SCTE CABLE-TEC EXPO 22
SEPTEMBER 19-22 • PHILADELPHIA, PA

2022 Fall Technical Forum
SCTE • CABLELABS • NCTA

Top-5 Things You Should Know About Operating a Virtual CMTS

A Technical Paper prepared for SCTE by

Brady Volpe
Founder
NimbleThis and The VolpeFirm
3000 Old Alabama Rd. Suite 119-434, Alpharetta, GA 30022
404.954.1233
brady.volpe@nimble-this.com :: brady.volpe@volpefirm.com

Thuy Nguyen
Cable Segment Lead
Intel Corporation
2200 Mission College Blvd, Santa Clara, CA 95054
thuy.nguyen@intel.com

Muhammad Siddiqui
Platform Solution Architect
Intel Corporation
2200 Mission College Blvd, Santa Clara, CA 95054
muhammad.a.siddiqui@intel.com

Michael Lafser
Technical Solution Engineer
World Wide Technology
1 World Wide Way, Maryland Heights, MO 63043
Mike.Lafser@wwt.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

1. Introduction

This paper will discuss the top-5 things an operator should know about operating a virtual cable modem termination system (vCMTS). Specifically, five key aspects will be discussed including deployment, scalability, capacity, troubleshooting, multitasking, power, and space needed for operating the vCMTS. The reader will understand that legacy CMTSs and vCMTSs are similar in most cases, especially from an operational and deployment standpoint. This means that if a user is familiar with operating a legacy CMTS then they should be comfortable migrating to a vCMTS. Further, the paper will explore some advantages that may make vCMTSs attractive to cable operators over legacy CMTSs. By the end of the paper the reader should have confidence in understanding that vCMTSs are quite like legacy CMTSs from a features and performance standpoint, while adding other valuable benefits and functionality that the reader may not have considered prior to reading this paper. Sit back and enjoy the read.

2. DOCSIS Primer

Data-Over-Cable Service Interface Specifications (DOCSIS®) technology is effectively a transparent Ethernet bridge over a hybrid fiber/coax (HFC) network. There are two functional components in a DOCSIS network, the cable modem (CM) on the subscriber side and the CMTS in the headend or hub site. The CMTS communicates with the CMs on one or more single carrier quadrature amplitude modulation (SC-QAM) channel(s) and/or orthogonal frequency division multiplexed (OFDM) channel(s). Data is on these channels is digitally encoded on radio frequency (RF) signals on the downstream path of an HFC network between 108 and 1.2 gigahertz (GHz). The CMs communicate with the CMTS using one or more SC-QAM and/or orthogonal frequency division multiple access (OFDMA) digitally encoded RF channels, transmitted on an upstream HFC frequency between 5 to 204 MHz (note these frequencies will change with DOCSIS 4.0). The digital data contains DOCSIS management information in addition to subscriber traffic. The CMTS is the system scheduler which coordinates the power level, frequency, transmit time, and pre-equalization of all CM signals on the DOCSIS network.

3. CMTS Overview

In any CMTS you have the hardware, the operating system software, the application software, and switching equipment. A vCMTS is made up of these same components by integrating the operating system, management software, and application software on a commercially available server.

What is a legacy CMTS as defined in the introduction of this paper? A legacy CMTS is a chassis based CMTS containing downstream and upstream RF cards utilizing on-board software and/or firmware for system operation. In DOCSIS 3.1, legacy CMTSs may be augmented to support distributed access architecture (DAA). In this scenario, the legacy CMTS may not have on board RF cards, but instead the RF and physical layer (PHY) portion will extend to a remote PHY (R-PHY) node, but the medium access control (MAC) processing is still performed in the legacy CMTS.

A virtual CMTS or vCMTS on the other hand, is software that is purpose built to run on commodity servers. Proprietary vCMTS software is installed on the server that converts the server into a vCMTS platform. A network interface card (NIC) attached to a switch connects the server to an R-PHY node or shelf and a DAA architecture is created, like the DAA architecture described with the legacy CMTS.

Figure 1 shows a high-level architecture of the cable access network showing both a converged cable access platform (CCAP) and a vCMTS with DAA deployments.

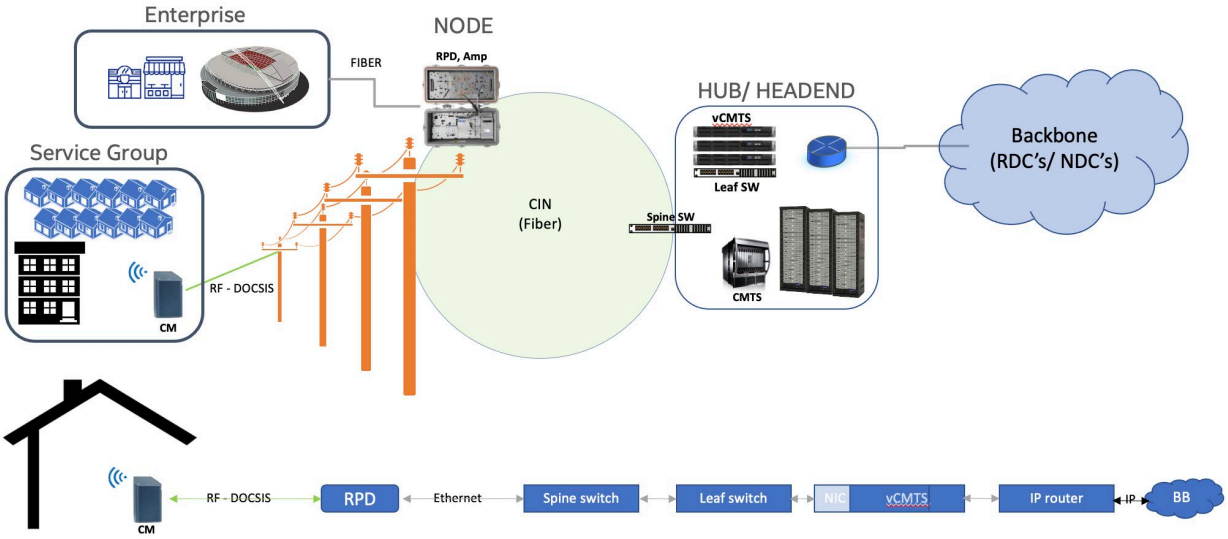

Figure 1 - Cable Access Network Diagram

4. How do I deploy?

The first aspect of turning up any DOCSIS HFC plant, be it legacy CMTS or vCMTS will be the initial deployment. While some aspects differ, particularly when comparing a legacy CMTS to a vCMTS initial deployment, many of the same setup procedures will be the same, just taking different forms. This makes sense because the functionality of a CMTS is just reproduced through virtualization of the CMTS. One area that greatly differs, is the amount of physical setup required for each device since legacy CMTS products require a large amount of either coax (analog) or fiber (digital) for the data ports, along with a much larger physical weight needing to be racked, as can be seen in Table 1.

Table 1 - Physical attribute comparison of different CMTS systems

	I-CCAP	D-CCAP	vCMTS
Data Cables	186 (RF) + 4 (Fiber)	68 (Fiber)	12 (Fiber)
Weight	429 lbs	429 lbs	36 lbs/server (3 servers = 108 lbs)
Power Cables	20 terminal cables	20 terminal cables	6 pluggable cables

Note:

- Weight is inclusive of the equipment such as chassis, line cards, power supplies, etc. Items such as optics, combiners, cables, are excluded from the weight calculations.
- D-CCAP may be several pounds lighter due to PHY modules being removed.

4.1. Legacy CMTS Deployment

Initial deployment of a legacy CMTS begins after the gear has been rack mounted, all linecards and physical NICs installed, and the devices are cabled up. Once the device is up and running, a console terminal is opened, and the management interface, security, host information, and secure shell host (SSH) access are configured through Command Line Interface (CLI). The software and firmware are then upgraded to the proper versions. Networking is then configured with wide area network (WAN) interfaces, loopback interfaces, and routing all being configured for the CMTS to communicate with the network elements (NE). Connectivity, routing, and networking are then verified, completing the initial setup.

DOCSIS configuration can now begin after the initial deployment is completed. There are multiple items that will need to be configured through CLI for the system before service group (SG) provisioning can occur. The first item to be configured is the quadrature amplitude modulation (QAM) profiles; an example of this is shown in Figure 2.

```
Router(config)#cable downstream qam-profile 13
Router(config-qam-prof)#  annex A
Router(config-qam-prof)#  modulation 256
Router(config-qam-prof)#  interleaver-depth I12-J17
Router(config-qam-prof)#  symbol-rate 6952
Router(config-qam-prof)#  spectrum-inversion off
Router(config-qam-prof)#description new-256-qam
Router(config-qam-prof)#
```

Figure 2 - QAM profile configuration

These profiles contain the physical layer information for the downstream (DS) channels in the system. OFDM and frequency profiles are configured, providing the information required for DS channel configuration for the DS ports, as seen below in Figure 3.

```
Router(config)#controller Integrated-Cable 3/0/0
Router(config-controller)# max-ofdm-spectrum 192000000
Router(config-controller)# max-carrier 32
Router(config-controller)# base-channel-power 36
Router(config-controller)# rf-chan 0 31
Router(config-rf-chan)#  type DOCSIS
Router(config-rf-chan)#  frequency 801000000
Router(config-rf-chan)#  rf-output NORMAL
Router(config-rf-chan)#  power-adjust 0
Router(config-rf-chan)#  qam-profile 1
Router(config-rf-chan)#  docsis-channel-id 1
Router(config-rf-chan)# !
Router(config-rf-chan)#
Router(config-rf-chan)#rf-chan 158
Router(config-rf-chan)#  power-adjust 0
Router(config-rf-chan)#  docsis-channel-id 159
Router(config-rf-chan)#  ofdm channel-profile 50 start-frequency 837000000 width 192000000 plc 930000000
```

Figure 3 - DS controller configuration

This process is repeated for the upstream (US) side with modulation and OFDMA profiles being configured.

