Real-World Deployment of a Virtual Cable Hub

A Technical Paper prepared for SCTE/ISBE by

Asaf Matatyaou
Vice President, Solutions and Product Management, Cable Edge Business
Harmonic Inc.
4300 North First Street
San Jose, CA 95134
408-490-6834
asaf.matatyaou@harmonicinc.com

Table of Contents

List of Figures

List of Tables

1. Introduction

The promise and potential of virtualizing a cable hub has been discussed over the past few years. While the opportunities are limitless, there must be a starting point and manageable steps for operators in migrating to virtualization.

This paper will focus on translating the theory of virtualization into real-world experiences and recommendations based on deployment. Different aspects of a cable hub will be evaluated for virtualization, including using cloud for telemetry and monitoring, configuration and orchestration, operations and back-office elements, as well as a software-based CCAP solution based on a virtual CMTS (vCMTS) implementation. Virtualization implementation approaches will be compared, such as containerization and virtual machines (VM) for different applications and services. In addition to operationalization considerations, support for maintaining legacy services, such as traditional broadcast video, VOD and out-of-band, as well as existing and future IP services will be considered in the transition to a virtual cable hub.

2. The Promise of Virtualization

The promise of virtualization is not new. In fact, it's delivered substantial benefits to many industries, such as data centers, wireless access and Internet of Things (IoT), and has delivered on this promise for decades. The definition of virtualization has broadened over these decades and practical use cases and real-world deployments continue to increase every year.

Virtualization promises to enable change at a pace which meets or exceeds customer demand in the most effective manner. Change can be defined in many ways, such as new services, additional security, elastic storage, more efficient infrastructure and in the case of the cable broadband industry it's all about speed. Virtualization delivers on this promise by separating applications or software from hardware. Key benefits are scalability, sustainability and elastic deployment with the quickness and agility needed to increase business efficiencies and productivity. This separation of software and hardware is the key to quickness and agility of change, as the software can be changed while running on deployed hardware. Upgrading software is not only quicker than replacing hardware, but requires less operational expenditure (OpEx), such as onsite labor and increase in power consumption requirements.

With all the benefits mentioned, why is the cable hub lagging behind in embracing virtualization? In the technical paper, *Transforming the HFC Access Network with a Software-based CCAP*, prepared for SCTE in October, 2015, the considerations for virtualizing the CMTS capabilities in a cable hub were described. Notably, key enablers identified are:

1. CableLabs Remote PHY standard: A standards-based approach separates the physical layer (PHY) from the CMTS Core, which "contains the DOCSIS MAC and the upper layer DOCSIS protocols. This includes all signaling functions, downstream and upstream bandwidth scheduling, and DOCSIS framing."[1]
2. Merchant silicon for a full spectrum DOCSIS 3.1 PHY layer: Remote PHY hardware will deliver on a long lifespan, with up to 10 Gbps[2] of downstream and up to 2 Gbps of upstream bandwidth.

---

[1] Remote PHY Specification, CM-SP-R-PHY-I07-170524, §5.1, pg. 25

[2] MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I11-1705, §5.1, pg. 51

3. Performance from Commercial Off-The-Shelf (COTS) Intel-based x86 servers: The performance delivered by today's Intel CPUs exceeds minimum product requirements to serve customers' needs and has the history and potential of continued annual performance gains.

These three key enablers are recent technological advances. Coupled with existing virtualization technologies, these key enablers and associated benefits are now ripe for cable edge deployments.

3. Today's Cable Hub

Today's cable hub is a combination of many legacy and purpose-built hardware-based solutions, which over the past 20 years (DOCSIS turned 20 years in March 2017[3]) has delivered on data, voice and legacy (MPEG-based) video services. To deliver these fundamental services (other services, such as home security and commercial services, are typically running over the DOCSIS data service as the fundamental service), active equipment found in a typical cable hub includes:

