Network Working Group
Internet-Draft
Updates: 2863 (if approved)
Intended status: Standards Track
Expires: April 5, 2020

D. Thaler
Microsoft
D. Romascanu
Independent

October 03, 2019

Guidelines and Registration Procedures for Interface Types and Tunnel Types
draft-thaler-iftype-reg-05

Abstract

The registration and use of interface types ("ifType" values) predated the use of IANA Considerations sections and YANG modules, and so confusion has arisen about the interface type allocation process. Tunnel types were then added later, with the same requirements and allocation policy as interface types. This document updates RFC 2863, and provides updated guidelines for the definition of new interface types and tunnel types, for consideration by those who are defining, registering, or evaluating those definitions.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The IANA IfType-MIB was originally defined in [RFC1573] as a separate MIB module together with the Interfaces Group MIB (IF-MIB) module. The IF-MIB module has since been updated and is currently specified in [RFC2863], but this latest IF-MIB RFC no longer contains the IANA IfType-MIB. Instead, the IANA IfType-MIB is now maintained as a separate module. Similarly, [RFC7224] created an initial IANA Interface Type YANG Module, and the current version is maintained by IANA.

The current IANA IfType registry is at [ifType-registry], with the same values also appearing in [yang-if-type], and the IANAifType textual convention at [IANAifType-MIB].

Although the ifType registry was originally defined in a MIB module, the assignment and use of interface type values are not tied to MIB modules or any other management mechanism. Interface type values can be used as values of data model objects (MIB objects, YANG objects, etc.), as parts of a unique identifier of a data model for a given

interface type (e.g., in an OID), or simply as values exposed by local APIs or UI on a device.

The TUNNEL-MIB was then defined in [RFC2667] (now obsoleted by [RFC4087]) which created a tunnelType registry ([tunnelType-registry] and the IANAtunnelType textual convention at [IANAifType-MIB]) and defined the assignment policy for tunnelType values to always be identical to the policy for assigning ifType values.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Problems

This document addresses the following issues:

1. As noted in Section 1, the former guidance was written with wording specific to MIB modules, and accordingly some confusion has resulted when using YANG modules. This document clarifies that ifTypes and tunnelTypes are independent from the type of, or even existence of, a data model.

2. The use of, and requirements around, sub-layers and sub-types are not well understood even though good examples of both exist. This is discussed in Section 4.

3. Since the ifType and tunnelType registries were originally defined, and are still retrievable, in the format of MIB modules (in addition to other formats), confusion arose when adding YANG modules as another format, as to whether each format is a separate registry. This is discussed in Section 5.

4. The registries are retrievable in the format of MIB and YANG modules, but there was no process guidance written to check that those formats were syntactically correct as updates were made, which led to the registry having syntax errors that broke tools. Section 6.1 adds a validation step to the documented assignment procedure.

5. Transmission values [transmission-registry] have often been allocated as part of ifType allocation, but no guidance exists about whether a requester must ask for it or not, and the request form has no such required field. As a result, IANA has asked the

Designated Expert to decide for each allocation, but no relevant guidance for the Designated Expert has been documented. This is remedied in Section 6.2.

6. Various documents and registries said to submit requests via email, but a web form exists for submitting requests, which caused some confusion around which is to be used. This is discussed in Section 7.

4. Interface Sub-Layers and Sub-Types

When multiple sub-layers exist below the network layer, each sub-layer can be represented by its own row in the ifTable with its own ifType, with the ifStackTable being used to identify the upward and downward multiplexing relationships between rows. Section 3.1.1 of [RFC2863] provides more discussion, and Section 3.1.2 of that RFC provides guidance for defining interface sub-layers. More recent experience shows that these guidelines are phrased in a way that is now too restrictive, since at the time [RFC2863] was written, MIB modules were the dominant data model.

This document clarifies that such guidance also applies to YANG modules.

Some ifTypes may define sub-types. For example, the tunnel(131) ifType defines sub-types, where each tunnelType can have its own MIB and/or YANG module with protocol-specific information, but there is enough in common that some information is exposed in a generic IP Tunnel MIB corresponding to the tunnel(131) ifType.