With the profiles now created, the controllers can then be configured by applying the profiles to them. With the controllers configured, the Cable, DS Cable, and US Cable interfaces are configured by associating the controllers to them, configuring the MAC-Domains as the channels get bound to it. Configuring SGs continues by creating a Fiber Node by assigning the US and DS cables to it, which creates and associates the US SG and DS SG to the MAC Domain. The CMTS's RF output is checked at the head end to ensure it falls within specified values, and the system is connected to the RF passive equipment. To monitor the system, simple network management protocol (SNMP) is configured and connected to an external software source.

4.2. vCMTS Deployment[1]

Initial deployments of a vCMTS cluster consist of a few steps. The server hardware, networking equipment (routers, switches) need to be racked and cabled up, with their management and baseboard management controller (BMC) IP addresses configured, along with the required network configurations so that access is allowed. From there, an external computer, or virtual machine with network access to the devices can be connected and loaded with an automation tool such as Ansible to act as a deployer. The vCMTS software images, operating systems and Ansible playbooks get loaded on to the deployer. A configuration yet another markup language (YAML)[2] file is created containing all the host information and passwords that are required for the vCMTS cluster deployment. A playbook is then executed, deploying the operating systems, vCMTS software packages, and containers which are automatically deployed and synced for the cluster. After the cluster is deployed, they are connected but unconfigured to talk to the rest of

[1] Much of this section, and its figures, are specific to Cisco's cloud Native Broadband Router (cNBR) but has been generalized where available.
[2] YAML is a human-readable data-serialization language. It is commonly used for configuration files and in applications where data is being stored or transmitted. YAML targets many of the same communications applications as Extensible Markup Language but has a minimal syntax which intentionally differs from SGML. (source: https://en.wikipedia.org/wiki/YAML)

the network elements. This configuration can occur through applying a pre-made JavaScript Object Notation (JSON) configuration files, or directly configuring the NE and networking addresses in a graphical user interface (GUI). These will include all the core addressing, precision time protocol (PTP) information, routing information, etc. for all interface elements, an example is shown below in Figure 4.

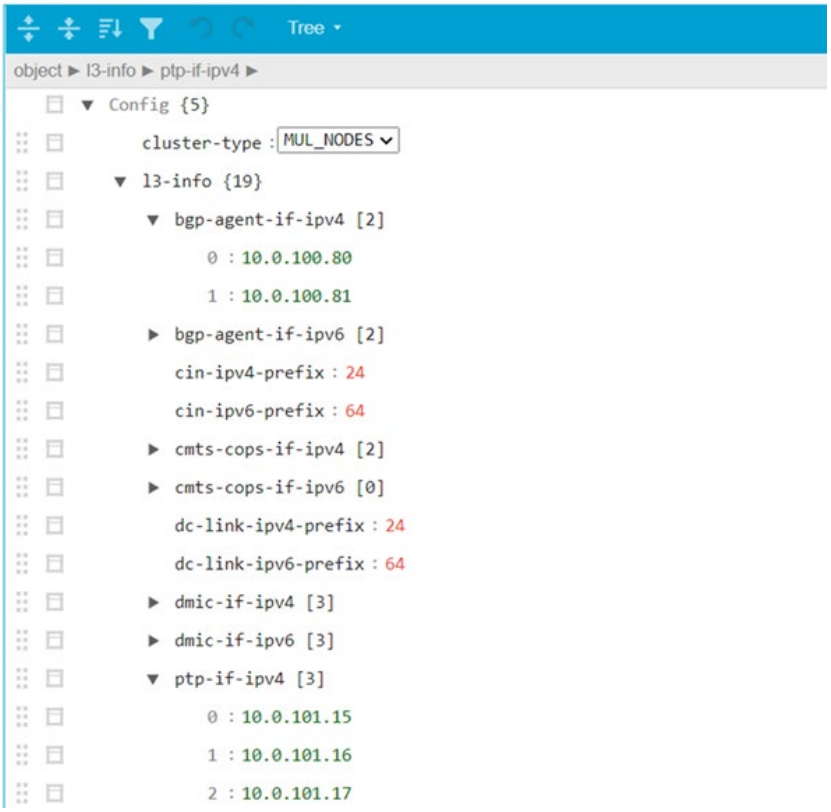

Figure 4 - GUI configuration example of interfaces

Once all the prerequisite day zero configurations are completed, and the vCMTS core cluster is talking to the NE, the basic network setup is completed, and it should be verified that all the NEs can talk to each other, and the systems' PTP clock is locked. This can be accomplished through various means, from ensuring ping capability, checking border gateway protocol (BGP) neighbors and routing tables.

DOCSIS configuration can begin after verification of connectivity. This is once again very similar to a legacy CMTS deployment, and all the elements that would go into configuring a CMTS SG. Profiles for the various elements, the DS channels, US channels, remote PHY device (RPD) PTP information, etc. will all need to be created. These will contain the same information as in a legacy CMTS, as Figure 5 illustrates. This can again be done through importing JSON files into the system, allowing for automation in the configuration process, or through use of a GUI interface.

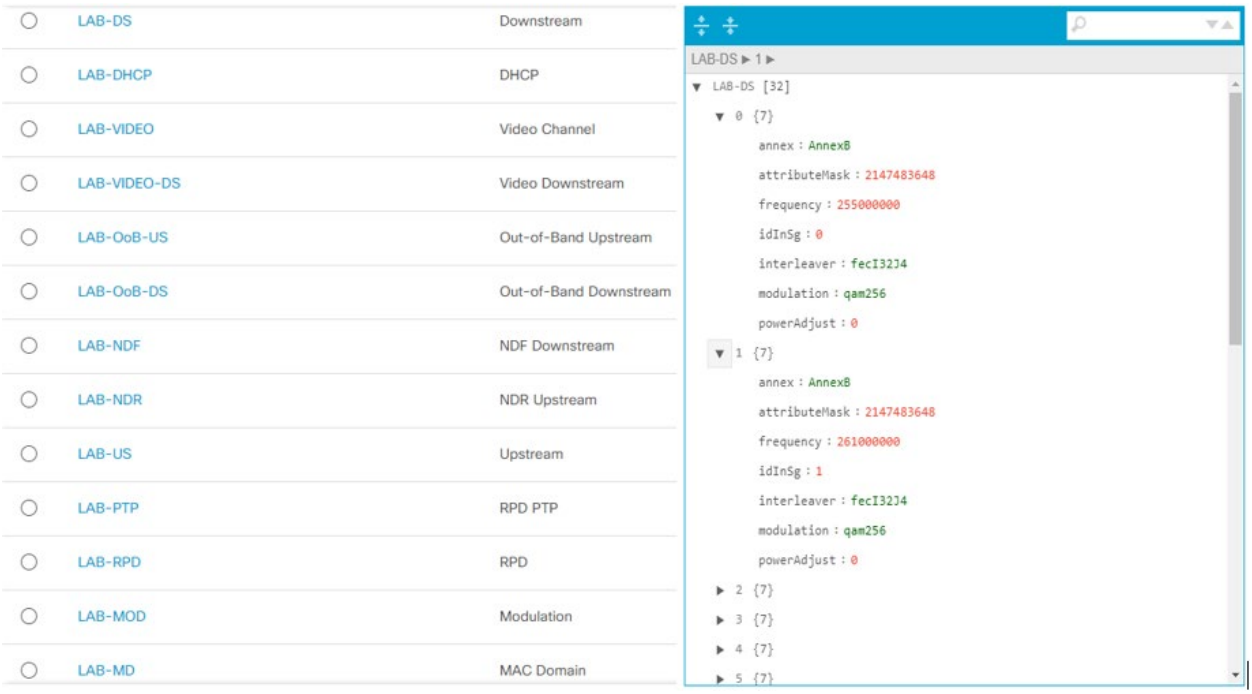

Figure 5 - Profile List (Left) and Sample Downstream GUI profile

Now that all the prerequisite DOCSIS and dynamic host configuration protocol (DHCP) profiles are in the system, SG templates can be created from the profiles, likewise, Layer 3 templates, acting as a cable helper configuration, containing the information for DHCP relay, and BGP peer information, can be created. These are, once again, like a legacy CMTS system and contain all the information required for provisioning.

Unlike previous sections, provisioning SGs in a vCMTS differs from a legacy CMTS. In this case, a separate RPD configuration containing the core information, and DS/US cable information is not needed, as these are configured automatically once a RPD (and consequently a SG) is created. Utilizing a GUI based interface, provisioning is also simplified. Here, the RPD MAC information, names, and the previously created templates provide the necessary information required to provision a SG. As shown in Figure 6, the vCMTS software then checks the configuration, creates the required cabling and routing information, then checks the online status before posting a status report.