1. Cable Modem Termination Systems (CMTS)
2. EdgeQAMs
3. Routers and switches
4. Out-of-band modulators and return path demodulators
5. FCC and LTE leakage signal generation
6. Provisioning servers (such as DHCP and TFTP)
7. Configuration tools
8. Monitoring tools
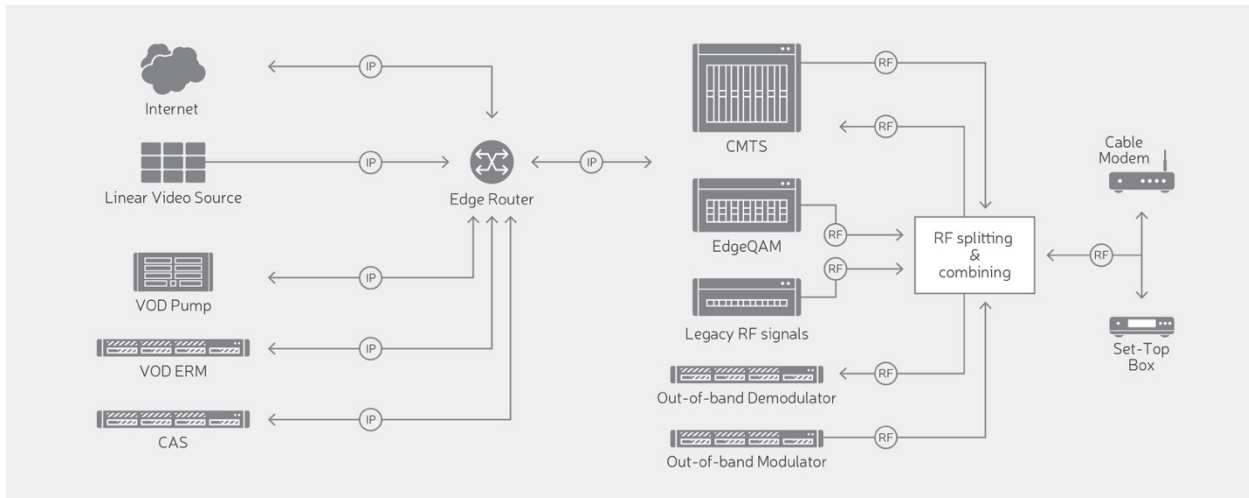
Figure 1 - Today's Cable Hub

Most of the equipment listed above can be categorized as either custom hardware (e.g. CMTS & EdgeQAM) or custom software (e.g. configuration and management tools) and most are purpose-built for the cable edge (with the exception of routers, switches and some of the provisioning server components). The challenges in continuing with a customized cable edge and hardware-based approach as compared to

---

[3] DOCSIS 1.0 specifications Version I01 were released March 26, 1997.

a virtualization approach, as described in *Transforming the HFC Access Network with a Software-based CCAP*, are shown in Table 1 - Virtualization Opportunities in a Cable Hub.

Table 1 - Virtualization Opportunities in a Cable Hub

Hardware-based Challenges	Virtualization Opportunities
Unsustainable space, power and cooling	Dramatically reduced space, power and cooling footprint
Proprietary and custom hardware, with long development cycles	More frequent and shorter development cycles
Complex and infrequent software upgradeability results in slow feature introduction	Agile development with frequent software upgrades delivers fast feature velocity
Limited capacity, scalability and flexibility	Sustainable capacity growth, elastic scalability and increased flexibility
Larger failure domain	Smaller failure domains
High operational expenditure (OpEx)	Improved Total Cost of Ownership (TCO), including reduced operational (OpEx) and capital expenditure (CapEx)

Today's cable hub is most challenged by the equipment which doesn't sustainably scale physically or operationally with service or capacity growth. Sustainable growth will happen when the performance and scale meets or exceeds consumer consumption demands, either driven by actual usage or competition from other access providers, such as Fiber To The Home (FTTH). In other words, virtualizing the cable hub is a high-tech solution for a low-tech problem, specifically, running out of facility space, as well as ever-increasing and recurring electricity expenses.

Another consideration for the ever-evolving cable hub is Remote PHY. While this standard technology enables virtualization in many ways, it also demands many more nodes (or service groups) being deployed, with some estimates exceeding a ten-fold increase in nodes. Today's deployment is manual and labor intensive and doesn't scale operationally, especially when considering the desired deployment rate of Remote PHY nodes. Today's cable hub tools include tried-and-true and very familiar tools such as Command Line Interface (CLI), Simple Network Management Protocol (SNMP) and Internet Protocol Detail Record (IPDR). In many cases, individual cable operators have customized home-grown tools, interfacing to hardware-based equipment over standard protocols (SNMP, IPDR) or proprietary CLI. These configuration and management interfaces are also archaic, slow and manual in many cases.

