Routing Area Common YANG Data Types
draft-ietf-rtgwg-routing-types-09

Abstract

This document defines a collection of common data types using the YANG data modeling language. These derived common types are designed to be imported by other modules defined in the routing area.

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 February 20, 2018.

Copyright Notice

Copyright (c) 2017 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.
1. Introduction

The YANG [RFC6020] [RFC7950] is a data modeling language used to model configuration data, state data, Remote Procedure Calls, and notifications for network management protocols. The YANG language supports a small set of built-in data types and provides mechanisms to derive other types from the built-in types.

This document introduces a collection of common data types derived from the built-in YANG data types. The derived types are designed to be the common types applicable for modeling in the routing area.

1.1. Terminology

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

2. Overview

This document defines the two models for common routing types, ietf-routing-types and iana-routing-types. The only module imports are from [RFC6991]. The ietf-routing-types model contains common routing types other than those corresponding directly to IANA mappings. These include:

router-id
Router Identifiers are commonly used to identify a node in routing and other control plane protocols. An example usage of router-id can be found in [I-D.ietf-ospf-yang].

route-target
Route Targets (RTs) are commonly used to control the distribution of virtual routing and forwarding (VRF) information, see [RFC4364], in support of BGP/MPLS IP virtual private networks (VPNs) and BGP/MPLS Ethernet VPNs [RFC7432]. An example usage can be found in [I-D.ietf-bess-l2vpn-yang].

ipv6-route-target
IPv6 Route Targets (RTs) are similar to standard Route Targets only they IPv6 Address Specific BGP Extended Communities as described in [RFC5701]. An IPv6 Route Target is 20 octets and includes an IPv6 address as the global administrator.

route-target-type
This type defines the import and export rules of Route Targets, as described in Section 4.3.1 of [RFC4364]. An example usage can be found in [I-D.ietf-idr-bgp-model].

route-distinguisher
Route Distinguishers (RDs) are commonly used to identify separate routes in support of virtual private networks (VPNs). For example, in [RFC4364], RDs are commonly used to identify independent VPNs and VRFs, and more generally, to identify multiple routes to the same prefix. An example usage can be found in [I-D.ietf-idr-bgp-model].

route-origin
Route Origin is commonly used to indicate the Site of Origin for routing and forwarding (VRF) information, see [RFC4364], in support of BGP/MPLS IP virtual private networks (VPNs) and BGP/MPLS Ethernet VPNs [RFC7432]. An example usage can be found in [I-D.ietf-bess-l3vpn-yang].

ipv6-route-origin
An IPv6 Route Origin would also be used to indicate the Site of Origin for routing and forwarding (VRF) information, see [RFC4364], in support of virtual private networks (VPNs). IPv6 Route Origins are IPv6 Address Specific BGP Extended Communities as described in [RFC5701]. An IPv6 Route Origin is 20 octets and includes an IPv6 address as the global administrator.

ipv4-multicast-group-address
This type defines the representation of an IPv4 multicast group address, which is in the range from 224.0.0.0 to 239.255.255.255. An example usage can be found in [I-D.ietf-pim-yang].

ipv6-multicast-group-address
This type defines the representation of an IPv6 multicast group address, which is in the range of FF00::/8. An example usage can be found in [I-D.ietf-pim-yang].

ip-multicast-group-address
This type represents an IP multicast group address and is IP version neutral. The format of the textual representation implies the IP version. An example usage can be found in [I-D.ietf-pim-yang].

ipv4-multicast-source-address
IPv4 source address type for use in multicast control protocols. This type also allows the indication of wildcard sources, i.e., "*". An example of where this type may/will be used is [I-D.ietf-pim-yang].

ipv6-multicast-source-address
IPv6 source address type for use in multicast control protocols. This type also allows the indication of wildcard sources, i.e., "*". An example of where this type may/will be used is [I-D.ietf-pim-yang].

bandwidth-ieee-float32
Bandwidth in IEEE 754 floating point 32-bit binary format [IEEE754]. Commonly used in Traffic Engineering control plane protocols. An example of where this type may/will be used is [I-D.ietf-ospf-yang].

link-access-type
This type identifies the IGP link type. An example of where this type may/will be used is [I-D.ietf-ospf-yang].

timer-multiplier
This type is used in conjunction with a timer-value type. It is generally used to indicate define the number of timer-value intervals that may expire before a specific event must occur. Examples of this include the arrival of any BFD packets, see [RFC5880] Section 6.8.4, or hello_interval in [RFC3209]. Example of where this type may/will be used is [I-D.ietf-idr-bgp-model] and [I-D.ietf-teas-yang-rsvp].

timer-value-seconds16
This type covers timers which can be set in seconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint16 (2 octets). An example of where this type may/will be used is [I-D.ietf-ospf-yang].

timer-value-seconds32
This type covers timers which can be set in seconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint32 (4 octets). An example of where this type may/will be used is [I-D.ietf-teas-yang-rsvp].

timer-value-milliseconds
This type covers timers which can be set in milliseconds, not set, or set to infinity. This type supports a range of values that can be represented in a uint32 (4 octets). Examples of where this type may/will be used include [I-D.ietf-teas-yang-rsvp] and [I-D.ietf-bfd-yang].

percentage
This type defines a percentage with a range of 0-100%. An example usage can be found in [I-D.ietf-idr-bgp-model].

