July 24, 2018

# ANALYZING AND TRANSFORMING TIME DIVISION MULTIPLEXING EQUIPMENT STRUCTURE

Randy Zhang

Weiping Wang

Mustafa Cevik

Walid Wakim

# ANALYZING AND TRANSFORMING TIME DIVISION MULTIPLEXING EQUIPMENT STRUCTURE

AUTHORS:
Randy Zhang
Weiping Wang
Mustafa Cevik
Walid Wakim

## ABSTRACT

The native Time Division Multiplexing (TDM) equipment hierarchy is not consistent and includes various non-hardware elements and many other obstacles. Techniques described herein provide a methodology for transforming the original structure to a modernized hierarchy that can correctly identify hardware elements and makes analysis and automation much more straightforward.

## DETAILED DESCRIPTION

Time Division Multiplexing (TDM) equipment does not follow the common hierarchy to describe relationships as in Internet Protocol (IP) and other equipment. Furthermore, it may vary depending on the type of TDM equipment, such as Digital Crossconnect Systems (DCSs) and Add-Drop Multiplexers (ADMs). This poses serious issues for data analysis and automation. Accordingly, described is a transformation algorithm to convert the native hierarchy into a modernized format that is consistent regardless of the original structure and equipment type. The algorithm may analyze and transform the native TDM equipment structure into a modernized form for more efficient analysis and automation.

Before the transformation, the original tree is constructed using the native equipment data. Each piece of equipment is identified by a Hardware ID (HID), parent HID, Relay Rack ID (RID), unit ID, subdivision, and assigned circuit ID. As illustrated in Figure 1 below, the methodology described herein includes two steps to transform the equipment hierarchy. First, the new structure is built by rebranching shelves and resetting the native structure. Second, any unneeded components are pruned from the rebranched structure, and role and use tagging is performed for each element.

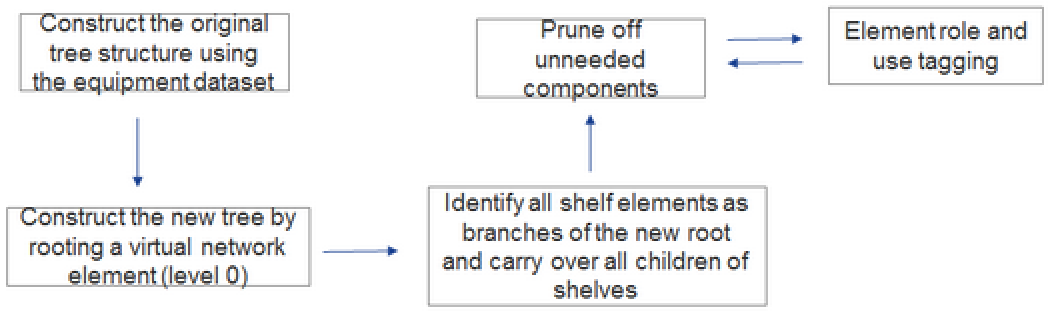


*Figure 1: Equipment hierarchy transformation workflow*

Figure 2 below illustrates a sample tree hierarchy from the original TDM database transformed after rebranching. Virtual Tributaries (VT) are lower order containers. The transformation starts by creating the root of the new tree by adding a level 0 Virtual Network Element (NE) using the master HID and RID to represent the entire NE. The original tree may be walked top down with each element. If the RID is different from its parent, its level is reset to 1 (shelf level). All level 1 elements and its children elements are rebranched to the root of the new tree. After rebranching, all of the shelves are moved to level 1 as children of the level 0 root, which is the virtual NE. All line card shelves with their children are carried over as branches and leaves while administrative shelves are childless.

