Guidelines and Registration Procedures for Interface Types and Tunnel Types
draft-thaler-iftype-reg-04

Dave Thaler
Dan Romascanu
Document status

• Suresh Krishnan is AD Sponsoring this doc
• -01 presented at IETF 104 INTAREAWG
• Four issues raised since then, tracked in github:
  • https://github.com/dthaler/iftype-reg/issues
• All have been addressed in doc updates (see next slides)
  • #1: UDP-based tunnels
  • #2: tunnelType registry reference
  • #3: Confusion around registries vs registry formats
  • #4: Registration Template for tunnel types
#3: Confusion around registries vs registry formats

**Problem:**
- Belief by some that draft-ietf-softwire-iftunnel (now in RFC Ed Queue) was creating a new registry

**Resolution:**
- draft-thaler-iftype-reg-03 clarified:
  - MIB module & YANG module are simply *alternate formats* in which these registries can be retrieved, just like HTML, XML, CSV already are
- Added Section 5 (“Available Formats”) with this discussion
- Some confusion stems from current presentation/labels on IANA site
  - ifType & tunnelType registries did not list MIB/YANG as formats, but looked more like links to other registries
  - YANG module “registry” pages were close already (e.g., “See ifType definitions registry.”), but MIB module “registry” page had no such statement
- Draft proposes changes to present them as Available Formats, not “registries”
Structure of Management Inform

Last Updated
2019-07-16

Note
For the management of hosts and gateways on the Internet, a structure for the information has been defined. This should be used with any of several possible protocols.

The data structure is the "Structure and Identifiers of Management Information Protocol" (SNMP) ["Common Management Information Protocol over TCP" (CMIP)].

ifType Definitions

Registration Procedure(s)
- Expert Review

Expert(s)
- Dave Thaler (primary), Dan Romascanu (secondary)

Description
iso.org.dod.internet.mgmt.mib-2.interface.ifTable.ifEntry.ifType (1.3.6.1.2.1.2.1.3)

Reference
[RFC1213][RFC2863][RFC7224]

Note
For every ifType registration, the corresponding transmission number value should be registered or marked "Reserved." In addition, the [IANAIfType-MIB] and [iana-if-type-YANG] modules should be updated in accordance with [RFC2863] and [RFC7224], respectively.

Note
For a functional mib language definition please see the following:
IANA registry ianaiftype-mib

Rules for real mib names:

```
#NAME?
"if its made of several words,"
the second and later word's first letter is uppercase
#NAME?
#NAME?
#NAME?
```

Thus by way of example we have:

```
traif    kosher
-------   ---------
ddn-x25   "ddnX25(4),"
FDDI      "Fddi(15),"
smds-dxi  "smdsDxi(43),"
IEEE802.11 "IEEE802.11(71),"
```

"Finally, the last item in the list has no comma, while all previous items have a comma.
#2: tunnelType registry reference

**Problem:**
- As discussed last IETF, the tunnelType registry is intentionally defined to always use the same assignment policy as ifType, and it has same Designated Experts
  - Part of ifType/tunnelType Expert Review includes verifying the right one of the two is being assigned
- Tunnel types were mentioned, but large portions of -01 only covered ifType

**Resolution:**
- Title changed to add “and Tunnel Types”
- Content now covers both ifType and tunnelType equally
### #4: Registration Template for tunnel types

**Problem:**
- ifType had a registration template in the draft (and in previous RFCs), and has an optional IANA form that matches it
  - “This template describes the fields that MUST be supplied in a registration request suitable for adding to the ifType registry:”
- tunnelType had neither, and so hard to apply the same “MUST” standard
- Conflicts with the RFC requirement to use “same assignment policy”

**Resolution:**
- Added a registration template in the draft, that has parity with ifType one
- (no statement about whether IANA should have a form, this is up to IANA)
Examples from existing RFCs

Plenty of non-IETF examples exist too

IETF 105 - INTAREA WG
#1: UDP-based tunnels (1/2)

• Mohamed Boucadair:
  “Add some text to encourage UDP-based tunnel protocol designers to register their own code instead of reusing the one currently assigned to generic UDP encap (8).”

• -01 had section on “Interface Sub-Layers and Sub-Types” but nothing explicit on alternate values

• -04 adds section 4.1 “Alternate Values” and uses Ethernet (see previous slide) and the UDP tunnel issue as 2 very different examples
  • The (unfortunately-named) udp(8) was originally added for [RFC1234] encap, which supports things like multicast
  • In contrast, other UDP encap mechanisms like teredo got different values because the link model is quite different
#1: UDP-based tunnels (1/2)

• After discussion of the two examples, concludes with:
  • “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.”
Questions?