timeticks64
This type is based on the timeticks type defined in [RFC6991] but with 64-bit precision. It represents the time in hundredths of a second between two epochs. An example usage can be found in [I-D.ietf-idr-bgp-model].

uint24
This type defines a 24-bit unsigned integer. It is used by target="I-D.ietf-ospf-yang"/

generalized-label
This type represents a generalized label for Generalized Multi-Protocol Label Switching (GMPLS) [RFC3471]. The Generalized Label does not identify its type, which is known from the context. An example usage can be found in [I-D.ietf-teas-yang-te].

mpls-label-special-purpose
This type represents the special-purpose Multiprotocol Label Switching (MPLS) label values [RFC7274]. An example usage can be found in [I-D.ietf-mpls-base-yang].

mpls-label-general-use
The 20 bits label values in an MPLS label stack entry, specified in [RFC3032]. This label value does not include the encodings of Traffic Class and TTL (time to live). The label range specified by this type is for general use, with special-purpose MPLS label
values excluded. An example usage can be found in [I-D.ietf-mpls-base-yang].

mpls-label
The 20 bits label values in an MPLS label stack entry, specified in [RFC3032]. This label value does not include the encodings of Traffic Class and TTL (time to live). The label range specified by this type covers the general use values and the special-purpose label values. An example usage can be found in [I-D.ietf-mpls-base-yang].

This document defines the following YANG groupings:

mpls-label-stack
This grouping defines a reusable collection of schema nodes representing an MPLS label stack [RFC3032]. An example usage can be found in [I-D.ietf-mpls-base-yang].

vpn-route-targets
This grouping defines a reusable collection of schema nodes representing Route Target import-export rules used in the BGP enabled Virtual Private Networks (VPNs). [RFC4364][RFC4664]. An example usage can be found in [I-D.ietf-bess-l2vpn-yang].

The iana-routing-types model contains common routing types corresponding directly to IANA mappings. These include:

address-family
This type defines values for use in address family identifiers. The values are based on the IANA Address Family Numbers Registry [IANA-ADDRESS-FAMILY-REGISTRY]. An example usage can be found in [I-D.ietf-idr-bgp-model].

subsequent-address-family
This type defines values for use in subsequent address family (SAFI) identifiers. The values are based on the IANA Subsequent Address Family Identifiers (SAFI) Parameters Registry [IANA-SAFI-REGISTRY].

3. IETF Routing Types YANG Module

<CODE BEGINS> file "ietf-routing-types@2017-08-16.yang"
module ietf-routing-types {
    prefix rt-types;

    import ietf-yang-types {
        prefix yang;
    }

This module contains a collection of YANG data types considered generally useful for routing protocols.

Copyright (c) 2017 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.";

reference "RFC XXXX";

revision 2017-06-29 {
  description
    "Initial revision.";
  reference "RFC TBD: Routing YANG Data Types";
}

/*** Identities related to MPLS/GMPLS ***/

identity mpls-label-special-purpose-value {

description
"Base identity for deriving identities describing
special-purpose Multiprotocol Label Switching (MPLS) label
values.";
reference
"RFC7274: Allocating and Retiring Special-Purpose MPLS
Labels.";
}

identity ipv4-explicit-null-label {
  base mpls-label-special-purpose-value;
  description
    "This identity represents the IPv4 Explicit NULL Label.";
  reference
    "RFC3032: MPLS Label Stack Encoding. Section 2.1.";
}

identity router-alert-label {
  base mpls-label-special-purpose-value;
  description
    "This identity represents the Router Alert Label.";
  reference
    "RFC3032: MPLS Label Stack Encoding. Section 2.1.";
}

identity ipv6-explicit-null-label {
  base mpls-label-special-purpose-value;
  description
    "This identity represents the IPv6 Explicit NULL Label.";
  reference
    "RFC3032: MPLS Label Stack Encoding. Section 2.1.";
}

identity implicit-null-label {
  base mpls-label-special-purpose-value;
  description
    "This identity represents the Implicit NULL Label.";
  reference
    "RFC3032: MPLS Label Stack Encoding. Section 2.1.";
}

identity entropy-label-indicator {
  base mpls-label-special-purpose-value;
  description
    "This identity represents the Entropy Label Indicator.";
  reference
    "RFC6790: The Use of Entropy Labels in MPLS Forwarding.
    Sections 3 and 10.1.";
}

identity gal-label {
  base mpls-label-special-purpose-value;
This identity represents the Generic Associated Channel Label (GAL)."
 reference "RFC5586: MPLS Generic Associated Channel. Sections 4 and 10.";
}

identity oam-alert-label {
    base mpls-label-special-purpose-value;
    description "This identity represents the OAM Alert Label.";
    reference "RFC3429: Assignment of the 'OAM Alert Label' for Multiprotocol Label Switching Architecture (MPLS) Operation and Maintenance (OAM) Functions. Sections 3 and 6.";
}

identity extension-label {
    base mpls-label-special-purpose-value;
    description "This identity represents the Extension Label.";
    reference "RFC7274: Allocating and Retiring Special-Purpose MPLS Labels. Sections 3.1 and 5.";
}

/*** Collection of types related to routing ***/
typedef router-id {
    type yang:dotted-quad;
    description "A 32-bit number in the dotted quad format assigned to each router. This number uniquely identifies the router within an Autonomous System.";
}

/*** Collection of types related to VPN ***/
typedef route-target {
    type string {
        pattern
            ' (0:(6553[0-5]|655[0-2][0-9]|65[0-4][0-9][0-9]|6[0-4][0-9]{0,3}|[0-5]?[0-9][0-9]:429496729[0-5]|42949672[0-8][0-9]|4294967[01][0-9]{2}|429496[0-6][0-9]{3})'
A route target is an 8-octet BGP extended community initially identifying a set of sites in a BGP VPN (RFC 4364). However, it has since taken on a more general role in BGP route filtering. A route target consists of two or three fields: a 2-octet type field, an administrator field, and, optionally, an assigned number field.

