Edgecore ASFvOLT16 VOLTHA Adapter and Driver

Kim Kempf, Sr. Systems Architect
CORD Build 2017, San Jose
November 8, 2017
• **Disaggregated OLT** means minimizing the amount of software in the device
• VOLTHA provides the vOLT control abstraction
• VOLTHA device adapter interacts with OLT hardware
• SW required in the OLT to interface to the adapter
  • But ASFvOLT16 has no existing OLT control software
  • No existing open source OLT control software

In-OLT control is:
- NETCONF?
- REST?
- Thin API?
- Custom?
Edgecore ASFvOLT16 Whitebox OLT

- In-OLT software is needed for:
  - Embedded CPU operating system
  - Broadcom Maple PON MAC SoCs
  - Broadcom DNX Qumran Ethernet switch
  - Other hardware and functions
    - XFP, QSFP, Fans, Timing
    - Software update
    - Health and status
Edgecore ASFvOLT16 Whitebox OLT

- Software solutions:
  - ONL can provide foundation for NOS
  - ONLP provides most HW peripheral support
  - Broadcom Maple SDK for PON MAC
  - Broadcom Switch SDK for Qumran switch
  - Broadcom BAL – Broadband Abstraction Layer
    - Provides abstraction to manage Maple and Qumran as an OLT system
    - Intended as foundation for OLT control FW
Edgecore ASFvOLT16 Whitebox OLT

- BAL provides object oriented OLT access API
- Hides details of PON MAC and switch API calls

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th>Parameter Description</th>
<th>Data Type</th>
<th>Access</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin_state</td>
<td>Administrative state</td>
<td>bcmbal_state</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>oper_status</td>
<td>Operational status</td>
<td>bcmbal_status</td>
<td>R</td>
<td></td>
<td></td>
</tr>
<tr>
<td>access_int_id</td>
<td>The ID of the subscriber side interface; i.e. PON</td>
<td>bcmbal_inf_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>network_int_id</td>
<td>The ID of the network side interface; i.e. NNI</td>
<td>bcmbal_inf_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sub_term_id</td>
<td>The ID of the subscriber terminal device</td>
<td>bcmbal_sub_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>svc_port_id</td>
<td>The ID of the service port (for GPON/XG-PON - GEM ID)</td>
<td>bcmbal_service_port_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>agg_port_id</td>
<td>The ID of the aggregate port (for GPON/XG-PON - ALLOC ID)</td>
<td>bcmbal_aggregation_port_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>resolve_mac</td>
<td>A flag indicating if the MAC address table should be used in DS GEM resolution</td>
<td>bcmmacs_bool</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>base_tc_id</td>
<td>The base index of the TC object(s) to be used for this flow</td>
<td>uint16_t</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classifier</td>
<td>The classifier for this flow</td>
<td>bcmbal_classifier</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>action</td>
<td>The action associated with the flow</td>
<td>bcmbal_action</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sla</td>
<td>SLA parameters for this flow</td>
<td>bcmbal_sla</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>cookie</td>
<td>Application cookie</td>
<td>bcmbal_cookie</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>priority</td>
<td>Priority for this flow in case of multiple match.</td>
<td>uint16_t</td>
<td>RW</td>
<td>1</td>
<td>255</td>
</tr>
<tr>
<td>group_id</td>
<td>RW - The multicast group associated with this flow, valid for type MULTICAST only</td>
<td>bcmbal_group_id</td>
<td>RW</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Edgecore ASFvOLT16 Whitebox OLT

• The problem?
  • BAL/Maple and Qumran SDK source code is proprietary and requires source code license agreement (SLA) between equipment provider and Broadcom
  • No equivalent to the OF-DPA Community Development Package (CDP) for BAL exists from Broadcom
  • The VOLTHA community wants ASFvOLT16 to be as open source as possible
• The solution?
  • Work with Broadcom to open the BAL API (calls and object model)
  • Propose ASFvOLT16 architecture to partition closed and open source
Edgecore ASFvOLT16 Whitebox OLT

- BAL API is open, adapter can access all functionality
- GRPC provides natural boundary between open adapter and in-OLT driver linking to closed BAL API source code from Broadcom
ASFvOLT16 System Architecture Overview

- Broadcom BAL is an important building block but not a complete OLT solution
- More software is needed to manage HW not covered by BAL
  - Fan, PSU, Optical modules, etc.
  - Vital Product Data (S/N, Model, rev)
  - Software update/activate
- Anything not under control of BAL is handled by DMTF Redfish®
ASFvOLT16 Whitebox OLT

