

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

Table of Contents

- [1. Introduction](#)
- [2. Terminologies](#)
- [3. APN Configuration data model](#)
 - [3.1. APN YANG Model Structure](#)
 - [3.2. APN ID Template](#)
 - [3.3. APN ID Marking](#)
 - [3.4. APN Policy Mapping](#)
- [4. APN YANG Module](#)
- [5. IANA Considerations](#)
- [6. Security Considerations](#)
- [7. Acknowledgements](#)
- [8. Normative References](#)
- [Authors' Addresses](#)

1. Introduction

Application-aware Networking (APN) is introduced in [[I-D.li-apn-framework](#)] and [[I-D.li-apn-problem-statement-usecases](#)]. APN data packets convey the APN attribute (incl. APN ID and/or APN Parameters). The APN ID is a structured value, treated as an opaque object in the network, to which the network operator applies policies in various nodes/service functions along the path so to provide corresponding services. In an IPv6 network, a design proposal of such structured value can refer to [[I-D.li-apn-header](#)]. The APN attribute can be encapsulated in various data plane adopted within a Network Operator controlled limited domain, e.g. IPv6, MPLS, and other tunnel technologies, which wait to be further specified.

This document defines a data model for APN using the YANG data modeling language [[RFC7950](#)]. This YANG model supports the APN Attribute options [[I-D.li-apn-framework](#)].

The modeling in this document complies with the Network Management Datastore Architecture (NMDA) defined in [[RFC8342](#)].

2. Terminologies

APN: Application-aware Networking

APN ID: APN Identifier

The terminology for describing YANG data models is found in [[RFC7950](#)].

Tree diagrams used in this document follow the notation defined in [[RFC8340](#)].

3. APN Configuration data model

3.1. APN YANG Model Structure

The APN YANG model includes the data plane protocol indication, the global actions, the apn-id-template configuration, the apn-id-marking, and the mapping policies for APN. The structure of the APN YANG model is shown in Figure 1.

The APN YANG model can cover several data plane protocols. In this model, only IPv6 is presented.

One global action is defined currently, i.e., the apn-id-inherit, which is used to configure the APN ID carried in the inner packet to be inherited (copied and encapsulated) into the outer tunnel header.

The apn-id-templates configures the templates of the APN ID. More than one templates can be configured.

The apn-id-marking configures the APN ID on the flow which is identified by the selected filter.

The mapping-policies configures the APN ID based on the selected template, and the to-be-mapped-into policy based on the configured APN ID. More than one policies can be configured.

```

module: ietf-apn
+--rw apn!
  +--rw ipv6!
    +--rw global
      | +--rw apn-id-inherit?  apn-id-inherit-type
    +--rw apn-id-templates
      | +--rw apn-id-template* [name]
      |   +--rw name          string
      |   +--rw app-info-fields!
      |     | +--rw app-fields
      |     |   +--rw app-field* [index]
      |     |     +--rw index    uint32
      |     |     +--rw name     string
      |     |     +--rw length?  uint32
      |     +--rw user-info-fields!
      |       +--rw user-fields
      |         +--rw user-field* [index]
      |           +--rw index    uint32
      |           +--rw name     string
      |           +--rw length?  uint32
    +--rw apn-id-marking!
      | +--rw filter
      | | +--rw filter-type?  apn-filter-type
      | | +--rw ace-name?     -> /acl:acls/acl/aces/ace/name
      | +--rw apn-ipv6-template  -> /apn/ipv6/apn-id-templates/apn-
      | +--rw app-fields
      | | +--rw app-field* [name]
      | |   +--rw name      -> /apn/ipv6/apn-id-templates/apn-id-tem
      | |   +--rw value    uint32
      | +--rw user-fields
      |   +--rw user-field* [name]
      |     +--rw name      -> /apn/ipv6/apn-id-templates/apn-id-tem
      |     +--rw value    uint32
    +--rw mapping-policys
      +--rw mapping-policy* [color]
        +--rw color          uint32
        +--rw name           string
        +--rw description?   string
        +--rw apn-id-template? -> /apn/ipv6/apn-id-templates/apn
        +--rw apn-ipv6-maps
          +--rw apn-ipv6-map* [index]
            +--rw index      uint32
            +--rw app-fields
            | +--rw app-field* [name]
            |   +--rw name    -> /apn/ipv6/apn-id-templates/a
            |   +--rw value  uint32
            +--rw user-fields
            | +--rw user-field* [name]