According to the data formats for type 0, 1, 2, and 6 defined in RFC4360, RFC5668, and RFC7432, the encoding pattern is defined as:

0:2-octet-asn:4-octet-number
1:4-octet-ipv4addr:2-octet-number
2:4-octet-asn:2-octet-number.
6:6-octet-mac-address.

Additionally, a generic pattern is defined for future route target types:

2-octet-other-hex-number:6-octet-hex-number

Some valid examples are: 0:100:100, 1:1.1.1.1:100, 2:1234567890:203 and 6:26:00:08:92:78:00;
typedef ipv6-route-target {
  type string {
    pattern
      '((::\{0-9a-fA-F\}{0,4})::\{0-9a-fA-F\}{0,4})\{0,5\}'
      + '(((::\{0-9a-fA-F\}{0,4})?::\{0-9a-fA-F\}{0,4}))\{3\}'
      + '+\{25[0-5]|2[0-4][0-9]|1\{01\}?[0-9][0-9]\}\{0,7\}'
      + '+\{25[0-5]|2[0-4][0-9]|1\{01\}?[0-9][0-9]\}\{0,3\}'
      + '1\{0-5\}?\{0-9\}[0,3]\{0-9\}';
    pattern
      '((\[^:]+::)\{6\}((\[^:]+::)\{.*\}))*'  
      + '((\[^:]+::*\{:*\}+)?::((\[^:]+::*\{:*\}+))' 
      + '+'
      + '1\{0-5\}|\{0-9\}\{0,5\}|\{0-9\}\{0,9\}2\{3\}'
      + '1\{0-5\}?\{0-9\}[0,3]\{0-9\}';
  }
  description
  "An IPv6 route target is a 20-octet BGP IPv6 address specific extended community serving the same function as a standard 8-octet route target only allowing for an IPv6 address as the global administrator. The format is <ipv6-address:2-octet-number>.

  Some valid examples are: 2001:DB8::1:6544 and 2001:DB8::5eb1:791:6b37:17958";
}

typedef route-target-type {
  type enumeration {
    enum "import" {
      value 0;
      description
      "The route target applies to route import.";
    }
    enum "export" {
      value 1;
  }
}
description
  "The route target applies to route export.";
}
enum "both" {
  value 2;
  description
    "The route target applies to both route import and route export.";
}

description
  "Indicates the role a route target takes in route filtering.";

reference "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs).";

typedef route-distinguisher {
  type string {
    pattern
      '0:(6553[0-5] | 655[0-2] | 65[0-4] | [0-9]):(255[0-2] | 25[0-4])' |
      '1:((25[0-2] | 25[0-4])\.[25[0-4] | 25[0-4])' |
      '655[0-2]' |
      '650[0-4] | 65[0-9]' |
      '0-5)' |
      '6:(255[0-2] | 25[0-4])' |
      '7:((25[0-2] | 25[0-4])\.[25[0-4] | 25[0-4]))' |
      '8:((25[0-2] | 25[0-4])\.[25[0-4] | 25[0-4]))' |
      '9:((25[0-2] | 25[0-4])\.[25[0-4] | 25[0-4]))' |
    }
  }

"A route distinguisher is an 8-octet value used to distinguish routes from different BGP VPNs (RFC 4364). As per RFC 4360, a route distinguisher will have the same format as a route target and will consist of two or three fields including a 2-octet type field, an administrator field, and, optionally, an assigned number field.

According to the data formats for type 0, 1, 2, and 6 defined in RFC4360, RFC5668, and RFC7432, the encoding pattern is defined as:

0:2-octet-asn:4-octet-number  
1:4-octet-ipv4addr:2-octet-number  
2:4-octet-asn:2-octet-number.  
6:6-octet-mac-address.

Additionally, a generic pattern is defined for future route discriminator types:

