

Virtual OLT (vOLT)

- [Introduction](#)
- [Initial Authentication Packet Trace](#)
- [VLAN Architecture](#)
- [Software Components](#)
 - [vOLT Agent](#)
 - [ONOS applications](#)

Introduction

OLT devices terminate the optical distribution network (ODN) link in the Central Office, with each physical termination point aggregating a set of subscriber connections. This paper describes a virtualized OLT, called vOLT, that replaces proprietary OLT devices with a combination of commodity hardware and open source software. It focuses on Gigabit Passive Optical Networks (GPON), but is applicable to other technologies as well. For example, a G.Fast version is under development.

The first step is to create an I/O blade with the PON OLT MAC. AT&T is working with the Open Compute Project to develop an open specification for a GPON MAC 1RU "pizza box", as shown in Figure 1. This blade includes the essential GPON Media Access Control (MAC) chip under control of a remote control program, which is, in turn, controlled from high level applications via OpenFlow. This replaces an existing closed and proprietary OLT chassis (not shown) that integrates this GPON MAC chip with GPON protocol management, 802.1ad-compliant VLAN bridging, and Ethernet MAC functions. These other functions from the legacy device are being unbundled and implemented in software. The box shown in Fig.1 is frequently referred to as a white-box OLT, but it is important to note that **each port on the box is a separate PON OLT** with 32 or 64 subscribers.



Figure 1. GPON OLT IO Blade.

There are two pieces of software that work together to implement the vOLT functionality. The first is a vOLT agent runs in a container or VM and facilitates a connection between ONOS and the hardware. The agent exposes an OpenFlow interface northbound which enables it to be controlled by ONOS. It then maps OpenFlow messages to the native APIs of the hardware device and OMCI messages that manage the PON ONTs. The second piece of software is a set of functions that manage some of the control plane functions of a traditional OLT, like 802.1X, IGMP Snooping, VLAN bridging, and OAM. These control functions are implemented as applications running on top of ONOS, facilitate subscriber attachment, authentication (AAA), establishes and manages VLANs connecting consumer devices (see next section) and the central office switching fabric on a per-subscriber basis, and manages other control plane functions of the OLT.

The vOLT agent also performs a bootstrap operation. During this process the agent discovers the IO blade and establishes a control connection to it. The vOLT software then exposes an OpenFlow interface to ONOS who then uses that interface to program the vOLT agent. The agent converts these OpenFlow messages to OMCI to provision the OLT.

Initial Authentication Packet Trace

Figure 2 depicts the end-to-end system, with an OpenFlow controlled CPE in the home (along with an Optical Network Termination, ONT), racks of OLT blades in the Central Office, and a collection of commodity servers connected by a leaf-spine switching fabric constructed from a collection of white-box switches. The figure shows a Radius server running on one of the servers, and vOLT "running on top of" the CORD software stack. The next section discusses the software organization in more detail, but for the purposes of this discussion, vOLT is a "control app" running on ONOS.

When the Open OLT boots, it attaches to a vOLT application and also starts the GPON MAC. The system learns about ONTs that are attached and powered on, or learns about them later when they attached and power on.

Each ONT that is successfully attached to the PON is reported to the vOLT using an OpenFlow Port status message that indicates that new ports have been added to the GPON system. The message also includes the serial number of the ONT. This message allows extending the vOLT logic to deal with new and unexpected ONT attachments.

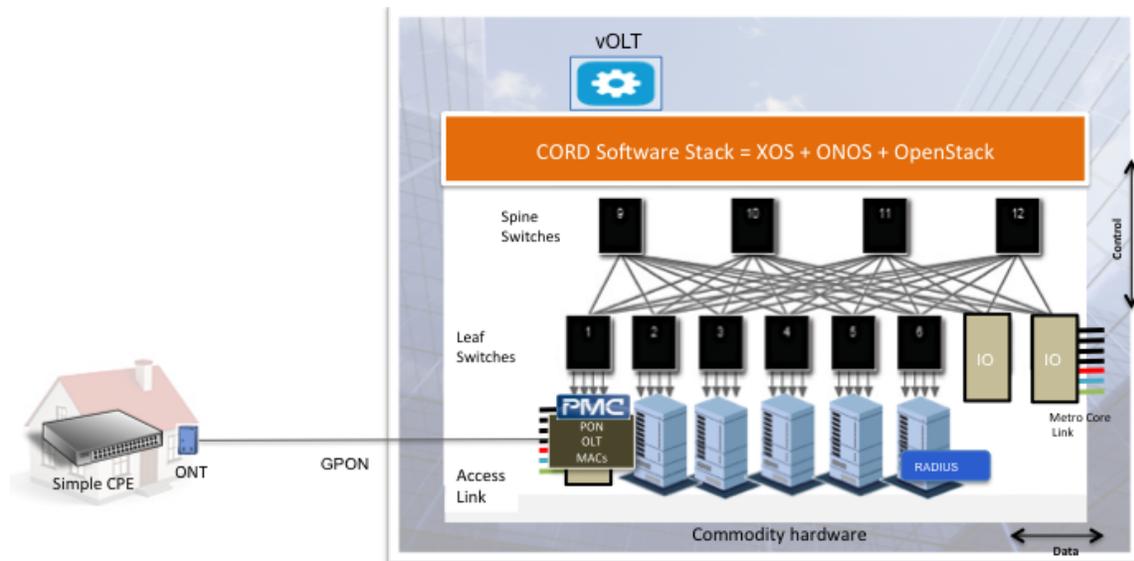


Figure 2. Basic CORD Hardware Configuration.