```

```

|     +--rw name      -> /apn/ipv6/apn-id-templates/a
|     +--rw value     uint32
+--rw (match-tunnel)
  +--:(sr-policy)
  |   +--rw color?    uint32
  +--:(ip)
    +--rw native-ip? empty

```

Figure 1. APN YANG Model Structure

3.2. APN ID Template

The APN ID template can be configured with the defined fields, including the app-info-fields and the user-info-fields, each of which can have several fields with their name and length configurable.

```

+--rw apn-id-templates
  +--rw apn-id-template* [name]
    +--rw name          string
    +--rw app-info-fields!
      |   +--rw app-fields
      |     +--rw app-field* [index]
      |       +--rw index    uint32
      |       +--rw name     string
      |       +--rw length?  uint32
    +--rw user-info-fields!
      +--rw user-fields
        +--rw user-field* [index]
          +--rw index    uint32
          +--rw name     string
          +--rw length?  uint32

```

3.3. APN ID Marking

The APN ID Marking uses the selected filter to identify the flow on which APN is applied. Multiple filter types exist. ACL [\[RFC8519\]](#) is a common way to specify a flow.

Upon the identified flow, the APN template is used to configure the APN ID with the defined fields, including the app-info-fields and the user-info-fields, each of which can have several fields with their name and length configurable.

```

+--rw apn-id-marking!
  +--rw filter
  | +--rw filter-type?   apn-filter-type
  | +--rw ace-name?     -> /acl:acls/acl/aces/ace/name
+--rw apn-ipv6-template  -> /apn/ipv6/apn-id-templates/apn-
+--rw app-fields
  | +--rw app-field* [name]
  |   +--rw name       -> /apn/ipv6/apn-id-templates/apn-id-tem
  |   +--rw value      uint32
+--rw user-fields
  +--rw user-field* [name]
    +--rw name        -> /apn/ipv6/apn-id-templates/apn-id-tem
    +--rw value       uint32

```

3.4. APN Policy Mapping

The APN policy mapping is for mapping to corresponding policies based on the APN ID being structured with the configured fields. The mapping into SR policy is presented in the model below.

```

+--rw mapping-policys
  +--rw mapping-policy* [color]
    +--rw color          uint32
    +--rw name           string
    +--rw description?   string
    +--rw apn-id-template? -> /apn/ipv6/apn-id-templates/apn
+--rw apn-ipv6-maps
  +--rw apn-ipv6-map* [index]
    +--rw index          uint32
    +--rw app-fields
    | +--rw app-field* [name]
    |   +--rw name       -> /apn/ipv6/apn-id-templates/a
    |   +--rw value      uint32
    +--rw user-fields
    | +--rw user-field* [name]
    |   +--rw name       -> /apn/ipv6/apn-id-templates/a
    |   +--rw value      uint32
    +--rw (match-tunnel)
      +--:(sr-policy)
      | +--rw color?     uint32
      +--:(ip)
        +--rw native-ip? empty

```

4. APN YANG Module

```

module ietf-apn {
  namespace "urn:ietf:params:xml:ns:yang:ietf-apn";
  prefix apn;

  import ietf-access-control-list {
    prefix "acl";
    reference
      "RFC 8519: YANG Data Model for Network Access Control
      Lists (ACLs)";
  }

  organization
    "APN";

  contact
    "Web: <https://datatracker.ietf.org/wg/apn/about/>
    WG List: <apn@ietf.org>
    Editor: pengshuping@huawei.com;

  description
    "This YANG module specifies a vendor-independent data
    model for the Application-aware Networking (APN).

    Copyright (c) 2020 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 2021-10-20 {
    description "Initial revision.";
    reference "draft-peng-apn-yang";
  }

/*
* IDENTITIES
*/
  identity base-filter {
    description
      "Base identity to represent a filter. A filter is used to
      specify the flow to mark the APN ID. ";
  }

```



```

}

identity acl-filter {
    base base-filter;
    description
        "Apply ACL rules to specify the flow.";
}

/*
 * TYPE DEFINITIONS
 */

typedef apn-id-inherit-type {
    type enumeration {
        enum "enable" {
            value 1;
            description
                "Inherit the APN ID.";
        }
        enum "disable" {
            value 2;
            description
                "Not inherit the APN ID.";
        }
    }
    description
        "APN ID inherit type.";
}

typedef template-state-type {
    type enumeration {
        enum "unavailable" {
            value 0;
            description
                "The APN ID template is unavailable.";
        }
        enum "available" {
            value 1;
            description
                "The APN ID template is available.";
        }
    }
    description
        "APN ID template state type.";
}

typedef apn-filter-type {
    type identityref {
        base base-filter;
    }
}