2-octet-other-hex-number:6-octet-hex-number

Some valid examples are: 0:100:100, 1:1.1.1.1:100, 2:1234567890:203 and 6:26:00:08:92:78:00";

typedef route-origin {
  type string {
    pattern
      0: (6553[0-5]|655[0-2]|65[0-4]|6[0-4]|[0-4])[0-9]|6[0-4]|0-9)\{2\} |'  
      (1: ((([0-9]|1[0-9])|1[0-9]|1\{0\}|1\{9\}|0-9)\{2\}|2[0-4]|0-9)\{4\} |'  
      25[0-5]\{3\} |'  
      655[0-2]|0-9) |'  
      65[0-4]|0-9)\{2\}|6[0-4]|0-9)\{3\} |'
+ (0-5)?[0-9]{0,3}\{0-9\})\)
+ '2:\([0-5]+[0-9]{0,3}\{0-9\}\)|
+ '4294967\{01\}[0-9]{2}\)
+ '429496\{0-6\}\{0-9\}\{3\}|42949\{0-5\}[0-9]{4}\)
+ '4294\{0-8\}[0-9]{5}\)
+ '429\{0-3\}\{0-9\}\{6\}|42\{0-8\}[0-9]{7}\{4\}[01][0-9]{8}\)
+ '('6553\{0-5\}|655\{0-2\}\{0-9\}|65\{0-4\}[0-9]{2}\)
+ '6\{0-4\}[0-9]{3}\)
+ '6:[a-fA-F0-9]{1,2}\)
+ '(((3-57-9a-fA-F)\{1-9a-fA-F\}[0-9a-fA-F]1,3):'
+ '[0-9a-fA-F]\{1,12\})

description
"A route origin is an 8-octet BGP extended community identifying the set of sites where the BGP route originated (RFC 4364). A route target consists of two or three fields: a 2-octet type field, an administrator field, and, optionally, an assigned number field.

According to the data formats for type 0, 1, 2, and 6 defined in RFC4360, RFC5668, and RFC7432, the encoding pattern is defined as:

0:2-octet-asn:4-octet-number
1:4-octet-ipv4addr:2-octet-number
2:4-octet-asn:2-octet-number.
6:6-octet-mac-address.

Additionally, a generic pattern is defined for future route origin types:

2-octet-other-hex-number:6-octet-hex-number

Some valid examples are: 0:100:100, 1:1.1.1.1:1:100, 2:1234567890:203 and 6:26:00:08:92:78:00";

reference
"RFC4360: BGP Extended Communities Attribute. RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs) RFC5668: 4-Octet AS Specific BGP Extended Community. RFC7432: BGP MPLS-Based Ethernet VPN";

typedef ipv6-route-origin {
  type string {
    pattern
      '((4\{0-9a-fA-F\}\{0,4\})\{0,4\})\{0,5\}'
  }
}
An IPv6 route origin is a 20-octet BGP IPv6 address specific extended community serving the same function as a standard 8-octet route only allowing for an IPv6 address as the global administrator. The format is <ipv6-address:2-octet-number>.


reference
"RFC5701: IPv6 Address Specific BGP Extended Community Attribute";

/*** Collection of types common to multicast ***/

typedef ipv4-multicast-group-address {
    type inet:ipv4-address {
        pattern '2((2[4-9])|(3[0-9]))\.(.*)';
    }
}
description
"This type represents an IPv4 multicast group address, which is in the range from 224.0.0.0 to 239.255.255.255."

reference "RFC1112: Host Extensions for IP Multicasting."

}

typedef ipv6-multicast-group-address {
    type inet:ipv6-address {
        pattern '(((fF){2}[0-9a-fA-F]{2})\.)';
    }
}
description
"This type represents an IPv6 multicast group address, which is in the range of FF00::/8.";

typedef ip-multicast-group-address {
    type union {
        type ipv4-multicast-group-address;
        type ipv6-multicast-group-address;
    }
    description
        "This type represents a version-neutral IP multicast group address. The format of the textual representation implies the IP version.";
}

typedef ipv4-multicast-source-address {
    type union {
        type enumeration {
            enum "*" {
                description
                    "Any source address.";
            }
        }
        type inet:ipv4-address;
    }
    description
        "Multicast source IPv4 address type.";
}

typedef ipv6-multicast-source-address {
    type union {
        type enumeration {
            enum "*" {
                description
                    "Any source address.";
            }
        }
        type inet:ipv6-address;
    }
    description
        "Multicast source IPv6 address type.";
}

/*** Collection of types common to protocols ***/

typedef bandwidth-ieee-float32 {
    type string {
        "RFC4291": IP Version 6 Addressing Architecture. Sec 2.7.
        "RFC7346": IPv6 Multicast Address Scopes.";
    }
pattern
'0\[xX\](0((\.|0?)?[pP]\(+\)?0?|0\))|'
+ '1\((0\rightarrow{9fA-F}|0(5)?[02468aCeE])\)?([pP]\(+\)?\{0-7\})|
+ '1\{01\}[0-9][0-9]?0\{0-9\})\'));
}
description
"Bandwidth in IEEE 754 floating point 32-bit binary format:
(-1)**S * 2**(Exponent-127) * (1 + Fraction),
where Exponent uses 8 bits, and Fraction uses 23 bits.
The units are octets per second.
The encoding format is the external hexadecimal-significant character sequences specified in IEEE 754 and C99. The format is restricted to be normalized, non-negative, and non-fraction: 0xl.hhhhhhp{+}d or 0X1.HHHHHHP{+}D
where 'h' and 'H' are hexadecimal digits, ‘d’ and ‘D’ are integers in the range of [0..127].
When six hexadecimal digits are used for ‘hhhhhh’ or ‘HHHHHH’, the least significant digit must be an even number. ‘x’ and ‘X’ indicate hexadecimal; ‘p’ and ‘P’ indicate power of two. Some examples are: 0x0p0, 0x1p10, and 0x1.abcde2p+20";
reference
}
typedef link-access-type {
type enumeration {
enum "broadcast" {

