A Framework For Upgrading Whitebox Firmware in a Distributed Disaggregated Chassis Implementation from a Service Provider Perspective
A Framework For Upgrading Whitebox Firmware in a Distributed Disaggregated Chassis Implementation from a Service Provider Perspective

Prasanna Ramachandran, Director, AT&T
Shivakumar Mysore, LMTS-1, AT&T
Tuan Duong, LMTS, AT&T
11 electronic components that need firmware code to operate. ➔ [11 firmware images + 11 Checksum files]

UTILITY Programs are needed to load the firmware image into the flash memory of each device ➔ 11 Utility programs.

This is one WB from one ODM..
Disaggregation Paradigm: Mix & Match
Problem Description: Firmware Integration & Release Coordination

**FW Support Integration Complexity Challenges**

- NOS Burden
  - Integrate FW Utils for each ODM WB
  - May require adjustment to Logic due to dependencies as release changes

**ODM Release Coordination Complexity**
- ODM may need to maintain different release trains depending upon the NOS Release/Development Cycle
Business and Operational Challenges in Disaggregation Paradigm

**UTILITIES to upgrade FW components:**
- Some available in source code. Some available in binary form for mainline operating systems. Utilities are unique to components and vary by ODM.

**Upgrade from ONIE not Acceptable:**
- Service Provider Operational teams accustomed to operating OEM based networking equipment
- Operational tools and methodologies have been optimized to work in this manner.
- New method should be as transparent as old CLI method of upgrade in the OEM systems.
- Minimum number of operator initiated reboots with minimum downtime.

**NOS Partner WB Business & Technical Challenge:**
- One NOS partner business model is purely white boxes and NOS is built on a mainline open-sourced linux branch
- Second NOS partner has a large embedded OEM business and is expanding to the white boxes/disaggregation opportunities with an established NOS built on a commercial/private linux branch.
- It is crucial that the whitebox/disaggregation model does not present an ever-growing development and support problem as the NOS supports more and more white boxes from different ODMs.
Traditional Method to upgrading FW from NOS: Challenges

- The issue is for some components, the source code of the utilities is available while for others, the utilities are available only in binary format for limited number of mainline OS/Linux distributions. The binary format is a challenge for NOS that are not built on mainline OS distributions to integrate and support those utilities.

- In a disaggregated model, the problem can grow in complexity for the NOS if those utilities need to be integrated into the NOS. As the number of supported white boxes grows, the number of utilities and the versioning of each utility supported can become unmanageable.

- The reverse problem is also present for the ODM as the same white box can work with a number of NOS providers where each NOS providers require different versioning support.

- Potentially, there can be interdependencies of the versions of the components on white box which dictates a certain order/sequence that the firmware needs to be upgraded. This sequence can change depending on the issue/bug at hand. The ODM of the white box would be the best source to know these peculiar details. However, because the firmware upgrade has been integrated into the NOS, it is now the responsibility of the NOS to sequence these upgrades correctly or provide complicated instructions to operator to sequence these upgrades.

- Finally, for the DDC version1 implementation, this issue is multiplied by 3 because the implementation involves hardware from three different manufacturers: HPE, Accton/EdgeCore, UFISpace.
**ONIE Multi-Firmware Update Relay Boot approach**

1a. **Operator Initiation:**
While NOS running on WB, Operator SSH into WB Copy (SCP) the ONIE Multi-FW Updater package into WB and initiates the update command from NOS.

1b. **NOS Staging:**
NOS would copy the package down to the appropriate directory in the ONIE partition, sets up necessary MetaData files, log files, and Next-Boot flags and reboot the WB.

2. **Multi-FW Updater runs under ONIE:**
After 1b, WB boots into ONIE and automatically runs the Multi-FW Updater developed and optimized by ODM. ODM has the best knowledge on how to upgrade all the FW components. As many reboots as required to complete the JOB. When completed, the Multi-FW Updater sets the Boot Flag back to NOS partition and reboots.

3. **NOS resume operation.**
After boots up and check flag to know that it has just booted from the ONIE Multi-FW Updater. Runs appropriate tasks to gather information from the FW Updater Log.

4. **Operator Regains Access.**
Operator regains access to WB and check on Upgrade status.
Requirements on ONIE Multi-Firmware-Updater

Firmware Licensing & Meta-Data File
ODM will handle all the licensing necessary to upgrade all the firmware on the white box when the Multi-Firmware Updater runs. A contribution for the definition of the Meta-Data is also being submitted to OCP “Firmware Upgrade for DDC”. When ODM delivers the TAR ball upgrade package for the relevant white box, the Meta-Data file should be included so that a controller software can manage the automation and upgrade process for white boxes from different vendors.

Prerequisites Checking and Sequencing Logic
ODM has complete control over implementation, sequencing and prerequisites checking to deliver the best and most reliable FW upgrade performance for each release/version.

FORCED Upgrade Switch
Multi-FW Updater needs to support FORCED switch to overwrite FW on component regardless of version checking. This is primarily for PPI to “PREP” WB received from factory for specific use case in Service Provider’s network.

Local Log File
Multi-FW Updater needs keep a local log file with sufficient details on the ONIE partition that is accessible to the NOS after NOS regains control.

CI/KGI for BIOS and BMC flash1 and Flash2
Multi-FW Updater needs install same FW version on both flash of BIOS and BMC for those WB that supports such HW configuration.
ONIE Multi-Firmware Updater in DDC

1. Operator Initiation:
   Operator SSH to “NOS-Brain” of DDC, downloads FW package, initiates FW upgrade to NCP0 and NCP2

6. NOS-Brain Status FW Upgrade:
   “NOS-Brain” reports status of FW upgrade on NCP0 and NCP2 to operator

NOTE: unlike Single Pizza Box System, Operator does not lose contact with “NOS-Brain” during NCPx or NCM or NCF upgrades

0. TAR Ball Firmware Package(s) for white boxes from different ODMs. TAR Ball from each ODM includes Meta-Data File downloaded to DCC of the DDC. DCC coordinates the upgrades of the FW on WBs in the DDC

The meta-data file specification standardize the interaction/communication between the hardware module with the “NOS-Brain” residing on the NCC
Call to Action

• This white paper is contributed under Telco-Edge Project.
• ODMs: Accton/Edgecore, UfiSpace, Delta Network International
• NOS: DriveNets, Cisco
• The ideas presented here are being implemented as enhancements to DDC implementation that is already deployed and carrying traffic.
• Encourage other Providers and Carriers to investigate/adopt.

Where to buy: Contact listed ODM or NOS

Project Wiki with latest specification: https://www.opencompute.org/contributions
Type AT&T into Search box
Open Discussion