```

```

    }
    description
        "Specifies a known type of filter.";
    }
}

/*
 * GROUP DEFINITIONS
 */

grouping apn-filter {
    description "A grouping for APN filter definition";

    leaf filter-type {
        type apn-filter-type;
        description "filter type";
    }

    leaf ace-name {
        when "../filter-type = 'apn:acl-filter'";
        type leafref {
            path "/acl:acls/acl:acl/acl:aces/acl:ace/acl:name";
        }
        description "Access Control Entry name.";
    }
}

container apn {
    presence "Enter apn view.";
    description
        "Application-aware Networking.";
    container ipv6 {
        presence "Enter apn-ipv6 view.";
        description
            "Application-aware Networking IPV6.";
        container global {
            description
                "Configure APN6 global config.";
            leaf apn-id-inherit {
                type apn-id-inherit-type;
                description
                    "Enable/disable APN ID inherit.";
            }
        }
    }
    container apn-id-templates {
        description
            "List of APN ID templates.";
        list apn-id-template {
            key "name";
            description

```

```

    "Configure an APN ID template.";
leaf name {
  type string {
    length "1..31";
    pattern '^[^ \?]*';
  }
  description
    "APN ID template name.";
}

container app-info-fields {
  presence "Enter app-info-fields view.";
  description
    "APP information fields.";
  container app-fields {
    description
      "List of APP fields.";
    list app-field {
      key "index";
      unique "name";
      max-elements "4";
      description
        "Configure an APP field.";
      leaf index {
        type uint32 {
          range "1..255";
        }
        description
          "APP field index.";
      }
      leaf name {
        type string {
          length "1..15";
          pattern '^[^ \?]*';
        }
        must "not(..../..../user-info-fields/user-fields/use
          mandatory true;
        description
          "APP field name.";
      }
      leaf length {
        type uint32 {
          range "1..32";
        }
        default "16";
        description
          "APP field length.";
      }
    }
  }
}

```

```

    }
  }
  container user-info-fields {
    presence "Enter user-info-fields view.";
    description
      "User information fields.";
    container user-fields {
      description
        "List of user fields.";
      list user-field {
        key "index";
        unique "name";
        max-elements "4";
        description
          "Configure an user field.";
        leaf index {
          type uint32 {
            range "1..255";
          }
          description
            "User field index.";
        }
        leaf name {
          type string {
            length "1..15";
            pattern '^[^ \?]*';
          }
          must "not(..../..../..../app-info-fields/app-fields/app-f
            mandatory true;
          description
            "User field name.";
        }
        leaf length {
          type uint32 {
            range "1..32";
          }
          default "16";
          description
            "APP field length.";
        }
      }
    }
  }
} //apn-id-templates

  container apn-id-marking {
    presence "Enter user-info-fields view.";
    description

```

```

"Configure apn id marking.";

    container filter {
uses apn-filter;
description
    "The filter which is used to indicate the flow to apply
    APN.";
    }

    leaf apn-ipv6-template {
type leafref {
    path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn:apn-id-templa
}
mandatory true;
description
    "APN IPv6 template.";
}
container app-fields {
description
    "List of APP fields.";
list app-field {
    key "name";
    max-elements "4";
description
    "Configure an APP field.";
leaf name {
    type leafref {
        path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn:apn-id-te
    }
description
    "APP field name.";
}
leaf value {
    type uint32 {
        range "1..4294967295";
    }
    mandatory true;
description
    "APP field value.";
}
}
}
container user-fields {
description
    "List of user fields.";
list user-field {
    key "name";
    max-elements "4";
description

```

```

    "Configure an user field.";
  leaf name {
    type leafref {
      path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn:apn-id-te
    }
    description
      "User field name.";
  }
  leaf value {
    type uint32 {
      range "1..4294967295";
    }
    mandatory true;
    description
      "User field value.";
  }
}
}
} // apn-id-marking

```

```

container mapping-policys {
  description
    "List of mapping policys.";
  list mapping-policy {
    key "color";
    unique "name";
    description
      "Configure a mapping policy.";
    leaf color {
      type uint32 {
        range "0..4294967295";
      }
      description
        "Color of a mapping policy.";
    }
    leaf name {
      type string {
        length "1..31";
        pattern '^[^ \?]*';
      }
      mandatory true;
      description
        "Mapping policy name.";
    }
    leaf description {
      type string {
        length "1..242";
      }
      description