The equipment and tools which benefit the most when transitioning to a sustainable growth deployment model will be highlighted as components in the cable hub which have been virtualized in a real-world deployment of a virtual cable hub.

4. Virtualization in All Shapes and Sizes

While virtualization is a relatively recent hot topic in the cable industry, it has been present for decades. Since the 1960's the definition of virtualization has broadened over the years. This is good. The power of virtualizing continues to expand, and its reach has delivered benefits to many industries and different types of implementation.

Wikipedia defines virtualization as "the act of creating a virtual (rather than actual) version of something, including virtual computer hardware platforms, storage devices, and computer network resources." Further, Wikipedia lists many different types of virtualization. To name a few: desktop, operating system-level, application, workspace, service, memory, storage, file system, data, database and network.[4]

When looking at how virtualization applies to the cable edge space, particularly in the cable hub, equipment and tools that are custom hardware-based or benefit from orchestration (for the purpose of eliminating labor intensive human interaction to operate and manage services) will be considered. To be specific, there is a disruptive change coming to cable hubs which is described in Table 2 - Virtualization Changes in the Cable Hub.

Table 2 - Virtualization Changes in the Cable Hub

Old Way	New Way
Application runs on custom hardware	Application runs on COTS hardware
Application is implemented partially or fully on hardware components (e.g. ASIC, FPGA)	Application is implemented in software
Replaced every three to five years	Long lifespan
Application is upgraded infrequently	Application is upgraded regularly
Equipment location is limited	Equipment location is varied
Services turned on are labor intensive	Services are turned on automatically with orchestration
Service monitoring is labor intensive	Service monitoring (telemetry) is performed by software analytics
Service events are limited to local hardware storage	Service events are stored in the cloud

To summarize, virtualization in a virtual cable hub means running virtual applications (for example, DOCSIS, video, OOB) on COTS x86 platforms, which can be located in cable hubs, more centrally in fewer locations (such as data centers) or even in smaller form factors in distributed locations (such as street cabinets).

In this virtual cable hub, telemetry and logging tools perform analysis of streaming data predicting potential impactful events or visualizing historical events in a holistic fashion. In many ways, the idiom "a picture is worth a thousand words" becomes reality. Instead of hours of labor intensive human scrutiny, which may result in a partial analysis and slower resolution of a field issue, a glance of dashboard provides an instantly clear picture showcasing visualized streaming data. The benefit is speed and accuracy in determining a more complete resolution.

---

[4] https://en.wikipedia.org/wiki/Virtualization

5. Virtualization Considerations

Defining cable edge virtualization as running cable-specific virtual applications in software is a first step in the right direction. This identifies what elements or equipment will be virtualized. However, other virtualization concepts are important to consider, specifically how the software is virtualized. Common concepts include containerization, virtual machines and bare metal approaches. Some of these methods are mutually exclusive, while others are complementary or even dependent on each other. While the methods described are common in different industries, there isn't always definition consensus for each concept. This is due to the wide variety and growing usage of virtualization for different objectives. For this paper, the following definitions will be used for these concepts and are based on the actual usage of virtualizing a cable hub.

- Bare metal: an application is executing on the native operating system (OS), in comparison to executing on a virtual machine or a virtual OS layer. In other words, the application can access "the metal" directly or via a native OS.
- Virtual machine: System virtual machines are capable of virtualizing a full set of hardware resources, including a processor (or processors), memory and storage resources and peripheral devices. A virtual machine monitor (VMM, also called hypervisor) is the piece of software that provides the abstraction of a virtual machine.[5]
- Virtual appliance: "a pre-integrated, self-contained system that is made by combining a software application (e.g., server software) with just enough operating system for it to run optimally on industry standard hardware or a virtual machine (e.g., VMWare, VirtualBox, Xen HVM, KVM)."[6]
- Containerization: "applications can be broken up into manageable, functional components, packaged individually with all of their dependencies, and deployed on irregular architecture easily."[7]
- Docker: a set of tools to package and deploy containers, which can specify container constraints and access permissions. Additionally, Docker sets up and deploys the container in Linux.
- Cloud native: "cloud native computing uses … software … to be containerized, dynamically orchestrated, and microservices oriented."[8]
- Sandboxing: an isolated computing environment for running applications.
- Single-tenant: a single instance of a single application type running a single physical hardware platform.
- Multi-tenant: multiple instances of one or more application types running on one or more physical hardware platforms.
- Kubernetes: "an open-source system for automating deployment, scaling, and management of containerized applications."[9]
- Kubernetes-native application: an application which is aware that it's being deployed or managed by Kubernetes.