For requests that involve multiple sub-layers below the network layer, requests MUST include (or reference) a discussion of the multiplexing relationships between sub-layers, ideally with a diagram. Various well-written examples exist of such definitions, including [RFC3637] section 3.4.1, [RFC4087] section 3.1.1, and [RFC5066] section 3.1.1.

Definers of sub-layers and sub-types should consider which model is more appropriate for their needs. A sub-layer is generally used whenever either a dynamic relationship exists (i.e., which instances layer over which other instances can change over time) or a multiplexing relationship exists with another sub-layer. A sub-type can be used when neither of these are true, but where one interface type is conceptually a subclass of another interface type, as far as a management data model is concerned.

In general, the intent of an interface type or sub-type is that its definition should be sufficient to identify an interoperable

protocol. In some cases, however, a protocol might be defined in a way that is not sufficient to provide interoperability with other compliant implementations of that protocol. In such a case, an ifType definition should discuss whether specific instantiations (or profiles) of behavior should use a sub-layer model (e.g., each vendor might layer the protocol over its own sub-layer that provides the missing details), or a sub-type model (i.e., each vendor might subclass the protocol without any layering relationship). If a sub-type model is more appropriate, then the data model for the protocol might include a sub-type identifier so that management software can discover objects specific to the subtype. In either case, such discussion is important to guide definers of a data model for the more specific information (i.e., a lower sub-layer or a subtype), as well as the Designated Expert that must review requests for any such ifTypes or sub-types.

4.1. Alternate Values

Another design decision is whether to reuse an existing ifType or tunnelType value, possibly using a sub-type or sub-layer model for refinements, or to use a different value for a new mechanism.

If there is already an entry that could easily satisfy the modeling and functional requirements for the requested entry, it should be reused so that applications and tools that use the existing value can be used without changes. If however, the modeling and functional requirements are significantly different enough such that having existing applications and tools use the existing value would be seen as a problem, a new value should be used.

For example, originally multiple ifType values were used for different flavors of Ethernet (ethernetCsmacd(6), iso88023Csmacd(7), fastEther(62), etc.), typically because they were registered by multiple vendors. [RFC3635] then deprecated all but ethernetCsmacd(6), since using different values was seen as problematic since all were functionally similar.

As another example, the Teredo tunnel protocol [RFC4380] encapsulates packets over UDP, and a udp(8) tunnelType value was defined in [RFC2667], with the description "The value UDP indicates that the payload packet is encapsulated within a normal UDP packet (e.g., RFC 1234)." However, the protocol model is quite different between [RFC1234] and Teredo. For example, [RFC1234] supports encapsulation of multicast/broadcast traffic whereas Teredo does not. As such, it would be more confusing to applications and tools to represent them using the same tunnel type, and so [RFC4087] defined a new value for Teredo.

In summary, definers of new interface or tunnel mechanisms should use a new ifType or tunnelType value rather than reusing an existing value when key aspects such as the header format or the link model (point-to-point, non-broadcast multi-access, broadcast capable multi-access, unidirectional broadcast, etc.) are significantly different from existing values, but reuse the same value when the differences can be expressed in terms of differing values of existing objects, other than ifType/tunnelType, in the standard YANG or MIB module.

5. Available Formats

Many registries are available in multiple formats. For example, XML, HTML, CSV, and Plain text are common formats in which many registries are available. This document clarifies that the [IANAifType-MIB], [yang-if-type], and [yang-tunnel-type] MIB and YANG modules are merely additional formats in which the ifType and tunnelType registries are available. The MIB and YANG modules are not separate registries, and the same values are always present in all formats of the same registry.

The confusion stemmed in part from the fact that the IANA "Protocol Registries" [protocol-registries] listed the YANG and MIB module formats separately, as if they were separate registries. However, the entries for the yang-if-type and iana-tunnel-type YANG modules said "See ifType definitions registry." and "See tunnelType registry (mib-2.interface.ifTable.ifEntry.ifType.tunnelType)." respectively, although the entry for the IANAifType-MIB had no such note.

