

Indexing in the CFM MEP/MIP Tables

Kevin Nolish, Ericsson

23 January 2008

The Problem

- There are places in the CFM MIBS where tables are indexed by a mechanism that specifies the location of a CFM entity or the traffic flows to which a CFM entity may respond.
- This is well and good, but as we come up with new places where we want to insert maintenance entities, e.g. 802.1Qay wanting to place these on an I-SID, we need to re-index the tables.
- Re-indexing tables is a nasty thing to do. It requires deprecating existing tables and then defining new tables with a new indexing scheme.
- The goal is to have a scheme that allows us to defined new ways of indicating the traffic to which a maintenance entity applies without invalidating existing MIBS.

Observations on MIB Indexing

- MIB tables can be thought of as a mapping from an index, identified by some sort of tuple, to a set of columns.
- It is easy to add new columns to such a mapping by either the `AUGMENTS` clause or via the sparse augmentation mechanism.
- It is not easy to map from different indices to the same set of columns. Thus MIBS cannot easily model the mathematical notion of different domains mapping to the same co-domain.

An Example Where This Occurs

- The ifTable and related tables.
 - Different kinds of physical and virtual interfaces map to the information in the ifTable (and friends).
 - DS1/E1 Ports, Frame Relay Circuits, Pseudo Wires, Ethernet Ports, etc...can all be used as Interfaces.
- Note that the basic scheme has been in place and has been extended with new types of physical interface without invalidating the structure or requiring modifications to the underlying Interface MIB tables.

How this applies to CFM

- The basic notion of CFM is that Maintenance Entities, MEs, (either MEPs or MIPs) are inserted into the network at interesting “places”.
- “places” is in quotes because a “place” in a network for the purpose of CFM represents a combination of a physical location, virtual location, and some sort of flow specifier.
- The problem is that different types of “places” are named differently. Consider a CFM located at a physical bridge port vs. a CFM entity located on a particular I-SID in a BEB.
- Furthermore, going forward, we are probably going to come up with new and interesting “places” to insert MEs.
- Re-indexing the CFM tables each time we cook up a new type of “place” is not an ideal solution.
- Note, however, that regardless of the “place” of a ME, the configuration information associated with the ME doesn’t change. In essence, the co-domain is fixed, but we wish to vary the domain.

Outline of the Solution

- Change the Dot1agCfmStackTable to be indexed by MepId.
- Add auxiliary tables that map “places” to MepId. For example, the current state of the CFM MIB would have a single table that takes
 - [ifIndex X VlanIdOr0 X Level X Direction] -> MepId
- One can add other tables to support I-SID mappings, etc...

Indexing Issues

- The proposed scheme makes it easy to go from a “place” to a ME, but it is difficult to follow the inverse mapping.
- One could add a type field, similar to IfType, that allows one to specify the sort of “place” to which a ME is bound.
 - This would make it somewhat easier to perform the “backwards” mapping, although a linear scan of the appropriate “place” table is still needed.

Limitations of the Approach

- It is difficult to go from a ME back to the definition of the traffic that the ME operates upon.
- The scheme is extensible, but it does have some aspects of a “using a sledgehammer to crack a walnut” solution.
 - If we are really, really sure that we don’t need new ways to specify the traffic upon which a ME operates, then we should probably try to solve the problem by using a single table with a complicated, but fixed, indexing scheme.