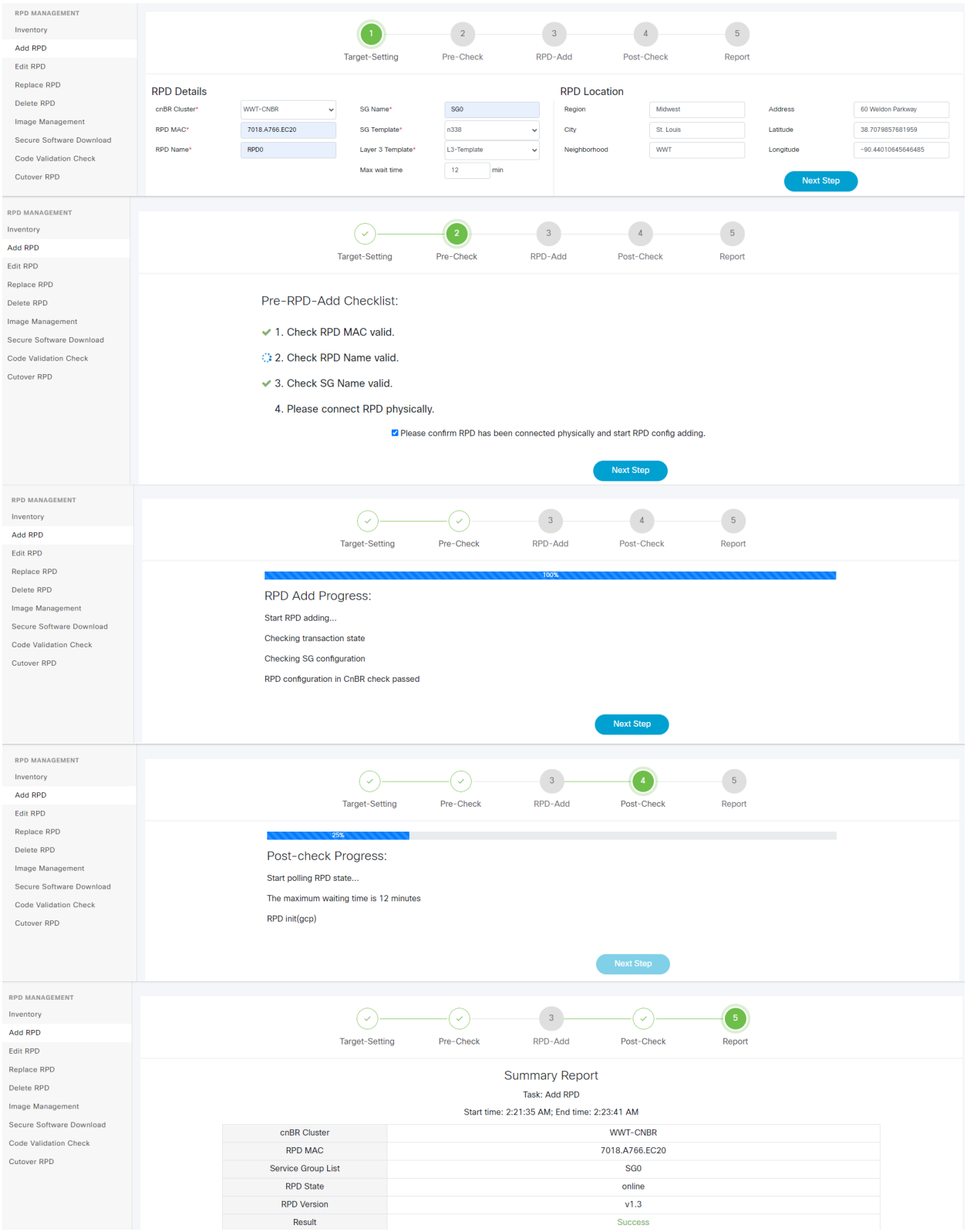

Figure 6: RPD/SG provisioning process

While this is a convenient and easy way to provision a SG, for ongoing deployments this would be quite tedious. However, since this is a vCMTS, it is an application programming interface (API) driven setup, provisioning a large amount of RPDs can be accomplished using API calls directly to import the configuration files into the system. As was previously stated, the routing and interfaces are configured automatically by the software, these interfaces are the same ones that would be used in legacy CMTS systems, while the routing is the information for the traffic and DHCP relay, and an example is shown in Figure 7.

BGP v4 Route Table

SG Name	SG ID	VRF	Prefix	Nexthop
SG0	0	default	10.0.122.1/32	10.0.100.103
SG0	0	default	10.0.122.1/24	10.0.100.2
SG0	0	default	10.0.121.1/32	10.0.100.103
SG0	0	default	10.0.121.1/24	10.0.100.2
SG0	0	default	10.0.120.59/32	10.0.100.2
SG0	0	default	10.0.120.1/32	10.0.100.103
SG0	0	default	10.0.120.1/24	10.0.100.2


```
{
  "cluster-id": "10.254.154.173.nip.io",
  "interface-list": [
    {
      "if-name": "10.254.154.173.nip.io_SG0_Cable0",
      "if-type": "DocsCableMaclayer",
      "if-mac": "84:b2:61:59:00:01",
      "if-stack-status": "Up",
      "if-in-octets": 0,
      "if-out-octets": 0
    },
    {
      "if-name": "10.254.154.173.nip.io_SG0_Upstream0",
      "if-type": "DocsCableUpstream",
      "if-mac": "",
      "if-stack-status": "Up",
      "if-in-octets": 0,
      "if-out-octets": 0
    },
```


Figure 7 - Routing (Left) and interfaces (Right) created

After the SG for the system has been provisioned, the configured elements need to be verified working as intended. As with legacy CMTS systems, the RF output of the node (RPD) in the field should be checked with probes. With a vCMTS system, however, many of the monitoring systems have been simplified due to the information being pulled from containers and direct information gathering by streaming telemetry viewable through available dashboards, shown in Figure 8.

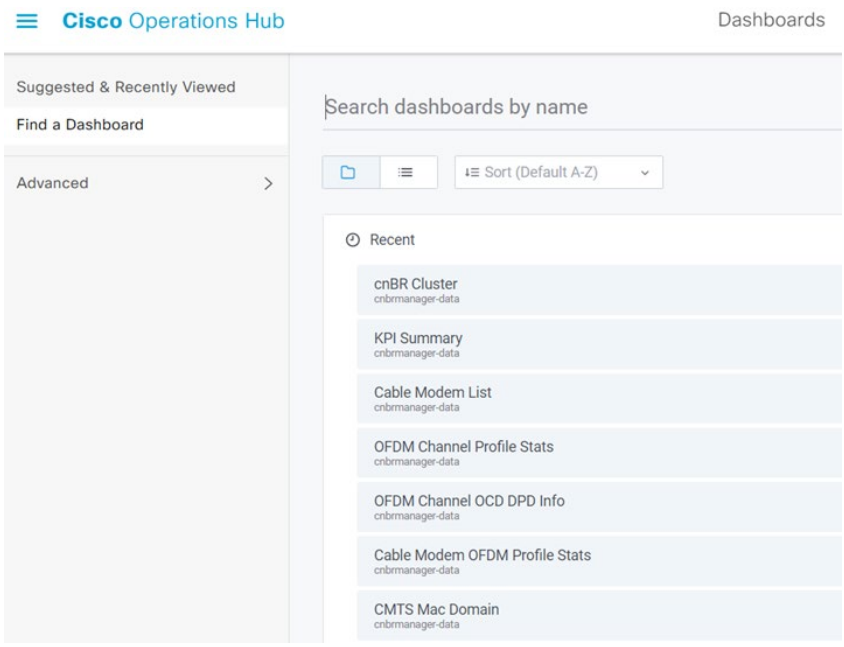

Figure 8 - Dashboard navigation

A few examples of this can be seen below, with Figure 9 showing the overall summary of the system, and Figure 10 showing information of an individual SG.

