Interface configuration

draft-bjorklund-netmod-interfaces-cfg-00

IETF 80
Martin Björklund
mbj@tail-f.com

Design Objectives

- Existing implementations will have to map to this model – keep it simple.
- The data model should be suitable to use as-is for new implementations.
- The data model must be extensible for different interface types.
- The mapping to `ifTable` must be well-defined.
- Must support interface layering
  - 1 over N
  - N over 1
- Focus on configuration – do not duplicate all stats objects from IF-MIB.

Data Model

- flat list keyed by name
- `name` is an arbitrary string, but an implementation MAY restrict it if it doesn't support arb. names; e.g. “ethN”
- `type` (*identityref*) – mandatory, but implementations MAY fill it in based on the name (if name is “ethN”, set type to 'ethernet')
- `location` for physical interfaces. format is vendor-specific. MAY be derived from the name (eth-1/2)
- `admin-status` to enable / disable an interface
- layering pushed to extension modules
  - pro: use natural objects for each technology
  - con: no way to generically figure out layering

Open Issues 1(2)

- Request from the ML:
  - Add a `oper-status` config false leaf, of type identityref
  - Why is current `ifOperStatus` not enough?
- Which objects from IF-MIB should we include? Request on the ML:
  - `ifLinkUpDownTrapEnable` – *should probably include it*
  - `ifPromiscuousMode` – *is this config or an action?*
  - `ifAlias` – *is it needed?*
  - `ifStackTable` – *not config, so suggest it is not included*
  - notifications `linkUp` and `linkDown` – *hand-written notifs will be "better" than the auto-generated*
- Should we handle *testing* `admin-status` (like `ifAdminStatus`)?

Open Issues 2(2)

- Is it a good idea to use an identity for the `type`, or should we re-use the `IANAifType`?
- Which generic config objects should we include? Currently, only `mtu` is included. Should `mtu` even be there?

One option is to define some groupings with common config objects, like `mtu`, `speed`, `duplex`(?). Media-specific modules can the use these groupings when they augment the interface list.