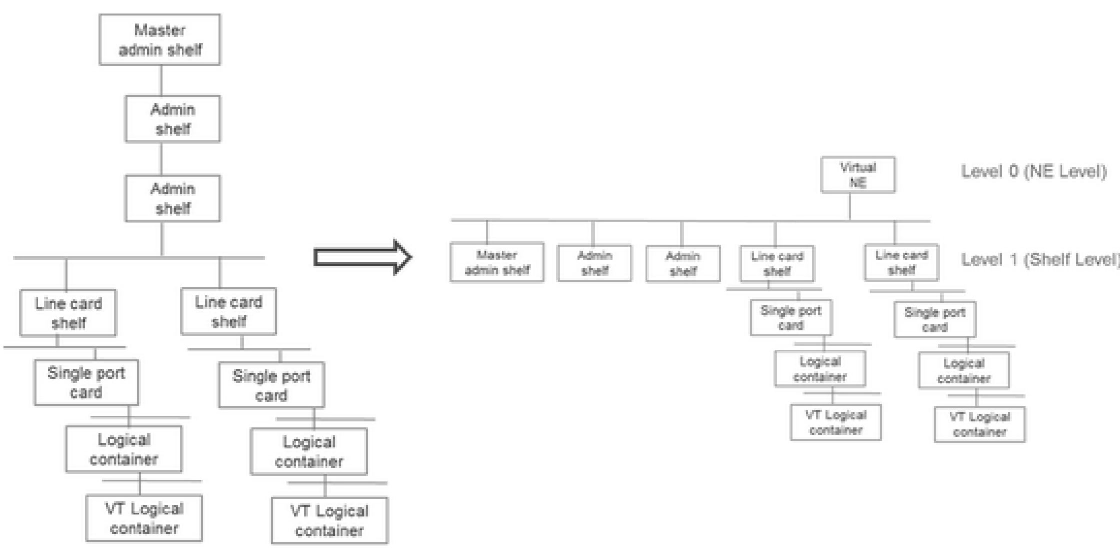


*Figure 2: Sample DCS native equipment tree (left) vs rebranched tree (right)*

With respect to pruning and tagging, first all VT HIDs from the new tree are pruned. Second, each element on the new tree is tagged with an element role of shelf, port, or card. All level 1 elements are tagged as shelf, all untagged leaves are tagged as port, and the rest are tagged as card. Figure 3 below illustrates the outcome of the new DCS tree. A single port card is treated as a port as the card functionality is combined with the port.

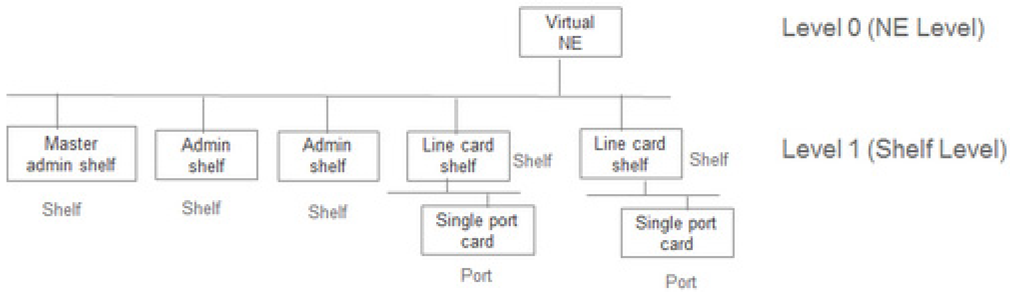


*Figure 3: Example of DCS equipment tree after final transformation*

Third, other logical containers found in ADM equipment tree are pruned. High-speed cards in ADMs bring in logical containers that will need to be pruned. In ADMs, a high-speed logical container is identified as a port and its unit ID follows some specific format(s). Figure 4 below illustrates an example of the ADM tree transformation from the original tree to the new tree. The high-speed cards are single port cards. HID and unit IDs are used to denote the formatting with actual values being fictitious.

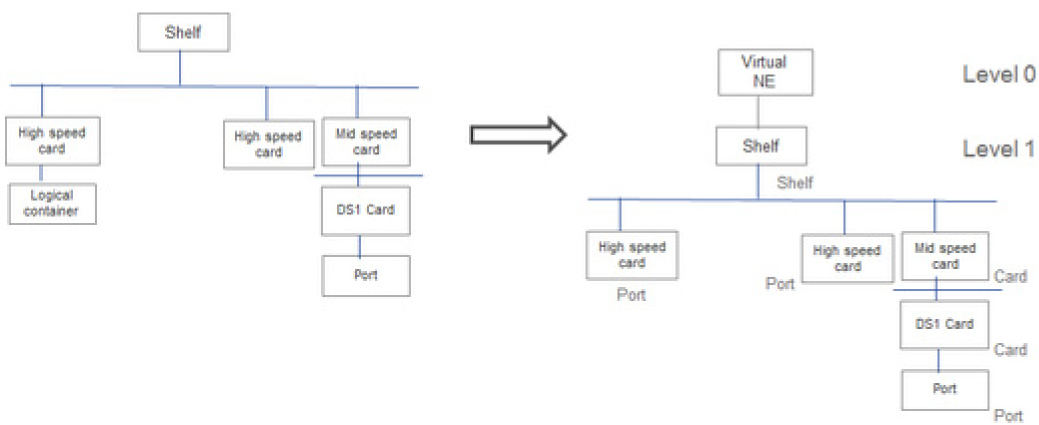


*Figure 4: Example of ADM equipment tree, native (left) vs after-transformation (right)*

Fourth, all redundant Virtual Concatenation (VCAT) members are pruned. VCAT is used to aggregate multiple lower rate member circuits into a higher speed circuit. Because all VCAT members share the same hardware, only one element should remain in the tree. Fifth, the tagging step is run again to retag all the elements after the pruning. This is the fully transformed tree.

In summary, techniques described herein provide a methodology for transforming the original structure to a modernized hierarchy that can correctly identify hardware elements and makes analysis and automation much more straightforward.