Figure 9 - System Cluster Summary

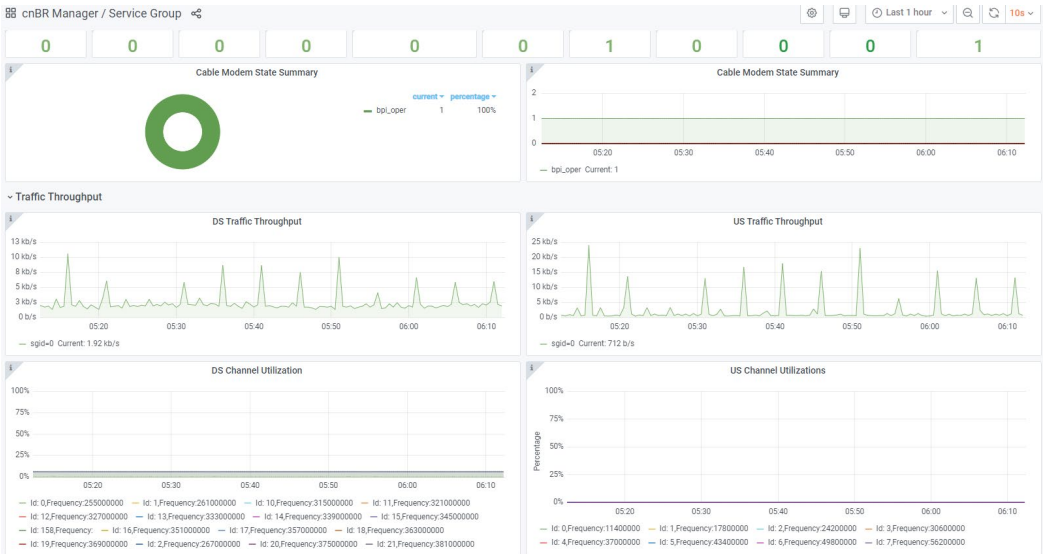

Figure 10 - Service group summary

Virtual CMTS deployments are heavily containerized architectures, with each container acting as an individual application. These containers are orchestrated using the open-source API Kubernetes (often referred to as K8s), or other container management platforms. These containers are applications running the individual components of the overall vCMTS architecture, such as packet processing, packet cable, telemetry, OFDM, SG management, MAC scheduler, etc. While all these containers are running in the background on a vCMTS cluster, they are not needed to be touched or managed by an operator as Kubernetes manages them and their services. So, while a vCMTS does run as all these individual pieces, it operates fundamentally the same as a legacy CMTS. After all, DOCSIS is DOCSIS.

5. How Do I Scale and Add capacity?

A common scenario with any CMTS deployment is increasing capacity. Traditionally, year-over-year growth has been roughly 20% in the downstream and upstream as can be seen in Figure 11. This traffic growth is projected to continue for the foreseeable future. The impact on any CMTS, legacy or virtual, is that the CMTS must be scalable and capable of supporting the continued traffic growth.

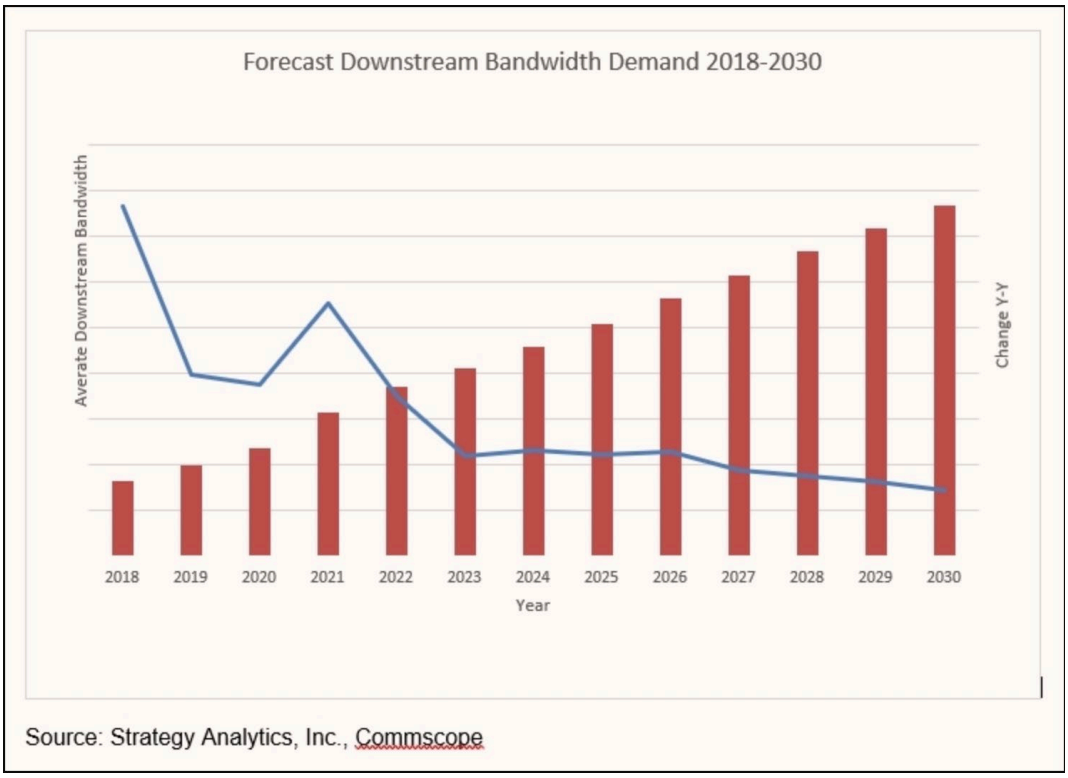

Figure 11 - Forecast Downstream Bandwidth Demand 2018-2030 (Source: Strategy Analytics, Inc., CommScope)

5.1. Capacity Expansion Legacy CMTS

Capacity expansion in a legacy CMTS is dependent on the CMTS vendor, though ultimately a well-defined process. Legacy CMTSs will have a finite number of downstream and upstream SC-QAM and OFDM/OFDMA channels. These channels maybe combined to create service groups. A SG is defined as a set of downstream channels and upstream channels which are further associated logically in the CMTS as a MAC domain. Downstream and upstream communication over the downstream and upstream RF channels to a specific fiber node will be associated together as traffic. One can think of this as a shared data pipe in the downstream and shared data pipe in the upstream. Similarly, on the CMTS, each individual channel in the downstream is shared or "utilized", provided the CMTS is properly configured. The same holds true for the upstream.

Consider a downstream with 32 SC-QAM channels at 256-QAM. Assuming a 256-QAM channel supports approximately 38 Mbps after overhead, the total downstream capacity would be:

32 x 38 Mbps = 1.216 Gbps

This is a common CMTS downstream configuration prior to DOCSIS 3.1 OFDM. When subscribers on this fiber node saturate the fiber node in the downstream with continuous traffic during peak hours (which is about 1.2 Gbps) chaos will break out and the CSR phone lines will

start to ring. Subscribers will experience poor quality of experience due to network congestion. This will eventually generate a work order for a node split on that fiber node.

Node splits are expensive and time consuming. One or more new fiber nodes will be installed to reduce the number of subscribers per SG. The CMTS must then be updated to add a new SG(s) for the new fiber node(s). This is a common and ongoing process in most DOCSIS networks. DOCSIS 3.1 has enabled cable operators to delay fiber splits by adding DOCSIS 3.1 OFDM channels in the downstream. These channels can add up to 1.89 Gbps per 192 MHz OFDM block. In either case, the CMTS must support the addition of licensing for more SC-QAM and/or OFDM channels for the SG. It is possible that the CMTS will need additional hardware such as line card(s). The line card(s) provide the conversion of data to RF signals which are referred to as SC-QAM and OFDM channels. If the CMTS is already completely full and no additional slots are available for new line cards, then the cable operator would need to buy a new CMTS chassis and perform the steps mentioned in section 4.1.

Once the CMTS has been licensed and upgraded with hardware, someone with knowledge of the CMTS and required SG update must program the CMTS. The programming is similar to that in section 4.1. CLI programming is not consistent across CMTS vendors. But in general, the concept is the same, driven by DOCSIS specifications. Some cable operators have automated the process to the point where a SG can be created by a script. These scripts may need to be updated as networks evolve and CMTS code changes.

5.2. Capacity Expansion vCMTS

One of the benefits of migrating from a traditional hardware-based legacy CMTS appliance to a software-based containerized vCMTS solution is the agility and time required to add capacity. Also, a legacy CMTS requires more rack space than a vCMTS. If a legacy CMTS is fully populated, then to add capacity, it may require an additional legacy CMTS chassis. A vCMTS on the other hand, that runs on common off the shelf (COTS) servers and may need as little as 1 rack unit (RU) of rack space to add new SGs. This is a significant space savings.

Kubernetes has gained popularity over the years and has now arguably become the de-facto standard for container orchestration and its lifecycle management. Most, if not all, vCMTS core offerings from independent software vendors (ISVs) are orchestrated and managed using Kubernetes APIs. As the existing Kubernetes cluster starts to run at maximum capacity and resources start to run out, to add more SGs, new K8s worker nodes can be added to an existing cluster that runs vCMTS applications. A single K8s cluster can run several hundred worker nodes though officially K8s claim to support up to 5000 worker nodes.

Adding new worker nodes in the headend is relatively easier if the management network and simple IT infrastructure are properly set up for remote access. When a new server with desired hardware specifications is acquired, it is mounted in the rack and connected to the management and data plane networks. An out-of-band management interface can be used to update basic input/output system (BIOS) and enable required hardware features (e.g., Hyperthreading), single root input/output (I/O) virtualization (SRIOV) etc. The new server can either have a pre-installed operating system (OS) with custom configurations (IP, ssh keys, etc.) before it's racked up, or the

OS can be installed remotely after the server is mounted in the rack. This paper presents one example of how a new server with no pre-installed OS can be provisioned in the virtualized cable headend which has a DHCP and preboot execution environment (PXE) boot server with desired OS image already setup. As the new server boots up for the first time it can be booted from the network using a PXE enabled NIC. This will let the OS image be downloaded from the PXE boot server (trivial file transfer protocol [TFTP] server) and install the OS on the newly provisioned server. The OS image can be customized beforehand to enable remote ssh access before its uploaded on to the TFTP server. Once the initial OS is installed using PXE, the server can then be accessed remotely and ready to be added as a new worker node to an existing vCMTS K8s cluster.