description
"Specify broadcast multi-access network.";
}
enum "non-broadcast-multiaccess" {

description
"Specify Non-Broadcast Multi-Access (NBMA) network.";
}
enum "point-to-multipoint" {

description
"Specify point-to-multipoint network."
}
enum "point-to-point" {

description
"Specify point-to-point network."
}

description
"Link access type."
}
typedef timer-multiplier {
  type uint8;
  description "The number of timer value intervals that should be interpreted as a failure.";
}

typedef timer-value-seconds16 {
  type union {
    type uint16 {
      range "1..65535";
    }
    type enumeration {
      enum "infinity" {
        description "The timer is set to infinity.";
      }
      enum "not-set" {
        description "The timer is not set.";
      }
    }
  }
  units "seconds";
  description "Timer value type, in seconds (16-bit range).";
}

typedef timer-value-seconds32 {
  type union {
    type uint32 {
      range "1..4294967295";
    }
    type enumeration {
      enum "infinity" {
        description "The timer is set to infinity.";
      }
      enum "not-set" {
        description "The timer is not set.";
      }
    }
  }
  units "seconds";
  description "Timer value type, in seconds (32-bit range).";
}
typedef timer-value-milliseconds {
  type union {
    type uint32 {
      range "1..4294967295";
    }
    type enumeration {
      enum "infinity" {
        description
        "The timer is set to infinity.";
      }
      enum "not-set" {
        description
        "The timer is not set.";
      }
    }
  }
  units "milliseconds";
  description
  "Timer value type, in milliseconds.";
}

typedef percentage {
  type uint8 {
    range "0..100";
  }
  description
  "Integer indicating a percentage value";
}

typedef timeticks64 {
  type uint64;
  description
  "This type is based on the timeticks type defined in
  RFC 6991, but with 64-bit width. It represents the time,
  modulo 2^64, in hundredths of a second between two epochs.";
  reference "RFC 6991 - Common YANG Data Types";
}

typedef uint24 {
  type uint32 {
    range "0 .. 16777215";
  }
  description
  "24-bit unsigned integer";
}

/*** Collection of types related to MPLS/GMPLS ***/
typedef generalized-label {
type binary;
description
"Generalized label. Nodes sending and receiving the
Generalized Label are aware of the link-specific
label context and type.";
reference "RFC3471: Section 3.2";
}
typedef mpls-label-special-purpose {
type identityref {
  base mpls-label-special-purpose-value;
}
description
"This type represents the special-purpose Multiprotocol Label
Switching (MPLS) label values.";
reference
"RFC3032: MPLS Label Stack Encoding.
RFC7274: Allocating and Retiring Special-Purpose MPLS
Labels.";
}
typedef mpls-label-general-use {
type uint32 {
  range "16..1048575";
}
description
"The 20-bit label values in an MPLS label stack entry,
specified in RFC3032. This label value does not include
the encodings of Traffic Class and TTL (time to live).
The label range specified by this type is for general use,
with special-purpose MPLS label values excluded.";
reference "RFC3032: MPLS Label Stack Encoding.";
}
typedef mpls-label {
type union {
  type mpls-label-special-purpose;
  type mpls-label-general-use;
}
description
"The 20-bit label values in an MPLS label stack entry,
specified in RFC3032. This label value does not include
the encodings of Traffic Class and TTL (time to live).";
reference "RFC3032: MPLS Label Stack Encoding.";
}

/*** Groupings **/
grouping mpls-label-stack {
    description "A grouping that specifies an MPLS label stack. List entries are ordered with the first entry being the top of stack, the next entry being the next entry on the stack, and so on.";
    container mpls-label-stack {
        description "Container for a list of MPLS label stack entries.";
        list entry {
            key "id";
            description "List of MPLS label stack entries."
            leaf id {
                type uint8;
                description "Identifies the entry in a sequence of an MPLS label stack entries. An entry with smaller ID value is precedes an entry in the label stack with a smaller ID. The value of this id has no semantic meaning other than ordering and referencing the entry.";
            }
            leaf label {
                type rt-types:mpls-label;
                description "Label value.";
            }
            leaf ttl {
                type uint8;
                description "Time to Live (TTL)."
                reference "RFC3032: MPLS Label Stack Encoding.";
            }
            leaf traffic-class {
                type uint8 {
                    range "0..7";
                }
                description "Traffic Class (TC)."
                reference "RFC5462: Multiprotocol Label Switching (MPLS) Label Stack Entry: ‘EXP’ Field Renamed to ‘Traffic Class’ Field.";
            }
        }
    }
}
grouping vpn-route-targets
  
  description
  "A grouping that specifies Route Target import-export rules
  used in the BGP enabled Virtual Private Networks (VPNs).";

  reference
  "RFC4364: BGP/MPLS IP Virtual Private Networks (VPNs).
  RFC4664: Framework for Layer 2 Virtual Private Networks
  (L2VPNs)";

  list vpn-target {
    key "route-target";

    description
    "List of Route Targets.";

    leaf route-target {
      type rt-types:route-target;
      description
      "Route Target value";
    }

    leaf route-target-type {
      type rt-types:route-target-type;
      mandatory true;
      description
      "Import/export type of the Route Target.";
    }
  }

<CODE ENDS>

4. IANA Routing Types YANG Module

<CODE BEGINS> file "iana-routing-types@2017-06-29.yang"
module iana-routing-types {
  namespace "urn:ietf:params:xml:ns:yang:iana-routing-types";

  prefix iana-rt-types;

  organization
  "IANA";

  contact
  "Internet Assigned Numbers Authority

  Postal: ICANN
  4676 Admiralty Way, Suite 330
  Marina del Rey, CA 90292

  Tel: +1 310 823 9358
  <mailto:iana@iana.org>";

  description
"This module contains a collection of YANG data types considered defined by IANA and used for routing protocols.

