MAC Address Format Issue

Scott Mansfield
Ericsson

MAC Address Format

- IETF and IEEE have different patterns for mac-address
  - IETF Format: pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
    - uses ':' as separator
  - IEEE Format: pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}";
    - uses '-' as separator
    - Also ':' has a defined meaning in IEEE specs (bit-reversal of each hex digit)
    - However the bit-reversal issue is historic (but there really should be an amendment to official recognize that fact)

Not just a ‘-’ or ‘:’ problem

- IEEE definition
- Pattern allows upper and lower case characters but description says uppercase is used.

```
typedef mac-address {
  type string {
    pattern "[0-9a-fA-F]{2}(-[0-9a-fA-F]{2}){5}";
  }
  description
    "The mac-address type represents a MAC address in the canonical
    format and hexadecimal format specified by IEEE Std 802. The
    hexidecimal representation uses uppercase characters.";
  reference
    "3.1 of IEEE Std 802-2014
    8.1 of IEEE Std 802-2014";
}
```

- IETF definition
- Pattern allows upper and lower case but makes no indication on which is used.

```
typedef mac-address {
  type string {
    pattern '[0-9a-fA-F]{2}(:[0-9a-fA-F]{2}){5}';
  }
  description
   "The mac-address type represents an IEEE 802 MAC address.
    The canonical representation uses lowercase characters.

    In the value set and its semantics, this type is equivalent
    to the MacAddress textual convention of the SMIv2.";
  reference
   "IEEE 802: IEEE Standard for Local and Metropolitan Area
              Networks: Overview and Architecture
    RFC 2579: Textual Conventions for SMIv2";
}
```

Issue with strings

- mac-address typedef is a string in YANG
- That means when mac-address is used as a key, the format used must match not only the separator (‘:’ or ‘-’) but the case of the character representing the hexadecimal number

Why SNMP is different

- In SNMP a MacAddress was an OCTET STRING of size 6 with a display hint.
- On the wire the MacAddress is treated as a string of octets that are not affected by the display hint or the separator used.
- So AE-12-FF would be the same as ae:12:ff

Co-existence of IETF and IEEE Definitions

- Greping the YANG repository there are places in IEEE where ietf-yang-types is imported.
  - However there are no places where yang:mac-address is used in IEEE
- So there doesn't seem to be pressing issue --- YET

What to do

- Common wisdom says it is too late to change either the IEEE or IETF definition to use a 6 byte binary array
  - This would fix the “on-the-wire” and key comparison issue
- Identify potential conflicts
  - Modules that use both yang:mac-address and ieee:mac-address and try to compare them
  - Even if only one definition is used, some hints or guidelines should be created because the format of the string (upper/lower case) matters for comparison
- IEEE should start a project to fix the definition of mac-address in ieee802-types.yang
  - Make align the format with the description
  - or as Don suggests
    - “The EtherType value represented in the canonical order defined by IEEE 802. This value can contain uppercase or lowercase alpha hex characters .”
  - Coordinate with IETF and OpenConfig to understand options when comparing IEEE formatted strings and IETF formatted strings