Once the ONT is attached, port up messages can be received for interfaces on the ONT that become active. This typically happens when the CPE is attached to the ONT with a physical cable, but it can also be the establishment of an internal interface when the two are combined into a single package.

At this point the vOLT applies an OpenFlow rule to the interface so that it gets any 802.1X packets presented by the CPE, but all other traffic is dropped.

When the CPE in the home first boots (or reboots), the packet sequence shown in Figure 3 fires:

1. The home CPE builds an 802.1x packet using a certificate that has been provisioned on it, and sends this packet upstream, where it is received by the OLT blade.
2. The OLT forwards the 802.1x packet to ONOS, which in turn delivers it to the vOLT control application.
3. The vOLT application uses the information in the 802.1x packet to query the RADIUS server, which authenticates the subscriber and returns the associated profile information provided the authentication was successful.

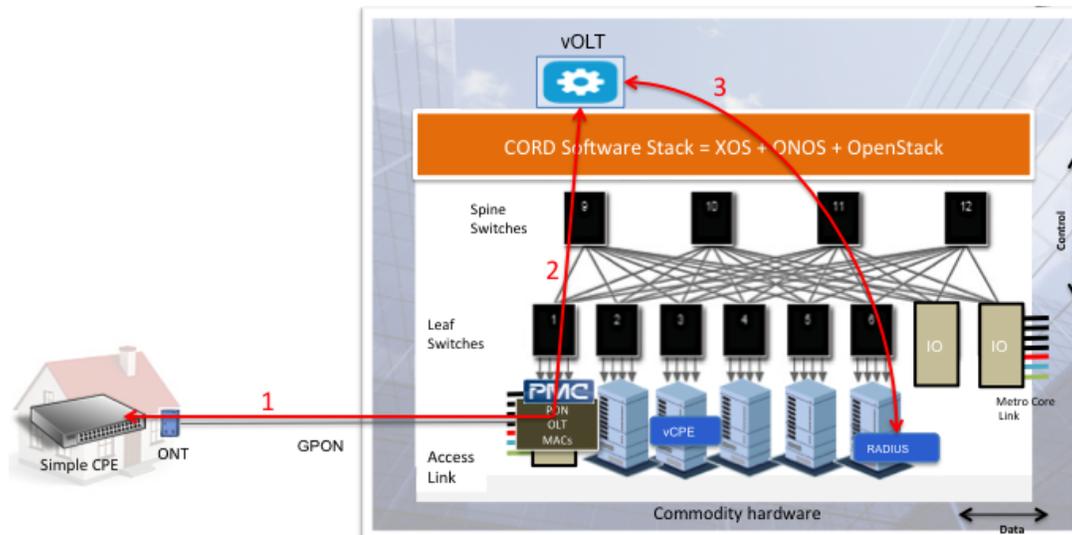


Figure 3. Initial packet exchange authenticating the subscriber.

At this point the vOLT control application sets a VLAN connecting the subscriber to vSG running on a commodity server in the Central Office. This involves:

1. Invoking a "create subscriber" call on XOS to spin up a vSG container on behalf of the subscriber.
2. Assigning a VLAN to the subscriber and informing the OLT that it should tag traffic with the appropriate VLANs.
3. Assigning packets with defined QoS markings to appropriate queues, both upstream and downstream.
4. Informing the ONOS control app managing the switching fabric (not shown) about the binding between vSG container and the VLAN.
5. Returning a successful authentication token to the CPE.

At this point user traffic can flow from any device in the subscriber's home to the vSG, tagged with the allocated VLAN. Such traffic flows from the CPE to the OLT blade, and then through the switching fabric to the vSG running on one of the commodity servers.

VLAN Architecture

Subscriber traffic is identified by two VLAN tags within the central office. The ONT and OLT hardware is responsible for tagging/untagging this traffic as it flows from/to the subscriber. ONOS instructs the OLT which VLANs to use through OpenFlow messages.