Copyright (c) 2017 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.";
reference "RFC XXXX";

revision 2017-06-29 {
  description
    "Initial revision.";
  reference "RFC TBD: IANA Routing YANG Data Types";
}

/*** Collection of IANA types related to routing ***/
/*** IANA address family Identities ***/

identity address-family {
  description
    "Base identity from which identities describing address families are derived.";
}

identity ipv4 {
  base address-family;
  description
    "IPv4 Address Family - IANA Registry Assigned Number: 1";
}

identity ipv6 {
  base address-family;
  description
    "IPv6 Address Family - IANA Registry Assigned Number: 2";
}

identity nsap {
  base address-family;
  description

"OSI Network Service Access Point (NSAP) Address Family - IANA Registry Assigned Number: 3";
}

identity hdlc {
  base address-family;
  description
    "High-Level Data Link Control (HDLC) Address Family - IANA Registry Assigned Number: 4";
}

identity bbn1822 {
  base address-family;
  description
    "Bolt, Beranek, and Newman Report 1822 (BBN 1822) Address Family - IANA Registry Assigned Number: 5";
}

identity ieee802 {
  base address-family;
  description
    "IEEE 802 Committee Address Family (aka, MAC address) - IANA Registry Assigned Number: 6";
}

identity e163 {
  base address-family;
  description
    "ITU-T E.163 Address Family - IANA Registry Assigned Number: 7";
}

identity e164 {
  base address-family;
  description
    "ITU-T E.164 (SMDS, Frame Relay, ATM) Address Family - IANA Registry Assigned Number: 8";
}

identity f69 {
  base address-family;
  description
    "ITU-T F.69 (Telex) Address Family - IANA Registry Assigned Number: 9";
}

identity x121 {
  base address-family;
}
description
"ITU-T X.121 (X.25, Frame Relay) Address Family -
IANA Registry Assigned Number: 10";
}

identity ipx {
  base address-family;
  description
    "Novell Internetwork Packet Exchange (IPX)
    Address Family - IANA Registry Assigned Number: 11";
}

identity appletalk {
  base address-family;
  description
    "Apple AppleTalk Address Family -
    IANA Registry Assigned Number: 12";
}

identity decnet-iv {
  base address-family;
  description
    "Digital Equipment DECnet Phase IV Address Family -
    IANA Registry Assigned Number: 13";
}

identity vines {
  base address-family;
  description
    "Banyan Vines Address Family -
    IANA Registry Assigned Number: 14";
}

identity e164-nsap {
  base address-family;
  description
    "ITU-T E.164 with NSAP sub-address Address Family -
    IANA Registry Assigned Number: 15";
}

identity dns {
  base address-family;
  description
    "Domain Name System (DNS) Address Family -
    IANA Registry Assigned Number: 16";
}

identity distinguished-name {
base address-family;
description
"Distinguished Name Address Family -
 IANA Registry Assigned Number: 17";
}

identity as-num {
  base address-family;
description
"AS Number Family -
 IANA Registry Assigned Number: 18";
}

identity xtp-v4 {
  base address-family;
description
"Xpress Transport Protocol (XTP) over IPv4
 Address Family - IANA Registry Assigned Number: 19";
}

identity xtp-v6 {
  base address-family;
description
"Xpress Transport Protocol (XTP) over IPv4
 Address Family - IANA Registry Assigned Number: 20";
}

identity xtp-native {
  base address-family;
description
"Xpress Transport Protocol (XTP) native mode
 Address Family - IANA Registry Assigned Number: 21";
}

identity fc-port {
  base address-family;
description
"Fibre Channel (FC) World-Wide Port Name
 Address Family - IANA Registry Assigned Number: 22";
}

identity fc-node {
  base address-family;
description
"Fibre Channel (FC) World-Wide Node Name
 Address Family - IANA Registry Assigned Number: 23";
}
identity gwid {
    base address-family;
    description "ATM Gateway Identifier (GWID) Number Family -
                IANA Registry Assigned Number: 24";
}

identity l2vpn {
    base address-family;
    description "Layer-2 VPN (L2VPN) Address Family -
                IANA Registry Assigned Number: 25";
}

identity mpls-tp-section-eid {
    base address-family;
    description "MPLS-TP Section Endpoint Identifier Address Family -
                IANA Registry Assigned Number: 26";
}

identity mpls-tp-lsp-eid {
    base address-family;
    description "MPLS-TP LSP Endpoint Identifier Address Family -
                IANA Registry Assigned Number: 27";
}

identity mpls-tp-pwe-eid {
    base address-family;
    description "MPLS-TP Pseudowire Endpoint Identifier
                Address Family - IANA Registry Assigned Number: 28";
}

identity mt-v4 {
    base address-family;
    description "Multi-Topology IPv4 Address Family -
                Address Family - IANA Registry Assigned Number: 29";
}