There could be multiple ways to add new nodes to a Kubernetes cluster but broadly it can be divided into two different approaches 1) manual 2) automated.

A manual installation approach would require all K8s node software components to be installed and configured individually. This could be a tedious task and prone to errors and doesn't reap the real benefits of virtualizing a cable headend.

A better approach is to adopt automation which significantly reduces time and effort to deploy and scale, while improving quality by preventing human errors. One popular automation tool adopted widely today is Ansible. Ansible is an agentless automation tool that runs Ansible "playbooks" from a remote Ansible controller to perform cluster provisioning, configuration management and performs various actions on remote machines. Ansible controller communicates with remote/host machines over SSH protocol. Ansible playbooks are configuration files that define tasks to be performed on host machines and let users customize multiple parameters to fit their installation needs. Playbooks can be used to deploy new K8s clusters and add new K8s nodes to a running cluster. The whole process of adding new vCMTS nodes takes a few hours if playbooks are configured properly.

Another automation tool widely used today with K8s is Helm. Helm is a package manager tool that runs on top of K8s to automate the installation process of plugins and K8s capabilities and used to deploy and manage applications like vCMTS applications. As the new K8s node is added, Helm can be used to deploy new vCMTS service groups to the newly provisioned K8s node. Scaling out new SGs to a new node consists of running a few K8s CLI commands that spins up new vCMTS SGs. The rest of the SG configurations are the same as previously explained in section 4.2.

6. How do I Troubleshoot the CMTS?

An important aspect of any CMTS is ensuring that subscribers continue to send and receive high speed data without interruption. Being able to identify if the CMTS is having a hardware and/or software issue or if there is an impairment in the plant impacting subscribers is critical. Troubleshooting the CMTS is a feature which must not be overlooked in any purchasing decision of a CMTS. If one is unable to quickly troubleshoot the CMTS and network, then the CMTS is failing at its main mission of delivering data to and from the subscriber.

6.1. Troubleshooting a Legacy CMTS

Legacy CMTSs have had over 20 years for vendors to harden the CMTS, identify internal bugs and resolve them. However, over the past 20 years the DOCSIS specification has continued to evolve meaning more complexity, features, higher density, and increased power use. Problems can and do occur in the CMTS hardware as hardware degrades over time. Legacy CMTS vendors include CLI and SNMP commands that allow one to continuous monitor all aspects of the CMTS for hardware failures.

6.1.1. Hardware Failures

As an example, a typical scenario may be the failure of a power supply. Using simple CLI commands one can see the status of all power supplies and the total power consumed by the CMTS. Figure 12 shows that the current CMTS supporting roughly 5k subscribers with 42 service groups is consuming about 4.4kW of power.

```
[redacted]-cbr8#show environment power
=============================
Slot    Controller       Value
-----------------------------
 P0       PEM Power     771 W
 P1       PEM Power     790 W
 P2       PEM Power     782 W
 P3       PEM Power     771 W
 P4       PEM Power     785 W
 P5       PEM Power     760 W
-----------------------------
Input Power Summary:   4659 W
=============================
 1        FRU Power     380 W
 7        FRU Power     390 W
 3        FRU Power     380 W
 9        FRU Power     370 W
 2        FRU Power     380 W
 6        FRU Power     380 W
 8        FRU Power     370 W
 R1       FRU Power     693 W
 R0       FRU Power     725 W
 0        FRU Power     320 W
-----------------------------
Power Consumed Summary:4388 W
=============================
More Cards can be supported:
-----------------------------
LC:                         0
=============================
```

Figure 12 - Power Overview of Legacy CMTS (4.388 kW used)

Figure 12 is only a high-level summary. One can drill down to detailed diagnostics of the overall powering of the system. Figure 13 shows a sample of another typical CMTS command where details of temperature and voltages can be observed on the legacy CMTS.

```
[redacted]-cbr8#show environment all
Sensor List:  Environmental Monitoring
 Sensor          Location         State            Reading
 12_CUR: Sens    1                Normal           20 mV
 12_CUR: Vin     1                Normal           12725 mV
 12_CUR: ADin    1                Normal           271 mV
 G0_CUR: Sens    1                Normal           73 mV
 G0_CUR: Vin     1                Normal           12575 mV
 G0_CUR: ADin    1                Normal           0 mV
 G1_CUR: Sens    1                Normal           74 mV
 G1_CUR: Vin     1                Normal           12625 mV
 G1_CUR: ADin    1                Normal           0 mV
 LB_CUR: Sens    1                Normal           20 mV
 LB_CUR: Vin     1                Normal           12575 mV
 LB_CUR: ADin    1                Normal           0 mV
 Temp: CAPRICA   1                Normal           54 Celsius
 Temp: BASESTAR  1                Normal           58 Celsius
 Temp: RAIDER    1                Normal           55 Celsius
 Temp: CPU       1                Normal           34 Celsius
 Temp: INLET     1                Normal           25 Celsius
 Temp: OUTLET    1                Normal           46 Celsius
 Temp: DIGITAL   1                Normal           35 Celsius
 Temp: UPX       1                Normal           45 Celsius
 Temp: LEOBEN1   1                Normal           48 Celsius
 Temp: LEOBEN2   1                Normal           50 Celsius
```

Figure 13 - Example of Detailed Powering and Temperature Statistics on Legacy CMTS

In Figure 13 it is evident that much greater detail is available on legacy CMTSs. This data can be extracted from the CMTS and stored in a database, then plotted over time. Trends which would indicate the CMTS is failing giving the cable operator time to replace the failing element or the entire CMTS if necessary. Further, nearly every element of a legacy CMTS is redundant. These include power supplies, processing cards, NICs, and even the RF cards (if desired). It is standard for legacy CMTSs to operate in high availability (HA) mode, such that if any single component or element fails, there is a backup to take over. This generally makes legacy CMTSs very reliable.

6.1.2. Software Failures

Because legacy CMTSs runs on proprietary code, which for most vendors has been built on years of development, software failures are uncommon. This does not mean that there are not compatibility issues between CMTSs and cable modems. It does happen and more frequently it has occurred in new standards, such as with DOCSIS 3.1 OFDMA. These are not so much software failures as they are software “issues”.

6.1.3. Subscriber Impairments

Troubleshooting subscriber impairments is a key feature for any legacy CMTS. Because legacy CMTSs are standards-based, they support standard management information base (MIBs) providing rich access to SNMP data. This SNMP data is used by reactive and proactive monitoring systems which provide the health and key performance indicator (KPI) metrics of each subscriber as well as the aggregate metrics of the network. Reactive and proactive monitoring systems give cable operators the visibility to know when subscribers are having problems or will have problems in the future. The aggregated view also enables cable operators to have visibility into the total traffic consumption on any given fiber node or an entire CMTS. This is important information to monitor and trend over time so that capacity expansion may be planned, as discussed in section 5.1.

SNMP can create a load on legacy CMTSs. Reactive and proactive systems need a lot of data from CMTSs to do their job. At the same time, legacy CMTSs have finite processing power, of which most is dedicated towards managing subscriber traffic. SNMP queries to the CMTS will impact the legacy CMTS processor. If there is too much strain on the CMTS from subscriber traffic, excessive SNMP queries may result in no response to the SNMP queries or even worse, SNMP queries may impact subscriber traffic. If the former, no SNMP data is returned, then the monitoring system will return false data, possibly causing the cable operator to react to bad data. In the latter case, excessive SNMP queries have been known to impede subscriber traffic and even crash a CMTS. This is quite serious and legacy CMTS vendors, monitoring software vendors and cable operators must be aware of these concerns and always ensure their CMTS are not overloaded by either subscriber traffic or SNMP utilization.

6.2. Troubleshooting the vCMTS

One of the benefits of virtualizing a cable headend is that it enables cable operators to collect huge amount of telemetry - much more than what was traditionally exposed by hardware appliance network functions - including hardware, vCMTS software, and environmental equipment. If the telemetry from vCMTS infrastructure is used efficiently it can provide a holistic understanding of infrastructure and services running on top of it. Telemetry makes it possible to achieve closed-loop automation, streamline root cause analysis, perform timely reactive maintenance, effectively plan for proactive maintenance, and much more.

The vCMTS and the platform telemetries are available through open and industry-standardized interfaces, so it can be used to feed a wide range of applications and workflows [3]. There are many telemetry data collection open-source software available that use various plugins to gather metrics from a variety of sources, including COTS servers and software applications like vCMTS. Some common metrics collection software widely used today are Collectd [4], Telegraf [5] and cAdvisor[6]. This telemetry can be integrated with various monitoring solutions that can help remediate platform and vCMTS-related issues quickly. These metrics collection daemon is typically run on non-dataplane central processing unit (CPU) cores so the metric collection processes doesn’t interfere with data plane performance. The metrics are collected periodically and stored in a time-series databases like Prometheus [7], InfluxdB [8], etc. These metrics from the hardware and the vCMTS application can be visualized on a single or multiple Grafana dashboards. Grafana alerts can be created in a single and consolidated view based on certain pre-set conditions, and application and platform metrics thresholds, that make headend management easier, enhance recovery time, and reduce service downtime significantly.