---

[5] https://en.wikipedia.org/wiki/Popek_and_Goldberg_virtualization_requirements

[6] https://www.turnkeylinux.org/virtual-appliance

[7] https://www.digitalocean.com/community/tutorials/the-docker-ecosystem-an-overview-of-containerization

[8] https://www.cncf.io/about/faq/

[9] https://kubernetes.io

- Microservice: "refers to an architectural approach that independent teams use to prioritize the continuous delivery of single-purpose services. The microservices model is the opposite of traditional monolithic software which consists of tightly integrated modules that ship infrequently and have to scale as a single unit."[10]

Clearly, there are many decisions to be made and tradeoffs to consider when evaluating which concept to use and when. The great news is that once custom hardware has been retired and applications are implemented in software on off-the-shelf hardware, there is no decision which can't be undone and no outcome which can't be upgraded or improved.

5.1. Virtualization Criteria in a Virtual Cable Hub

The key criteria for deciding between the various virtualization approaches are:

1. Time to market (TTM): the time criticality to deliver a minimum feature set of a virtualized set of applications. Over time, the feature set will grow with periodic software upgrades to the virtual applications.
2. Performance: the minimum application processing required to deliver a cost-effective footprint of COTS x86 servers. Over time, the performance will improve with periodic software upgrades to the virtual applications and the performance per rack unit will increase or the number of rack units will diminish to deliver the same performance.
3. Scale of deployment: the minimum quantity of consumers supported by a virtual cable hub. Over time, the scale of deployment per rack unit will increase or the number of rack units will diminish to support the same quantity of consumers.
4. Application flexibility: the minimum set of application types and elasticity to execute different instances of different applications on a single physical server. Over time, the ability to execute many and different instances of different application types on a variable set of physical servers will be possible with periodic software upgrades.

The common theme for all four criteria (TTM, performance, scale and application flexibility) is that there is a minimum or "good enough" starting point and that future software upgrades improve the virtual cable hub capabilities in different dimensions.

5.2. Crawl, Walk, Run, Fly Approach to a Virtual Cable Hub

The most common assumption when discussing virtualizing a cable hub is the notion that using virtual machines (VMs) is required. It's a possibility, but not a necessity. Let's compare bare metal, containerization with Docker and virtual machine approaches to a virtual cable hub, taking into account the four key virtualization criteria.

Two possible approaches include:

1. Single-tenant application running on bare metal or a virtual machine
2. Multi-tenant containerized application instances packaged and deployed by Docker on bare metal or virtual machine

---

[10] https://pivotal.io/microservices

5.2.1. Hit the ground running

The most complex and performance-intensive application in a virtual cable hub is the virtual CMTS (vCMTS) component. As defined by the CableLabs Remote PHY standard, the DOCSIS physical layer is separated from all the upper layers, via the standards-based protocols DEPI, UEPI and GCP.[11] When referring to the vCMTS component in the context of virtualization, the CMTS Core functionality (as defined in the CableLabs Remote PHY standards) is implemented as a virtual application.

An incremental approach when virtualizing the cable hub is a crawl, walk, run, fly approach[12]. With a virtualized approach rooted in software, maturing from crawling to flying is entirely performed by software upgrades along the way, and the penalty of tripping over oneself is limited to a software release iteration with no need to replace hardware.

One metric of vCMTS performance is the packet processing rate, which in turn results in the bandwidth or throughput capabilities of a vCMTS. Meeting real-world performance requirements dictates that a single-tenant vCMTS application running on bare metal has the quickest TTM, while still delivering on many virtualization benefits. Effectively, this is a virtual appliance approach, which can grow and scale accordingly by adding more servers, each running a single instance of a vCMTS virtual application. Concluding quickly on whether the simplest virtualization approach meets the performance and scale requirements to deliver on the stated benefits of virtualization while meeting or exceeding functional requirements of traditional hardware-based CMTS approaches is vital.

