            }
        }

        leaf enable-app {
            type boolean;
            default true;
            description
            "Enable/disable Address Pooling of Paired
            (APP) feature.";
        }

        leaf max-softwire-per-subscriber {
            type uint8;
            default 1;
            description
            "Configures the maximum softwire per subscriber
             feature.";
        }
```

```
       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.";
           }
       }

       leaf new-mappings-rate-limit {
           type uint32;
           description
            "Rate-limit sessions per subscriber.";
       }

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

       leaf port-randomization-enable {
           type boolean;
           description
           "Enable/disable port randomization feature.";
       }

       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.";
       }

       uses lifetime;
```

```
       leaf v6-v4-dscp-preservation {
           type boolean;
           mandatory true;
           description
            "Copies the DSCP value from the IPv6 header
            and vice versa.";
       }

       leaf logging-enable {
           type boolean;
           description
            "Enable logging features.";
       }

       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
                    notificationis 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;
```

```
           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 enacapsulated 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-aftr-config {
       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 AFTR State
    */

   container dslite-aftr-state {

       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 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 {
                     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 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.";
                        }
                }
        }
    }

    /*
     * 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-aftr-state/dslite-aftr-instances/"
               + "dslite-aftr-instance/id";
            }
                  description
                    "AFTR instance ID.";
         }

        leaf notify-pool-threshold {
                  type percent;
                  mandatory true;
                  description
                    "A treshhold 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-aftr
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-aftr
namespace: urn:ietf:params:xml:ns:yang:ietf-dslite-aftr
prefix: dslite-aftr
reference: RFC XXXX
```

6. Acknowledgements

Thanks to Q. Wu for identifying a compiling error.

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

[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>.

[RFC6334] Hankins, D. and T. Mrugalski, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Option for Dual-Stack Lite", RFC 6334, DOI 10.17487/RFC6334, August 2011, <http://www.rfc-editor.org/info/rfc6334>.

[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>.

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