Like legacy CMTSs, vCMTSs can also have hardware failure from time to time. Since a vCMTS runs on commodity servers the common hardware failures are related to power supply, memory dual in-line memory modules (DIMMs), NIC port, disk corruption etc. One great benefit of virtualization is that you can have redundancy available at almost all levels with properly configured HW and closed-loop automation. Intel servers provide a great deal of telemetry from different server components and if monitored properly can indicate hardware failures beforehand. This lets cable operators plan maintenance proactively and isolate bad hardware to mitigate downtime significantly. The hardware telemetry can also be used to enhance scheduling decisions in K8s which enables intelligent placement of vCMTS SGs pods based on up-to-date

platform telemetry and vCMTS workload requirements. If one of the K8s nodes reports any bad hardware metrics, the K8s scheduler will automatically cordon off those nodes and de-schedule SGs from that node and move them to a healthy node.

One key consideration for cable operators is to use automation to manage their virtualized headend's network operations. Automation helps to manage growing and changing networks, fix problems faster, and helps adhere to customer service level agreements (SLAs). To perform automation effectively, it requires end-to-end monitoring of software, services, and the hardware on which these services are running within the network. Intel server telemetry spans a vast number of domains including utilization, power consumption, fault detection, and performance. To offer meaningful insights from this information, Intel has created a portfolio of telemetry reports that provides actionable data about the current status of the server [9]. Combining these insights with vCMTS performance data allows for a more holistic view of the vCMTS network function. These reports can be used to help automate orchestration, self-healing, and energy optimization. There are four telemetry reports currently being developed:

- Platform health; covers overall platform health covering compute, memory, storage, and network interfaces
- Utilization; covers platform utilization and capacity indicators
- Congestion; covers CPU overload or network congestion scenarios
- Configuration; covers platform misconfigurations

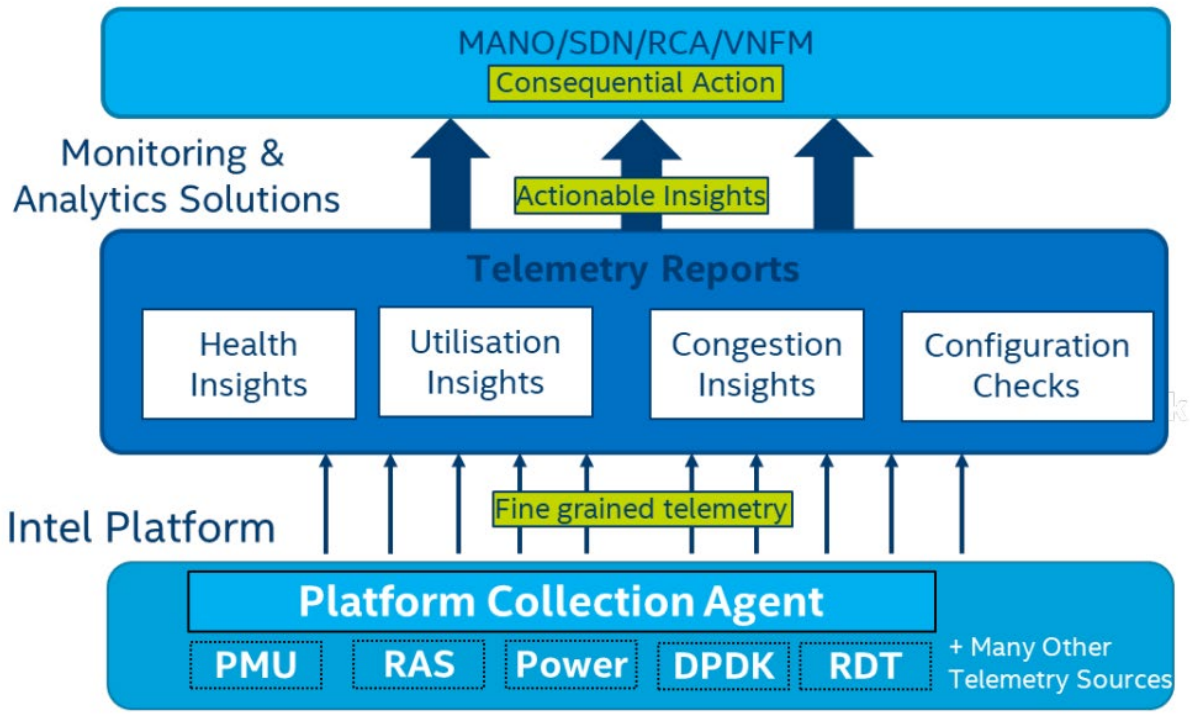

Figure 14 - Telemetry and insights data flow

Figure 14 shows how the telemetry reports use Intel server telemetry provided by Collectd and Telegraf via intel plugins to a monitoring solution that forms the input for the reports. The output

of the reports can be read by an operator or consumed by other management systems. The cable operators can feed these insights generated by the reports into their monitoring systems, which are then processed by online or offline automated systems and can also be used for visualization purposes. Figure 15 (below) illustrates a simple Grafana dashboard visualizing two of the memory health reports, availability and errored seconds. Errors can be documented in this instance and clear indications of these errors can be visualized. [9]

Figure 15 -Grafana dashboard visualizing two memory reports

In addition to the platform metrics in a vCMTS, Grafana dashboards can provide metrics to diagnose and examine subscriber issues. This can be seen in Figure 16 below from a cnBR show cable modem PHY dashboard. This displays the PHY output of the cable modems, just like running the command in CLI on a legacy CMTS.

cnBR Manager / SCM PHY

Last 30 minutes

cnBR Name: WWT-CNBR | cnBR ID: 10.254.154.173.nip.io

cnBR Cluster > Cable Modem List > SCM PHY

Download CSV

show cable modem phy (wideband cable modem)

MAC Address	I/F	USPwr	USSNR	Timing Offset	DSPwr	DSSNR
acdb.489e.f109	C5/0/U0	44.75	39.09	256	-3.80	46.10
acdb.489e.f109	C5/0/U1	32.25	42.04	255	-3.80	46.10
acdb.489e.f109	C5/0/U2	19.75	42.04	255	-3.80	46.10
acdb.489e.f109	C5/0/U3	7.25	42.04	256	-3.80	46.10
acdb.489e.f109	C5/0/U4	58.75	42.04	256	-3.80	46.10
acdb.489e.f109	C5/0/U5	58.75	39.09	1006	-3.80	46.10
acdb.489e.f109	C5/0/U6	46.25	42.04	256	-3.80	46.10
acdb.489e.f109	C5/0/U7	33.75	42.04	256	-3.80	46.10
acdb.489e.f83d	C4/0/U0	51.00	42.04	1389	-3.00	43.10

Figure 16 - Show Cable Modem PHY dashboard

Using the metrics collected by the vCMTS software, troubleshooting issues follows the same paths as legacy CMTS systems. The information is provided from the various containers and displaying them in easily navigable outputs. Subscriber impairments can be investigated by examining an individual subscriber and the reported information from the modem. The way this information is presented can vary by vendor, with the subscriber information in Figure 19 being displayed by Harmonic's CableOS, while Figure 20 displaying similar information as reported by Cisco's cnBR. While these software packages come with default dashboards, since these are using Grafana, the operator is able to modify, or completely create new dashboards as needed.