This document clarifies the relationship for the ifType and tunnelType registries with the following actions:

1. Add the following note to the entry for the IANAifType-MIB at [protocol-registries]: "This is one of the available formats of the ifType and tunnelType registries."

2. Change the note on the entry for the iana-if-type YANG module at [protocol-registries] to read: "This is one of the available formats of the ifType registry."

3. Change the note on the entry for the iana-tunnel-type YANG module at [protocol-registries] to read: "This is one of the available formats of the tunnelType registry."

4. Create a section for "Interface Parameters" at [protocol-registries], with entries for "Interface Types (ifType)" [ifType-registry], "Tunnel Types (tunnelType)" [tunnelType-registry], and "Transmission Values" [transmission-registry].

5. Update the ifType definitions registry [ifType-registry] to list MIB [IANAifType-MIB] and YANG [yang-if-type] as Available Formats.

6. Update the tunnelType definitions registry [tunnelType-registry] to list MIB [IANAifType-MIB] and YANG [yang-tunnel-type] as Available Formats, and change the title to "tunnelType Definitions" for consistency with the [ifType-registry] title.

7. Replace the [yang-if-type] page with the YANG module content, rather than having a page that itself claims to have multiple Available Formats.

8. Replace the [yang-tunnel-type] page with the YANG module content, rather than having a page that itself claims to have multiple Available Formats.

6. Registration

The IANA policy (using terms defined in [RFC8126]) for registration is Expert Review, for both Interface Types and Tunnel Types. The role of the Designated Expert in the procedure is to raise possible concerns about wider implications of proposals for use and deployment of interface types. While it is recommended that the responsible Area Director and the IESG take into consideration the Designated Expert opinions, nothing in the procedure empowers the Designated Expert to override properly arrived-at IETF or working group consensus.

6.1. Procedures

Someone wishing to register a new ifType or tunnelType value MUST:

1. Check the IANA registry to see whether there is already an entry that could easily satisfy the modeling and functional requirements for the requested entry. If there is already such an entry, use it or update the existing specification. Text in an Internet-Draft, or part of some other permanently available, stable specification may be written to clarify the usage of an existing entry or entries for the desired purpose.

2. Check the IANA registry to see whether there is already some other entry with the desired name. If there is already an unrelated entry under the name, choose a different name.

3. Prepare a registration request using the template specified in Section 6.3. The registration request can be contained in an Internet-Draft, submitted alone, or as part of some other

permanently available, stable, specification. The registration request can also be submitted in some other form (as part of another document or as a stand-alone document), but the registration request will be treated as an "IETF Contribution" under the guidelines of [RFC5378].

4. Submit the registration request (or pointer to document containing it) to IANA at iana@iana.org or (if requesting an ifType) via the web form at https://www.iana.org/form/iftype .

Upon receipt of a registration request, the following steps MUST be followed:

1. IANA checks the submission for completeness; if required information is missing or any citations are not correct, IANA will reject the registration request. A registrant can resubmit a corrected request if desired.

2. IANA requests Expert Review of the registration request against the corresponding guidelines from this document.

3. The Designated Expert will evaluate the request against the criteria.

4. Once the Designated Expert approves registration, IANA updates [ifType-registry], [IANAifType-MIB], and [yang-if-type] to show the registration for an Interface Type, or [tunnelType-registry], [IANAifType-MIB], and [yang-tunnel-type] to show the registration for a Tunnel Type. When adding values, IANA should verify that the updated MIB module and YANG module formats are syntactically correct before publishing the update. There are various existing tools or web sites that can be used to do this verification.

5. If instead the Designated Expert does not approve registration (e.g., for any of the reasons in [RFC8126] section 3), a registrant can resubmit a corrected request if desired, or the IESG can override the Designated Expert and approve it per the process in Section 5.3 of [RFC8126].