In practice, running a single instance of a vCMTS application on x86 COTS servers delivers tens of Gbps of packet processing performance per x86 server rack unit to dozens of service groups, while reducing the space, power and cooling footprint by up to 90% relative to existing hardware-based integrated CMTSes.[13]

The result of this first phase of a virtual cable hub is already delivering substantial capital and operating expenditure benefits to cable operators, and it can be debated as to whether the benefits already justify stating that this approach is "running."

5.2.2. How fast do you want to fly?

Continuing with the crawl, walk, run, fly analogy, let's shift gears and see how fast a virtual cable hub can fly. The next set of critical benefits to a cable operator when looking at a cable hub are still covered by the four criteria (TTM, performance, scale and application flexibility):

1. TTM: the speed to turn on consumer services to a single set of consumers
2. Performance: improving uptime by limiting the scope of service outages
3. Scale: the quantity of consumers which can be supported in a given footprint
4. Application flexibility: the set of virtual applications which are required for a virtual cable hub

The approach of multi-tenant containerized application instances packaged and deployed by Docker on bare metal provides many of these benefits. Let's identify which virtualization concept delivers on these benefits.

---

[11] Modular Headend Architecture v2 Technical Report, CM-TR-MHAv2-V01-150615

[12] Paraphrasing Martin Luther King Jr from his April 26, 1967 speech in Cleveland

[13] Based on Harmonic Inc's analysis of real-world case studies

Kubernetes provides the orchestration to deploy pods (a group of one or more containers), in conjunction with Docker as the tool to package the pods. "A pod models an application-specific 'logical host' - it contains one or more application containers which are relatively tightly coupled — in a pre-container world, they would have executed on the same physical or virtual machine. While Kubernetes supports more container runtimes than just Docker, Docker is the most commonly known runtime, and it helps to describe pods in Docker terms. Pods serve as unit of deployment, horizontal scaling, and replication."[14]

The speed to turn on consumer services is increased significantly when using a combination of Kubernetes and a Docker approach to pods deployment. This speed increase is gained when shifting from a human interaction to configure manually each unit of consumer deployment to an orchestrated and automated process. Additionally, reliability increases as the more error-prone manual method of configuring new consumer services is reduced or eliminated.

Improving uptime is a function of reducing failure domain size, which is a major benefit of a containerized approach. Determining the failure domain size of each pod or container provides the knob when determining the tradeoff between potentially more shared resources in a single pod (increasing CPU utilization percentage) and limiting the pod to a single consumer group, such as a service group. Software failures will happen and uptime is improved as failure domains are reduced. By any measure, the performance metrics of uptime percentage and the number of service calls received over a period of time is tracked by cable operators.
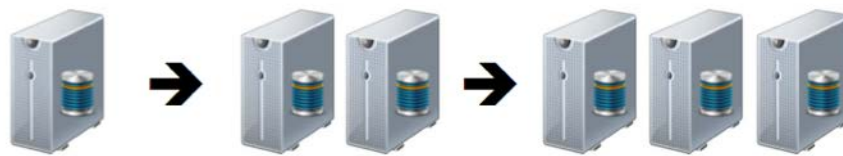
Containerization also has the benefit of horizontal scaling (scale-out) in comparison to vertical scaling (scale-up). Vertical scaling increases a single application instance's set of specifications, such as bandwidth or subscriber count, by increasing the number of x86 CPU cores, storage or NIC speed. A virtual cable hub uses vertical scaling to grow capacity when a single application instance can do more. Horizontal scaling increases a virtual cable hub's scale with a virtual set of application instances (pods), each with a specific and purposefully bounded scale specification. When the limitation of a single pod is reached, more pods are deployed.

---

[14] https://kubernetes.io/docs/concepts/workloads/pods/pod/

Figure 2 - Vertical Scaling

Figure 3 - Horizontal Scaling

Figure 2 and Figure 3 visualize the difference in approaches between vertical and horizontal scaling. The benefits of horizontal scaling through containerization becomes apparent: smaller failure domain, simpler to develop, simpler to test and more cost-effective COTS x86 server hardware requirements. Development and testing simplicity is explained when considering smaller data sets and test case parameters, such as developing to and testing to a scale of 200 as compared to 200,000 subscribers.