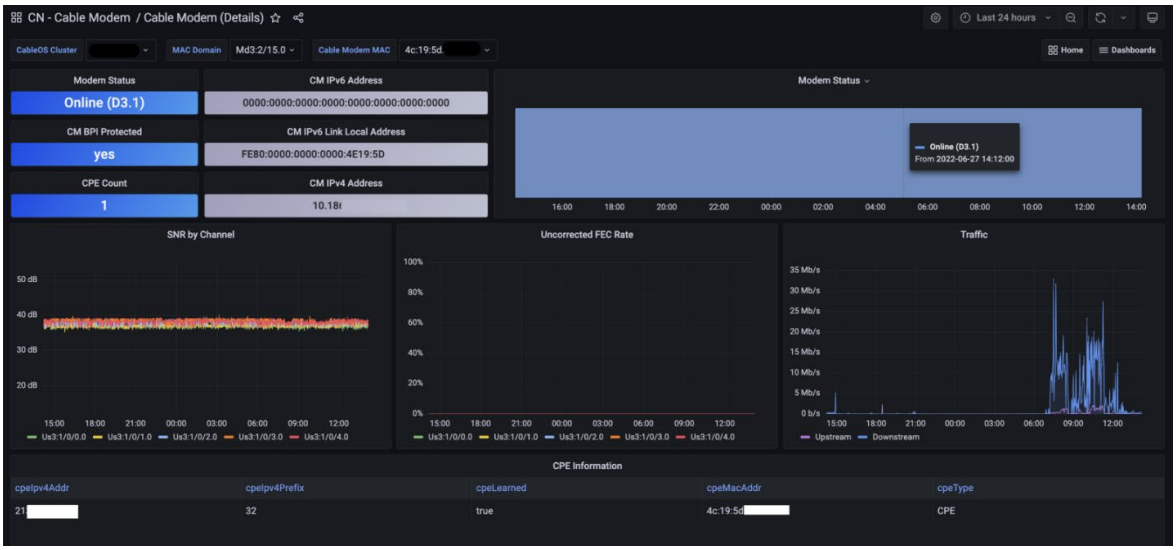

Figure 19 - CableOS dashboard displaying modem detail

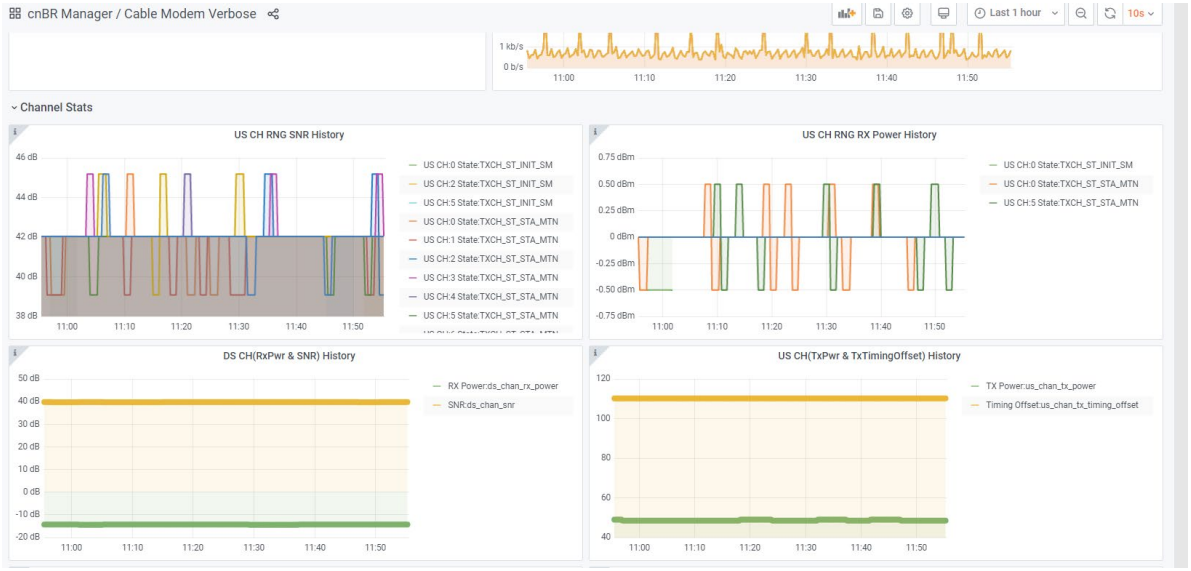

Figure 20 - cnBR dashboard displaying modem detail

A vCMTS ultimately provides the same information that a legacy CMTS provides. As such when diagnosing DOCSIS issues the troubleshooting techniques likewise follow the same steps that are performed in section 6.1. The difference being that the information is provided in an easier to digest form using GUI information, and not requiring SNMP or CLI intervention to obtain the data, thus simplifying operations. However, new telemetry-based tools, such as a Telegraf, InfluxDB and Grafana (TIG) stack, along with a new set of skills and training may be required to take advantage of the improved reporting features available in the vCMTS.

7. How do I Multi-task with a vCMTS?

The disaggregation of the PHY layer-1, MAC layer-2, and IP layer-3 of the CMTS makes it possible for cable operators to scale their access networks - virtually infinitely. As we have seen with the adoption and deployments of DAA, cable operators are reaping the benefits of running the DOCSIS PHY in the field by deploying RPDs. Upstream RF signals are demodulated at the RPD. This means RF signals are transported back digitally to the CMTS. Combining RF signals in racks of RF-combining equipment is no longer required. This has benefits for cable operators looking to reclaim headend space, reduce power and cooling costs in the headend.

The standardization of DAA for DOCSIS and further advancements in the flexible MAC architecture (FMA) standard have enabled the transition to a software-centric cable network infrastructure. [2] One option that the DAA and FMA architectures allows for the DOCSIS MAC software to be deployed as a virtual network function (VNF) on general purpose x86 servers in a cable operator headend as a vCMTS, while the DOCSIS PHY layer is housed in an HFC node near subscribers and business customers.

7.1. What is required to run the DOCSIS workload in software on a general purpose x86 server?

In Figure 17 below, we illustrate the typical components that make up a virtual CMTS deployment. There are hardware and software elements needed to deliver the DOCSIS workload to cable modem users.

The **hardware** required is COTS hardware that is typically found in a data center or server room of companies across all sectors of today's information age such as telecom, media & entertainment, manufacturing, medical & health services, automotive, and financial services. The servers are sold by original equipment manufacturers (OEM) such as HPE, Dell, Cisco, Lenovo, SuperMicro, Intel, and others. The NIC are made by various OEM suppliers such as Intel, Mellanox, Broadcom, Marvell, Cisco, and others. The data center switches are made by OEM vendors such as Arista, Cisco, Dell, Extreme, HPE, Juniper, and others. The hardware comes in various sizes and configurations to meet the compute and networking requirements to serve the workload in that location of the network topology. The benefit of running the DOCSIS workload in software on general purpose servers and switches is the economies-of-scale.

The **software** elements listed in the Figure 17 below vary depending on the vCMTS software selected by the cable operator. Typically, different software is needed for the OS, the container orchestration, management, automation, monitoring, analytics, and the vCMTS VNF. Some of the software to run the server platform and container network function is available in opensource and has been proven to scale to billions of end-users.

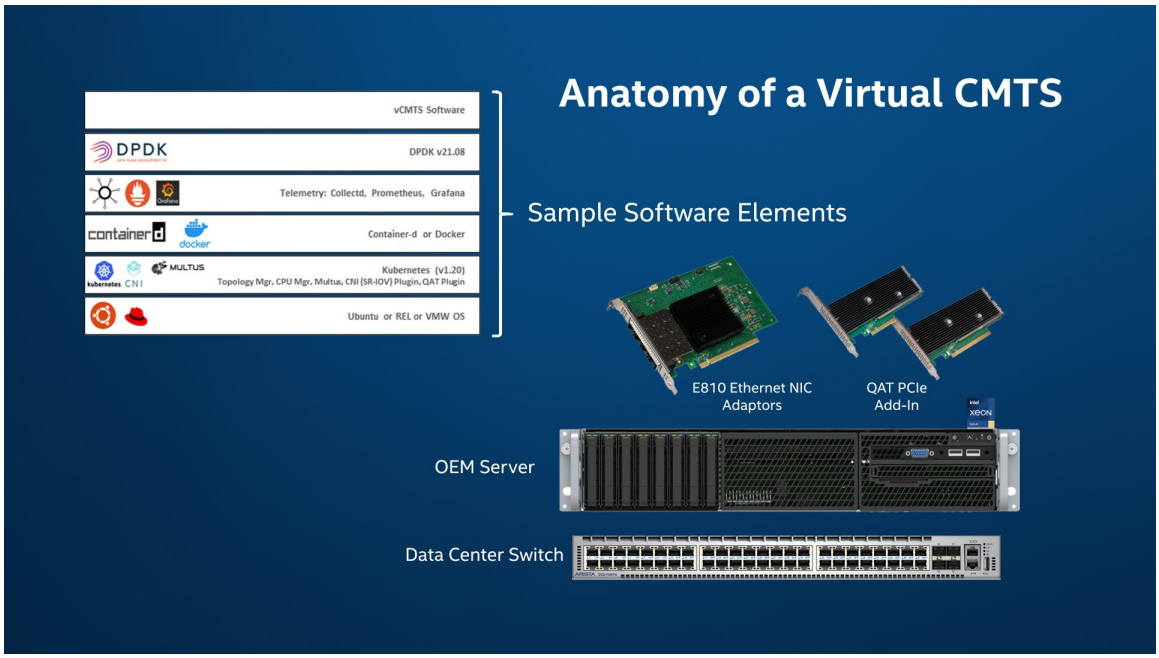

Figure 17 - Anatomy of a Virtual CMTS deployment

The advent of SDN and NFV has accelerated innovation for the separation of the data plane function, the control plane function, and the management functions in the delivery of networking applications. Containerization allows further disaggregation of compute, networking, and storage resources needed to move, process, and store packets and data at a much more granular level than previously possible with purpose-built, fixed-function, hardware. An example of this is a CMTS that is built to serve the function of modulating and de-modulating DOCSIS encapsulated packets, processing the packets, and moving them between end-users and the Internet. At the turn of the century, the legacy CMTS platforms were built with shared compute resources to service pre-defined network segments such as an upstream port or a downstream port. In the new world of container networking running on commodity servers, more control and freedom is given to the cable operator to allocate (i.e. orchestrate) compute and networking resources required to service the customer-defined segments. The cable operator can allocate the compute resources down to the granular level of threads in the CPU cores, and the amount of IO on the NIC logical ports at the service group, the upstream, and the downstream level to maximize resource allocation needed per service element. Not all service groups are the same, therefore, the compute and IO resources to serve them can be customizable.