identity mt-v6 {
    base address-family;
    description "Multi-Topology IPv6 Address Family -
                Address Family - IANA Registry Assigned Number: 30";
}
identity eigrp-common-sf {
        base address-family;
        description
        "Enhanced Interior Gateway Routing Protocol (EIGRP)
        Common Service Family Address Family -
        IANA Registry Assigned Number: 16384";
    }

identity eigrp-v4-sf {
        base address-family;
        description
        "Enhanced Interior Gateway Routing Protocol (EIGRP)
        IPv4 Service Family Address Family -
        IANA Registry Assigned Number: 16385";
    }

identity eigrp-v6-sf {
        base address-family;
        description
        "Enhanced Interior Gateway Routing Protocol (EIGRP)
        IPv6 Service Family Address Family -
        IANA Registry Assigned Number: 16386";
    }

identity lcaf {
        base address-family;
        description
        "LISP Canonical Address Format (LCAF)
        Address Family - IANA Registry Assigned Number: 16387";
    }

identity bgp-1s {
        base address-family;
        description
        "Border Gateway Protocol - Link State (BGP-LS)
        Address Family - IANA Registry Assigned Number: 16388";
    }

identity mac-48 {
        base address-family;
        description
        "IEEE 48-bit Media Access Control (MAC)
        Address Family - IANA Registry Assigned Number: 16389";
    }

identity mac-64 {
        base address-family;
        description
"IEEE 64-bit Media Access Control (MAC) Address Family - IANA Registry Assigned Number: 16390";
}

identity trill-oui {
    base address-family;
    description
        "TRILL IEEE Organizationally Unique Identifier (OUI) - Address Family - IANA Registry Assigned Number: 16391";
}

identity trill-mac-24 {
    base address-family;
    description
        "TRILL Final 3 octets of 48-bit MAC address Address Family - IANA Registry Assigned Number: 16392";
}

identity trill-mac-48 {
    base address-family;
    description
        "TRILL Final 5 octets of 64-bit MAC address Address Family - IANA Registry Assigned Number: 16393";
}

identity trill-rbridge-port-id {
    base address-family;
    description
        "TRILL Remote Bridge (RBridge) Port ID Address Family - IANA Registry Assigned Number: 16394";
}

identity trill-nickname {
    base address-family;
    description
        "TRILL Nickname Address Family - IANA Registry Assigned Number: 16395";
}

/*** SAFIs for Multi-Protocol BGP Identities ***/

identity bgp-safi {
    description
        "Base identity from which identities describing BGP Subsequent Address Family Identifier (SAFI) - RFC 4760.";
}

identity unicast-safi {

base bgp-safi;
description
"Unicast SAFI -
IANA Registry Assigned Number: 1";
}

identity multicast-safi {
  base bgp-safi;
  description
  "Multicast SAFI -
  IANA Registry Assigned Number: 2";
}

identity labeled-unicast-safi {
  base bgp-safi;
  description
  "Labeled Unicast SAFI -
  IANA Registry Assigned Number: 4";
}

identity multicast-vpn-safi {
  base bgp-safi;
  description
  "Multicast VPN SAFI -
  IANA Registry Assigned Number: 5";
}

identity pseudowire-safi {
  base bgp-safi;
  description
  "Multi-segment Pseudowire VPN SAFI -
  IANA Registry Assigned Number: 6";
}

identity tunnel-enap-safi {
  base bgp-safi;
  description
  "Tunnel Encap SAFI -
  IANA Registry Assigned Number: 7";
}

identity mcast-vpls-safi {
  base bgp-safi;
  description
  "Multicast Virtual Private LAN Service (VPLS) SAFI -
  IANA Registry Assigned Number: 8";
}
identity tunnel-safi {
    base bgp-safi;
    description
        "Tunnel SAFI -
         IANA Registry Assigned Number: 64";
}

identity vpls-safi {
    base bgp-safi;
    description
        "Virtual Private LAN Service (VPLS) SAFI -
         IANA Registry Assigned Number: 65";
}

identity mdt-safi {
    base bgp-safi;
    description
        "Multicast Distribution Tree (MDT) SAFI -
         IANA Registry Assigned Number: 66";
}

identity v4-over-v6-safi {
    base bgp-safi;
    description
        "IPv4 over IPv6 SAFI -
         IANA Registry Assigned Number: 67";
}

identity v6-over-v4-safi {
    base bgp-safi;
    description
        "IPv6 over IPv4 SAFI -
         IANA Registry Assigned Number: 68";
}

identity l1-vpn-auto-discovery-safi {
    base bgp-safi;
    description
        "Layer-1 VPN Auto Discovery SAFI -
         IANA Registry Assigned Number: 69";
}

identity evpn-safi {
    base bgp-safi;
    description
        "Ethernet VPN (EVPN) SAFI -
         IANA Registry Assigned Number: 70";
}
identity bgp-ls-safi {
    base bgp-safi;
    description "BGP Link-State (BGP-LS) SAFI - IANA Registry Assigned Number: 71";
}

identity bgp-ls-vpn-safi {
    base bgp-safi;
    description "BGP Link-State (BGP-LS) VPN SAFI - IANA Registry Assigned Number: 72";
}

identity sr-te-safi {
    base bgp-safi;
    description "Segment Routing - Traffic Engineering (SR-TE) SAFI - IANA Registry Assigned Number: 73";
}