- Source code for ASFvOLT16 adapter
  - `git clone https://gerrit.opencord.org/voltha`
  - `voltha/adapters/asfvolt16_olt`
    - Interface adapter for ASFvOLT16
    - GRPC client/server connection classes
  - `voltha/adapters/asfvolt16_olt/protos`
    - Protobuifs for BAL object model and API
ASFvOLT16 Whitebox OLT

• Source code for ASFvOLT16 driver (voltha_bal_driver)
  • git clone https://gerrit.opencord.org/asfvolt16-driver
  • See src/README.md for build instructions
• The following components are required from Broadcom:
  • BAL/Maple SDK – version 2.4.3.6
  • Qumran SDK – version 6.5.7
  • ASFvOLT16 BAL patch – version ACCTON_BAL_2.4.3.6-V201710131639
  • Contact Dave Baron @ Broadcom reference case CS3233745
• ONL Build environment for ASFvOLT16
  • https://wiki.opencord.org/download/attachments/2556712/ONL_Bui
d_Environment_Installation_Guide.pdf?api=v2
Edgecore ASFvOLT16

- Possible deployment scenarios
  - Similar to Edgecore OF-DPA
  - Pre-built binary package from Edgecore, ONF or SI (system integrator)
Edgecore ASFvOLT16

• Future work
  • C++ based voltha_bal_driver
  • Use C++ based GRPC
  • In-band management support
  • OLT auto discovery
  • Expose more Maple features though BAL
  • Expanded use of syslog by voltha_bal_driver
Community Collaborators for ASFvOLT16

- Accton/Edgecore
- ALTEN Calsoft Labs
- AT&T
- Broadcom
- Ciena
- ONF
- Radisys
radisys

Thank You
BACK UP
ASFvOLT16 Whitebox OLT

- ASFvOLT16 Device Management proposal (VOL-248 epic)
  - Create a Redfish RESTful query and control service in OLT
  - Redfish server will provide access to functions such as:
    - Firmware update and activate
    - OLT graceful reboot
    - Monitoring of fans and power supply units (PSU)
    - Monitoring of XFP/QSFP optical modules
    - Session management
    - Extraction of vital product data
ASFvOLT16 Whitebox OLT

- ASFvOLT16 Device Management proposal (VOL-248 epic)
  - Create a Redfish client service in asfvolt16_olt adapter
  - Uses REST to access Redfish server in OLT
  - Redfish client will implement:
    - Session login
    - Get status of fans, PSUs, XFPs, QSFPs
    - Initiate and monitor firmware update and activation
    - OLT reboot
    - Submit alarms to VOLTHA
    - Configure board options (watchdog, SyncE, port modes)
Example:

Redfish
Server Data
Model for
Health Status
Community Collaborators for ASFvOLT16

• **Accton/Edgecore**
  - Reliable ASFvOLT16 hardware
  - Porting Broadcom BAL/Maple/Qumran SDK to ONL kernel for x86_64 arch
  - BAL patch and patch versioning/build documentation
  - Redfish server support from OCP design

• **Broadcom**
  - Provided the open-source API for the BAL framework
  - Distributing the ASFvOLT16 BAL patch via CSP
  - Supporting PoC demos and driver development
Community Collaborators for ASFvOLT16

• ONF
  - Edgecore adapter project kickoff
  - Proposal for GRPC device simulator w/Juniper C-GRPC
  - BAL API .proto files
  - Asfvolt16_olt.py adapter initial framework and adapter class

• AT&T
  - Use case and feature set required from BAL API

• ALTEN Calsoft Labs/Ciena
  - Proposal for OLT device management using Redfish client/server
  - Redfish client in asfvolt16_olt adapter using REST APIs
  - Data models for management and configuration
Community Collaborators for ASFvOLT16

• Radisys
  • Leading community asfvolt16_olt adapter and in-OLT driver initiative
  • Developing in-OLT BAL API control driver (voltha_bal_driver)
  • Worked directly with Broadcom to:
    • Develop understanding of VOLTHA community need for open BAL API
    • Negotiate required BAL API feature set
    • Define BAL package and patch distribution model suitable for SLA
  • Worked directly with Edgecore/Accton to:
    • Transfer PoC result on Broadcom reference OLT to ASFvOLT16
    • Develop BAL acceptance test to ensure voltha_bal_driver foundation
    • Define BAL patch components and versioning method