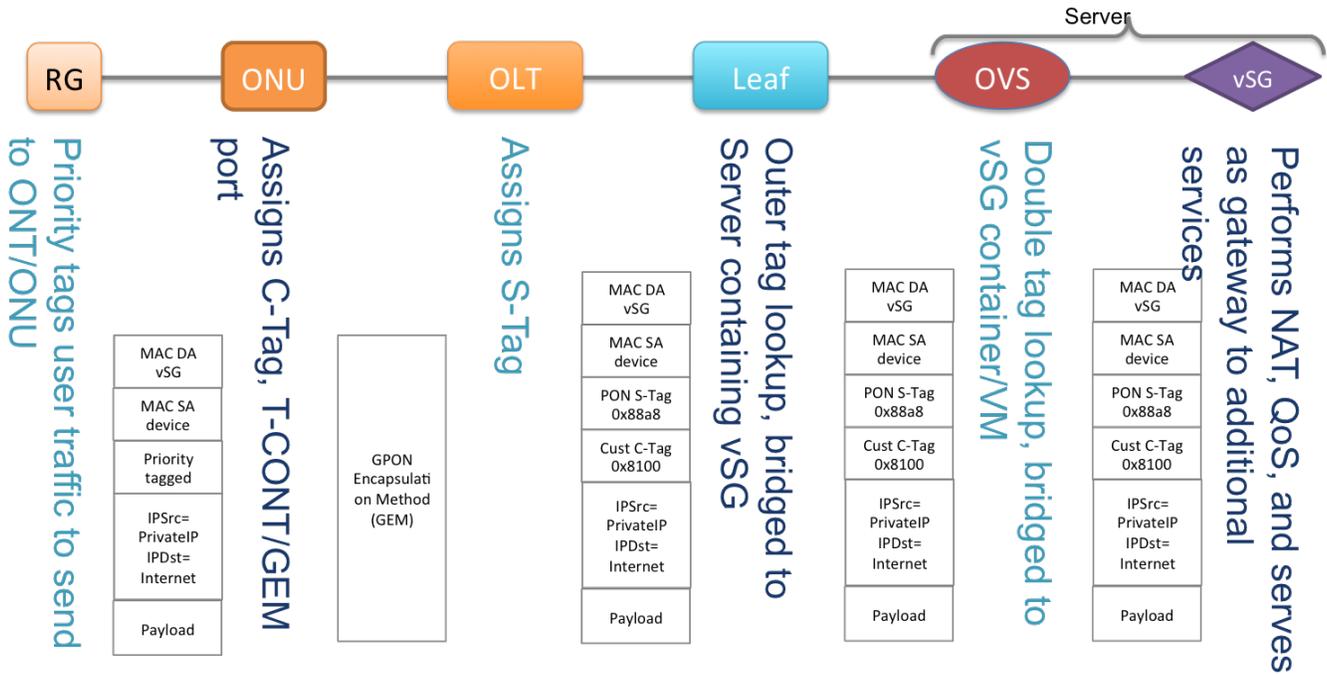


Figure 4. Packet walkthrough RG -> vSG

Figure 4 shows where VLAN tagging happens as traffic flows from a subscriber's home to the Internet. Within the home there are no VLAN tags. A priority VLAN tag (VLAN ID 0) is added at the physical home CPE/RG in order to carry Ethernet precedence bits. The OLT instructs the ONT (via OMCI messaging sent from the vOLT agent) to add/remove the appropriate C-Tag to the traffic inside the PON, and the OLT adds an S-Tag to the packet in order to uniquely identify the subscriber's traffic in the fabric (This is 1:1 VLAN mapping in GPON systems. Other mappings (eg. N:1) are also possible). The inner tag (C-tag) identifies the specific subscriber within the fabric. The outer tag (S-tag) identifies the PON. Taken together, these two tags can identify a subscriber uniquely across all OLT devices in the system. The fabric maps the outer VLAN tags to a server which hosts the vSGs for all the subscribers in the PON (i.e. outer Tag). In a future release, we will remove this limitation by adding the capability to place the vSG for a subscriber anywhere in the cloud. The vSG strips the VLAN tags, performs its functions, and then forwards the packet along the service graph. VLAN tags are not needed after the vSG because the vSG also NATs the traffic to the vSG's WAN address, which is unique in the system.

Software Components

vOLT Agent

The vOLT Agent is comprised of indigo, netconfd, the proprietary APIs for the PMC silicon and an OMCI stack. An interworking function maps OpenFlow and Netconf to the capabilities of the OLT and ONT through their respective APIs in order to emulate a distributed Ethernet switch to the OpenFlow controller. When the vOLT agent boots up, it connects to the hardware. It then initiates an OpenFlow connection to the controller. As mentioned before, the agent abstracts the entire PON system as a single switch to the controller. Each ONU appears as a separate OpenFlow port(s), even though multiple ONUs are connected to the same physical PON port, and there is a single port for the uplink connection. The agent is able to understand a limited subset of OpenFlow messages from the controller and configure the hardware appropriately.

ONOS applications

- vOLT application (onos-app-olt): Responsible for configuring VLAN tags on the OLT
- AAA application (onos-app-aaa): Responsible for brokering the authentication between the residential gateway (home CPE) and the Radius server. When the subscriber has authenticated, the application will call into XOS to spin up the services for the subscriber.
- Multicast: Performs IGMP snooping and adds/removes OLT ports to/from multicast groups.