Lastly, multi-tenant in the context of a virtual cable hub, is having multiple instances of a single or multiple application types. Each pod of containers may be a different application type, and potential applications include CMTS Core, Out-of-band Core, Video Core, proactive network maintenance (PNM), orchestration and telemetry. Over time, different and new applications will be considered for virtualization. The benefit is leveraging the same COTS x86 servers for different application types, which delivers improved sparing (for hardware failures and replacement) and economy of scale benefits when procuring many of the same platform. When different applications are able to execute on the same COTS x86 server, dynamic and elastic utilization of CPU resources can be determined during run-time, based on usage metrics and analytics to shift resources from lightly loaded applications to applications running hot.

5.2.3. Microservices

The combination of the approaches described and benefits can be also referred to as microservices, which was previously defined as a "continuous delivery of single-purpose services."[15] It's evident that shifting from hardware-based to software-based implementations, as well as a shift in software development methodology from waterfall to agile, results in dramatically improved feature velocity. Shifting from traditional monolithic software to a microservices approach is another lever which improves feature velocity and delivers more frequent and higher quality software upgrades.

---

[15] https://pivotal.io/microservices

Monolithic software is released as a single unit, and due to size and complexity, has more software defects as well as a longer regression test cycle time. Microservices are lighter-weight modular units of software, which can be defined with a limited set of capabilities with published interfaces. With monolithic software, it's all or nothing. With microservices, each upgraded service software can be tested with lighter-weight automation and shorter regression test cycle time.

A notable benefit is improved software upgradeability, in terms of total time and service outage potential. Microservices, by definition, have smaller code size than a single monolithic software image. This results in less time to download software images, less time to upgrade software to a limited set of one or more microservices and reducing the minimal amount of code changes when correcting defective software. When performing hitless or in-service-software-upgrades (ISSU), redundancy or protection mechanisms are typically employed to activate a protected unit of software while the originally active software unit is upgraded without impacting service. However, during this time, overall system protection is diminished. With microservices, the total amount of unprotected time during ISSU is reduced, improving overall high-availability of the virtual cable hub.