identity labeled-vpn-safi {
    base bgp-safi;
    description "MPLS Labeled VPN SAFI - IANA Registry Assigned Number: 128";
}

identity multicast-mpls-vpn-safi {
    base bgp-safi;
    description "Multicast for BGP/MPLS IP VPN SAFI - IANA Registry Assigned Number: 129";
}

identity route-target-safi {
    base bgp-safi;
    description "Route Target SAFI - IANA Registry Assigned Number: 132";
}

identity ipv4-flow-spec-safi {
    base bgp-safi;
    description "IPv4 Flow Specification SAFI - IANA Registry Assigned Number: 133";
}
identity vpnv4-flow-spec-safi {
    base bgp-safi;
    description
        "IPv4 VPN Flow Specification SAFI -
         IANA Registry Assigned Number: 134";
}

5.  IANA Considerations

RFC Ed.: In this section, replace all occurrences of 'XXXX' with the actual RFC number (and remove this note).

This document registers the following namespace URIs in the IETF XML registry [RFC3688]:

--------------------------------------------------------------------
Registrant Contact: The IESG.
XML: N/A, the requested URI is an XML namespace.
--------------------------------------------------------------------

--------------------------------------------------------------------
Registrant Contact: IANA
XML: N/A, the requested URI is an XML namespace.
--------------------------------------------------------------------

This document registers the following YANG modules in the YANG Module Names registry [RFC6020]:

--------------------------------------------------------------------
name:         ietf-routing-types
prefix:       rt-types
reference:    RFC XXXX
--------------------------------------------------------------------

--------------------------------------------------------------------
name:         iana-routing-types
prefix:       iana-rt-types
reference:    RFC XXXX
--------------------------------------------------------------------
5.1. IANA-Maintained iana-routing-types Module

This document defines the initial version of the IANA-maintained iana-routing-types YANG module.

The iana-routing-types YANG module is intended to reflect the "Address Family Numbers" registry [IANA-ADDRESS-FAMILY-REGISTRY] and "Subsequent Address Family Identifiers (SAFI) Parameters" registry [IANA-SAFI-REGISTRY].

IANA has added this notes to the "iana-routing-types YANG Module" registry:

Address Families and Subsequent Address Families must not be directly added to the iana-routing-types YANG module. They must instead be respectively added to the "Address Family Numbers" and "Subsequent Address Family Identifiers (SAFI) Parameters" registries.

When an Address Family or Subsequent Address Family is respectively added to the "Address Family Numbers" registry or the "Subsequent Address Family Identifiers (SAFI) Parameters" registry, a new "identity" statement must be added to the iana-routing-types YANG module. The name of the "identity" is the same as the corresponding address family or SAFI only it will be a valid YANG identifier in all lowercase and with hyphens separating individual words in compound identifiers. The following substatements to the "identity" statement should be defined:

"base": Contains the value "address-family" for address families or "bgp-safi" for subsequent address families.

"status": Include only if a registration has been deprecated (use the value "deprecated") or obsoleted (use the value "obsolete").

"description": Replicate the description from the registry, if any. Insert line breaks as needed so that the line does not exceed 72 characters.

"reference": Replicate the reference from the registry, if any, and add the title of the document.

Unassigned or reserved values are not present in these modules.

When the iana-routing-types YANG module is updated, a new "revision" statement must be added in front of the existing revision statements.
IANA has added this new note to the the "Address Family Numbers" and "Subsequent Address Family Identifiers (SAFI) Parameters" registries:

When this registry is modified, the YANG module iana-routing-types must be updated as defined in RFC XXXX.

6. Security Considerations

This document defines common data types using the YANG data modeling language. The definitions themselves have no security impact on the Internet, but the usage of these definitions in concrete YANG modules might have. The security considerations spelled out in the YANG specification [RFC7950] apply for this document as well.

7. Acknowledgements

The Routing Area Yang Architecture design team members included Acee Lindem, Anees Shaikh, Christian Hopps, Dean Bogdanovic, Ebben Aries, Lou Berger, Qin Wu, Rob Shakir, Xufeng Liu, and Yingzhen Qu.

Thanks to Martin Bjorkland, Tom Petch, Stewart Bryant, and Radek Krejci for comments on the model and document text. Thanks to Jeff Haas and Robert Raszuk for suggestions for additional common routing types.

8. References

8.1. Normative References


8.2. Informative References


[I-D.ietf-teas-yang-te]

[I-D.ietf-bess-l2vpn-yang]

[I-D.ietf-bess-l3vpn-yang]

[I-D.ietf-mpls-base-yang]


Authors’ Addresses

Xufeng Liu
Jabil
8281 Greensboro Drive, Suite 200
McLean  VA 22102
USA
EMail: Xufeng_Liu@jabil.com

Yingzhen Qu
Futurewei Technologies, Inc.
2330 Central Expressway
Santa Clara  CA 95050
USA
EMail: yingzhen.qu@huawei.com

Acee Lindem
Cisco Systems
301 Midenhall Way
Cary, NC  27513
USA
EMail: acee@cisco.com
Christian Hopps
Deutsche Telekom
EMail: chopps@chopps.org

Lou Berger
LabN Consulting, L.L.C.
EMail: lberger@labn.net