Servers can be purchased as single socket or dual socket systems. Meaning, the server can be configured to have 1 CPU or 2 CPUs in the same system that houses the power supplies, fans, memory, and NIC cards. Each CPU will have multiple CPU cores. A CPU core is a virtual CPU that is a self-contained unit of compute within a CPU. Each CPU core can perform different tasks. Today's servers have CPU sockets, CPU cores and NIC interfaces that do the compute processing and network movements of the data packets. The vCMTS software solution typically has default allocations for CPU sockets, CPU core, threads, and NIC partitions that are pre-defined for ease of operation of the DOCSIS function that cable operators adopt "out of the box."

If desired, an operator could potentially orchestrate the container functions and land a DOCSIS workload on one of the two CPUs in a dual-socket server that services 10 DOCSIS service groups in a small town, while orchestrating the other CPU socket to run a cloud Digital Video Recorder (cDVR) function with some cloud storage resources. Even more granular, an operator may choose to mix the CPU cores in the CPU socket to do different functions. For example, if the operator was running 2 service groups with a few of the CPU cores and decided that there are CPU cores available for running the virtual optical line terminal (OLT) function of the passive optical network (PON) network, or, for terminating the Soft Generic Routing Encapsulation (Soft-GRE) tunnels for the community WiFi service, or for running another service, the operator can do so with flexibility. The x86 CPU cores and NIC cards and ports, coupled with container networking software allows operators much more control, flexibility, and optionality to maximize usage of the computing resources. This flexibility to allocate compute, IO, and storage resources at a much more granular level allows cable operators to scale their networks much more cost-effectively while converging multiple services on commercially available servers.

8. Does a vCMTS Consume More or Less Power Than a Legacy CMTS?

When migrating from legacy CMTS to vCMTS, cable operators will realize significant reductions in power consumption. Depending on the configuration of the vCMTS server clusters, Cable operators will see reductions in power that are 3 to 5 times less than legacy CMTS. Furthermore, coupled with the reduction in vCMTS power consumption and the associated heat being generated, cable operators will see significant reductions in power consumption of the heating, ventilation and air conditioning (HVAC) cooling systems. The reduction in rack space to house vCMTS servers to service the same, or more, service groups will also result in significant cost reductions.

When comparing power consumption of legacy CMTS versus vCMTS, the units of measurements will be normalized down to a cable service group. Below is a chart comparing a legacy CMTS operating in I-CCAP mode, a legacy CMTS performing the DOCSIS MAC functions connected to a remote PHY device, and a vCMTS doing the DOCSIS MAC processing connected to a remote PHY device.

On a per service group per watt comparison, without including the HVAC cooling reductions, a vCMTS uses 2.5 times less power than a distributed converged cable access platform (D-CCAP) and 3.8 times less power than an integrated converged cable access platform (I-CCAP). When comparing the amount of DOCSIS throughput delivered per megabit per second (Mbps) per watt, a vCMTS delivers nearly 10 times as much bandwidth as an I-CCAP.

There are additional power conservation possibilities when running vCMTS software on Intel servers. One example that cable operators can employ is to turn down the CPU clock frequency during off-peak network periods to reduce the power consumption of the vCMTS servers.

Table 2 - Power consumption comparison

	Scenario 1	Scenario 2	Scenario 3
	I-CCAP CBR-8	D-CCAP CBR-8	vCMTS 3 servers + switch + timing server
Power consumed	5,200 watts	5,000 watts	2,635 watts
Service Groups served	56	84	108
Watts per Service Group	92.8	59.5	24.4
Rack space	14 RU	14 RU	8 RU
Aggregate Throughput	100 Gbps	100 Gbps	480 Gbps
Mbps per watt	19.23	20.0	182.2

Notes:

- For this comparison, speed of 3 Gbps downstream and 200 Mbps upstream configurations for each service group were used.
- The RPD power consumption is not included as it will be the same for scenarios 2 and 3. Scenario 1 would have multiple racks of required RF cable combining equipment that are not included in this power consumption analysis of the DOCSIS MAC processing.
- The vCMTS configuration in scenario 3 includes three (3) dual-socket servers powered by Intel 3rd Generation Scalable Processor CPUs, one (1) Cisco Nexus 3232c leaf switch, and one (1) timing server. Each server consumed 635 watts while producing 160 Gbps of aggregate active throughput and powering 36 active service groups and 12 standby service groups.
- Aggregate throughput of I-CCAP and D-CCAP are limited to 100 Gbps due to physical limitations of the I/O cards. Similarly, the aggregate throughput limit of the vCMTS server is the NIC capacity, which is typically 100 Gbps, 200 Gbps, or 400 Gbps per server, as procured today.

9. Conclusion

In conclusion, we hope the readers will appreciate that operating a vCMTS is not much different than operating a legacy CMTS. Some would argue that the web interfaces for managing a vCMTS are easier to navigate than the CLI of legacy CMTSs, similar to the transition from MS DOS to Windows in 1995, which made using a personal computer (PC) much easier.

As the CMTS workload is performed in software, cable operators can run the vCMTS software on commodity servers. These general-purpose servers are used for running a plethora of applications such as DNS, email, video streaming, web services, chat, user authentication services, firewalls, security, billing, monitoring, video encoding, video transcoding, and so many other applications. Thus, millions of servers are purchased and deployed annually by companies, big and small, to run their business services. By running the DOCSIS workload on commodity servers and switches, cable operators can buy the hardware at a fraction of the capital cost of purpose-built hardware. Furthermore, if no longer needed for running DOCSIS, cable operators can repurpose the commodity servers to run other functions such as email, video streaming,

proxy caching, or SD-WAN, or any other software application on the same CPU cores in the servers located at that location in the network.

CMTS hardware has served cable operators well to win market share in the burgeoning broadband market over the past 25 years. Compared to legacy CMTS equipment, vCMTS software running on servers can do the job at a fraction of the hardware cost, consume a fraction of the rack space, use a fraction of the power, and allow cable operators more flexibility to quickly scale and remain competitive.

Abbreviations

API	application programming interface
bps	bits per second
BIOS	basic input output system
BGP	border gateway protocol
BMC	baseboard management controller
CDVR	cloud digital video recorder
CLI	command line interface
CM	cable modem
CMTS	cable modem termination system
CNF	container network function
COTS	commercial off-the-shelf
CPU	central processing unit
DAA	distributed access architecture
DHCP	dynamic host configuration protocol
DIMM	dual in-line memory module
DOCSIS	Data-Over-Cable Service Interface Specification
DPDK	data plane development kit
FEC	forward error correction
FMA	flexible MAC architecture
GHz	gigahertz
GUI	graphical user interface
HD	high definition
HE	headend
HFC	hybrid fiber coax
HVAC	heating ventilation and air conditioning
Hz	hertz
IO	input-output
ISV	independent software vendor
JSON	JavaScript object notation
K8s	Kubernetes
KPI	key performance indicator
MAC	media access control
MANO	management and orchestration
MIB	management information base
MSO	multiple system operation
NE	network element
NFV	network function virtualization
NFVO	network functions virtualization orchestration
NIC	network interface card
OEM	original equipment manufacturer
OFDM	orthogonal frequency-division multiplexing
OFDMA	orthogonal frequency-division multiple access
OLT	optical line terminal
OS	operating system

PHY	physical layer
PMU	performance monitoring units
PON	passive optical network
PTP	precision timing protocol
PXE	preboot execution environment
QAM	quadrature amplitude modulation
RAS	reliability, availability, serviceability
RCA	root cause analysis
RDT	Resource Director Technology
RF	Radio Frequency
RMD	Remote MACPHY Device
RPD	Remote PHY Device
R-PHY	Remote PHY
RU	Rack Unit
S-CDMA	Synchronous Code Division Multiple Access
SC-QAM	Single Carrier QAM
SCTE	Society of Cable Telecommunications Engineers
SDN	Software Defined Networking
SGs	Service Groups
SLA	Service Level Agreement
SNMP	simple network management protocol
SoftGRE	soft generic routing encapsulation
SR-IOV	Single Root I/O Virtualization
SSH	Secure Shell
TDMA	Time Division Multiple Access
TFTP	Trivial File Transfer Protocol
TIG	Telegraf, InfluxDB, and Grafana
US	upstream
vCMTS	Virtual CMTS
VNF	Virtual Network Function
VNFM	virtual network functions manager
YAML	yet another markup language

Bibliography & References

[1] DOCSIS 4.0 - A Key Ingredient of the 2030's Broadband Pie, A Technical Paper prepared for SCTE by Maricevic, Andis, Cloonan, Ulm, 2021

[2], B. Ryan, M. O'Hanlon, D. Coyle, R. Sexton and S. Ravisundar, “Maximizing vCMTS Data Plane Performance with 3rd Gen Intel® Xeon® Scalable Processor Architecture,” [Online]. Available: https://networkbuilders.intel.com/solutionslibrary/maximizing-vcmts-data-plane-performance-with-3rd-gen-intel-xeon-scalable-processor-architecture

[3] https://www.intel.com/content/dam/www/public/us/en/documents/guides/nt-telemetry-eguide-for-web-team.pdf

[4] https://collectd.org/

[5] https://www.influxdata.com/time-series-platform/telegraf/

[6] https://prometheus.io/docs/guides/cadvisor/

[7] https://prometheus.io/

[8] https://www.influxdata.com/

[9] https://builders.intel.com/docs/networkbuilders/telemetry-reporting-for-network-infrastructure-solution-brief.pdf