```

```

    "Description of a mapping policy.";
}

leaf apn-id-template {
  /// when "../match-type='apn-ipv6'";
  type leafref {
    path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn:apn-id-
  }
  must "(count(/apn:apn/apn:ipv6/apn:apn-id-templates/apn:ap
description
  "APN ID template.";
}

container apn-ipv6-maps {
  /// when "../match-type='apn-ipv6'";
  description
    "List of APN IPv6 maps.";
  list apn-ipv6-map {
    key "index";
    description
      "Configure an APN IPv6 map.";
    leaf index {
      type uint32 {
        range "1..4294967295";
      }
      must "((../index = 4294967295 and (count(../app-fields
description
      "Index.";
    }
  container app-fields {
    when "../index != 4294967295";
    description
      "List of APP fields.";
    list app-field {
      key "name";
      max-elements "4";
      description
        "Configure an APP field.";
      leaf name {
        type leafref {
          path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn
        }
        description
          "APP field name.";
      }
      leaf value {
        type uint32 {
          range "1..4294967295";
        }
      }
    }
  }
}

```

```

        mandatory true;
        description
            "APP field value.";
    }
}
}
container user-fields {
    when "../index != 4294967295";
    description
        "List of user fields.";
    list user-field {
        key "name";
        max-elements "4";
        description
            "Configure an user field.";
        leaf name {
            type leafref {
                path "/apn:apn/apn:ipv6/apn:apn-id-templates/apn
            }
            description
                "User field name.";
        }
        leaf value {
            type uint32 {
                range "1..4294967295";
            }
            mandatory true;
            description
                "User field value.";
        }
    }
}
}
choice match-tunnel {
    mandatory true;
    description
        "Match tunnel.";
    case sr-policy {
        description
            "Flow match sr-policy.";
        leaf color {
            type uint32 {
                range "0..4294967295";
            }
            must "not(..../apn-ipv6-map[color=current()]][inde
            description
                "Color of an SR Policy.";
        }
    }
}
}
case ip {

```


preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations.

7. Acknowledgements

The authors would like to thank the careful reviews and valuable comments from Mengdi Li, Qingyu Guan, Sheng Fang, and Stefano Previdi.

8. Normative References

[I-D.li-6man-app-aware-ipv6-network]

Li, Z., Peng, S., Li, C., Xie, C., Voyer, D., Li, X., Liu, P., Liu, C., and K. Ebisawa, "Application-aware IPv6 Networking (APN6) Encapsulation", Work in Progress, Internet-Draft, draft-li-6man-app-aware-ipv6-network-03, 22 February 2021, <<https://datatracker.ietf.org/doc/html/draft-li-6man-app-aware-ipv6-network-03>>.

[I-D.li-apn-framework] Li, Z., Peng, S., Voyer, D., Li, C., Liu, P., Cao, C., and G. S. Mishra, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-li-apn-framework-07, 3 April 2023, <<https://datatracker.ietf.org/doc/html/draft-li-apn-framework-07>>.

[I-D.li-apn-header] Li, Z., Peng, S., and S. Zhang, "Application-aware Networking (APN) Header", Work in Progress, Internet-Draft, draft-li-apn-header-04, 12 April 2023, <<https://datatracker.ietf.org/doc/html/draft-li-apn-header-04>>.

[I-D.li-apn-problem-statement-usecases]

Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z., and G. S. Mishra, "Problem Statement and Use Cases of Application-aware Networking (APN)", Work in Progress, Internet-Draft, draft-li-apn-problem-statement-usecases-08, 3 April 2023, <<https://datatracker.ietf.org/doc/html/draft-li-apn-problem-statement-usecases-08>>.

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

[RFC6536]

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

[RFC7950]

Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.

[RFC8340]

Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <<https://www.rfc-editor.org/info/rfc8340>>.

[RFC8342]

Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

[RFC8519]

Jethanandani, M., Agarwal, S., Huang, L., and D. Blair, "YANG Data Model for Network Access Control Lists (ACLs)", RFC 8519, DOI 10.17487/RFC8519, March 2019, <<https://www.rfc-editor.org/info/rfc8519>>.

Authors' Addresses

Shuping Peng
Huawei Technologies
Beijing
China

Email: pengshuping@huawei.com

Zhenbin Li
Huawei Technologies
Beijing
China

Email: lizhenbin@huawei.com

Ran Pang
China Unicom
Beijing
China

Email: pangran@chinaunicom.cn

Huiyue Zhang
China Telecom
Beijing

China

Email: zhanghy30@chinatelecom.cn