6.2. Media-specific OID-subtree assignments

The current IANAifType-MIB notes:

> The relationship between the assignment of ifType values and of OIDs to particular media-specific MIBs is solely the purview of IANA and is subject to change without notice. Quite often, a media-specific MIB's OID-subtree assignment within MIB-II's 'transmission' subtree will be the same as its ifType value.

However, in some circumstances this will not be the case, and implementors must not pre-assume any specific relationship between ifType values and transmission subtree OIDs.

The following change is to be made:

OLD: For every ifType registration, the corresponding transmission number value should be registered or marked "Reserved."

NEW: For future ifType assignments, an OID-subtree assignment MIB-II's 'transmission' subtree will be made with the same value.

Rationale: (1) This saves effort in the future since if a transmission number is later needed, no IANA request is needed that would then require another Expert Review. (2) The transmision numbering space is not scarce, so there seems little need to reserve the number for a different purpose than what the ifType is for. (3) The Designated Expert need not review whether a transmission number value should be registered when processing each ifType request, thus reducing the possibility of delaying assignment of ifType values. (4) There is no case on record where allocating the same value could have caused any problem.

6.3. Registration Template

6.3.1. ifType

The following template describes the fields that MUST be supplied in a registration request suitable for adding to the ifType registry:

Label for IANA ifType requested: As explained in Section 7.1.1 of [RFC2578], a label for a named-number enumeration must consist of one or more letters or digits, up to a maximum of 64 characters, and the initial character must be a lower-case letter. (However, labels longer than 32 characters are not recommended.) Note that hyphens are not allowed.

Name of IANA ifType requested: A short description (e.g., a protocol name), suitable to appear in a comment in the registry.

Description of the proposed use of the IANA ifType: Requesters MUST include answers, either directly or via a link to some document with the answers, to the following questions in the explanation of the proposed use of the IANA IfType:

- How would IP run over your ifType?

- Would there be another interface sublayer between your ifType and IP?

- Would your ifType be vendor-specific or proprietary? (If so, the label MUST start with a string that shows that. For example, if your company's name or acronym is xxx, then the ifType label would be something like xxxSomeIfTypeLabel.)

- Would your ifType require or allow vendor-specific extensions? If so, would the vendor use their own ifType in a sub-layer below the requested ifType, or a sub-type of the ifType, or some other mechanism?

Reference, Internet-Draft, or Specification: A link to some document is required.

Additional information or comments: Optionally any additional comments for IANA or the Designated Expert.

6.3.2. tunnelType

Prior to this document, no form existed for tunnelType (and new tunnelType requests did not need to use the ifType form that did exist). This document therefore specifies a tunnelType form.

The following template describes the fields that MUST be supplied in a registration request suitable for adding to the tunnelType registry:

Label for IANA tunnelType requested: As explained in Section 7.1.1 of [RFC2578], a label for a named-number enumeration must consist of one or more letters or digits, up to a maximum of 64 characters, and the initial character must be a lower-case letter. (However, labels longer than 32 characters are not recommended.) Note that hyphens are not allowed.

Name of IANA tunnelType requested: A short description (e.g., a protocol name), suitable to appear in a comment in the registry.

Description of the proposed use of the IANA tunnelType: Requesters MUST include answers, either directly or via a link to some document with the answers, to the following questions in the explanation of the proposed use of the IANA tunnelType:

- How would IP run over your tunnelType?

- Would there be another interface sublayer between your tunnelType and IP?

- Would your tunnelType be vendor-specific or proprietary? (If so, the label MUST start with a string that shows that. For example, if your company's name or acronym is xxx, then the tunnelType label would be something like xxxSomeIfTypeLabel.)

- Would your tunnelType require or allow vendor-specific extensions? If so, would the vendor use their own tunnelType in a sub-layer below the requested tunnelType, or some sort of sub-type of the tunnelType, or some other mechanism?

Reference, Internet-Draft, or Specification: A link to some document is required.

Additional information or comments: Optionally any additional comments for IANA or the Designated Expert.

7. Submitting Requests

At the time of this writing, [IANAifType-MIB] said: "Requests for new values should be made to IANA via email (iana&iana.org)." However, a web form exists (https://www.iana.org/form/iftype), which is an apparent contradiction, but submissions either way are accepted.

IANA is requested to update the MIB module to instead say: "Interface types must not be directly added to the IANAifType-MIB MIB module. They must instead be added to the "ifType definitions" registry at [ifType-registry]."

(Note that [yang-if-type] was previously updated with this language.)

8. IANA Considerations

This entire document is about IANA considerations.

9. Security Considerations

Since this document does not introduce any technology or protocol, there are no security issues to be considered for this document itself.

10. References

10.1. Normative References

[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>.

[RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, DOI 10.17487/RFC2578, April 1999, <https://www.rfc-editor.org/info/rfc2578>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC5378] Bradner, S., Ed. and J. Contreras, Ed., "Rights Contributors Provide to the IETF Trust", BCP 78, RFC 5378, DOI 10.17487/RFC5378, November 2008, <https://www.rfc-editor.org/info/rfc5378>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

10.2. Informative References

[IANAifType-MIB] IANA, "IANAifType-MIB", February 2019, <http://www.iana.org/assignments/ianaiftype-mib>.

[ifType-registry] IANA, "ifType definitions", June 2019, <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-5>.

[protocol-registries] IANA, "Protocol Registries", June 2019, <https://www.iana.org/protocols>.

[RFC1234] Provan, D., "Tunneling IPX traffic through IP networks", RFC 1234, DOI 10.17487/RFC1234, June 1991, <https://www.rfc-editor.org/info/rfc1234>.

[RFC1573] McCloghrie, K. and F. Kastenholz, "Evolution of the Interfaces Group of MIB-II", RFC 1573, DOI 10.17487/RFC1573, January 1994, <https://www.rfc-editor.org/info/rfc1573>.

[RFC2667] Thaler, D., "IP Tunnel MIB", RFC 2667, DOI 10.17487/RFC2667, August 1999, <https://www.rfc-editor.org/info/rfc2667>.

[RFC3635] Flick, J., "Definitions of Managed Objects for the Ethernet-like Interface Types", RFC 3635, DOI 10.17487/RFC3635, September 2003, <https://www.rfc-editor.org/info/rfc3635>.

[RFC3637] Heard, C., Ed., "Definitions of Managed Objects for the Ethernet WAN Interface Sublayer", RFC 3637, DOI 10.17487/RFC3637, September 2003, <https://www.rfc-editor.org/info/rfc3637>.

[RFC4087] Thaler, D., "IP Tunnel MIB", RFC 4087, DOI 10.17487/RFC4087, June 2005, <https://www.rfc-editor.org/info/rfc4087>.

[RFC4380] Huitema, C., "Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs)", RFC 4380, DOI 10.17487/RFC4380, February 2006, <https://www.rfc-editor.org/info/rfc4380>.

[RFC5066] Beili, E., "Ethernet in the First Mile Copper (EFMCu) Interfaces MIB", RFC 5066, DOI 10.17487/RFC5066, November 2007, <https://www.rfc-editor.org/info/rfc5066>.

[RFC7224] Bjorklund, M., "IANA Interface Type YANG Module", RFC 7224, DOI 10.17487/RFC7224, May 2014, <https://www.rfc-editor.org/info/rfc7224>.

[transmission-registry]
IANA, "transmission definitions", June 2019, <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-7>.

[tunnelType-registry]
IANA, "Internet-standard MIB - mib-2.interface.ifTable.ifEntry.ifType.tunnelType", June 2019, <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml#smi-numbers-6>.

[yang-if-type]
IANA, "iana-if-type YANG Module", February 2019, <http://www.iana.org/assignments/iana-if-type>.

[yang-tunnel-type]
IANA, "iana-tunnel-type YANG Module", June 2019,
<https://www.iana.org/assignments/iana-tunnel-type>.

Authors' Addresses

Dave Thaler
Microsoft

EMail: dthaler@microsoft.com

Dan Romascanu
Independent

EMail: dromasca@gmail.com