6. The Virtual Cable Hub

Let's revisit the cable hub previously shown in Figure 1. The CMTS and EdgeQAM, which are historically implemented in big-iron hardware-based chassis, as well as legacy RF signal and out-of-band signal generation performed in the cable hub are shown as example applications which are virtualized as multi-tenant containerized applications turned on with automated orchestration on COTS x86 servers (shown in blue in Figure 4). In this future vision of a virtual cable hub, all data, video and voice services for residential and business customers are IP-based.
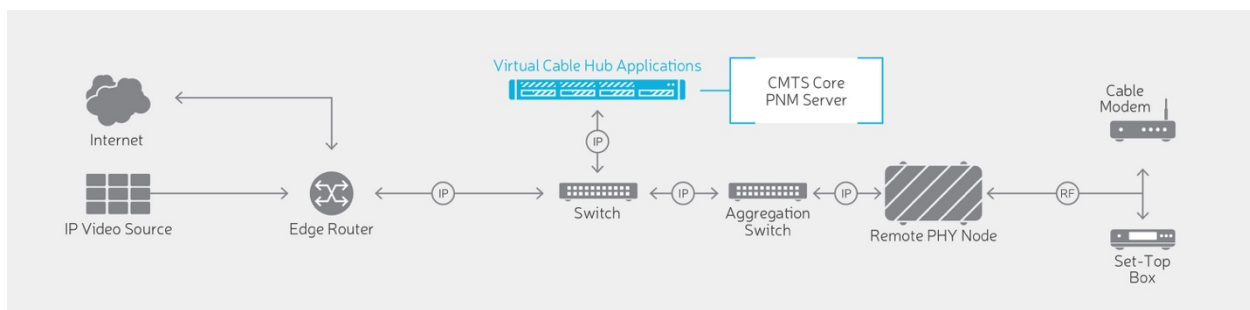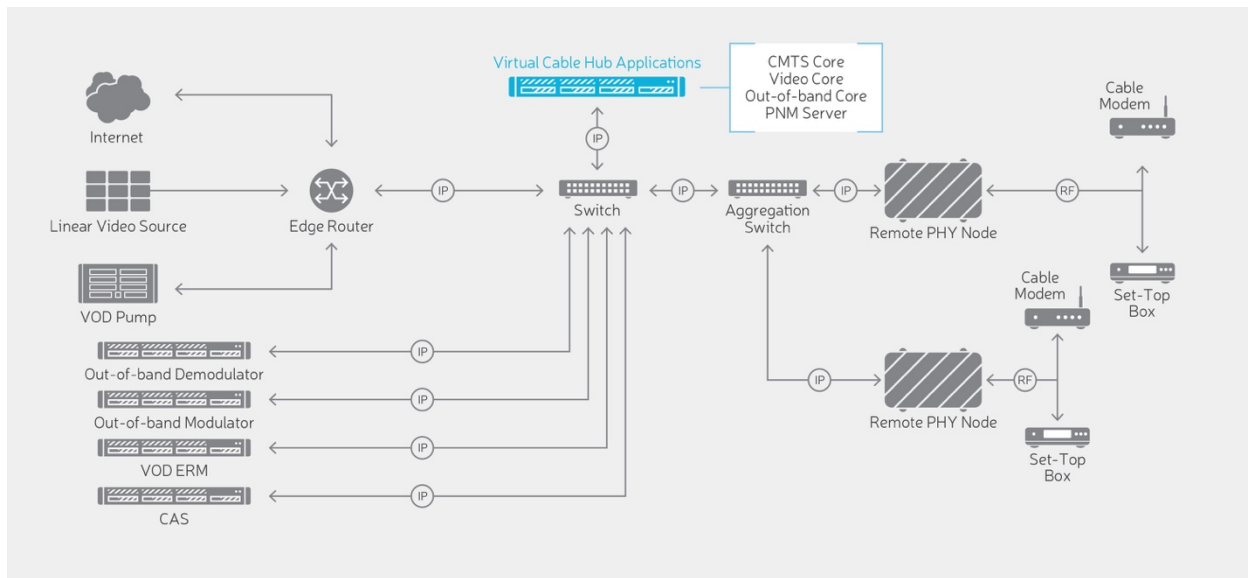

Figure 4 - Virtual Cable Hub

6.1. Moving Forward While Maintaining Legacy Services

The virtual cable hub depicted in Figure 4 is a bit too idealistic as a starting point, with the noted legacy services of linear/broadcast MPEG video, video-on-demand (VOD), switched digital video (SDV), as well as out-of-band signaling for set-top box (STB) and legacy RF signal generation such as HMS and FM. With millions of legacy consumer devices at subscribers' homes, these legacy services will be reduced over many years, eventually being replaced by pure IP-based services. Until that moment, the virtual cable hub, deployed in a DAA architecture such as Remote PHY, will need to support IP transport of the RF signals at the virtual cable hub, with the IP transport converted and modulated to RF signals at the Remote PHY node.

The applications to encapsulate the IP transport of the legacy RF signals don't require much processing and are scalable. With a virtual cable hub, instead of requiring single-purpose custom hardware-based solutions for each and every specialized legacy function, a virtual application can be deployed on available server resources in a cluster of servers to deliver the necessary capabilities. Figure 5 shows a virtual cable hub with legacy services supported.

Figure 5 - Virtual Cable Hub with Legacy Services

6.2. The Cloud for the Virtual Cable Hub

The virtual cable hub described in this paper can be considered to be "cloud native" as it meets the conditions defined: containerized, dynamically orchestrated and microservices oriented. However, "cloud native" can also refer to applications executing in the cloud (private/on-premises or public). An example of a public cloud service is Amazon Web Services (AWS), which claims "on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform."[16]

The virtual cable hub is advancing quickly in real-world deployments and delivering a full set of services, meeting demanding performance requirements for speed, latency and jitter. With vital residential and commercial services being delivered, it's important to intelligently weigh the tradeoffs if services might be hindered in a measurable way. Even in the early days of a virtual cable hub, some configuration and monitoring may be serviced in the cloud, public or private. In particular, logging and monitoring is well suited for the cloud, with on-demand increase in database storage and easy accessibility to telemetry and analytics.

Consider the limitations of a traditional hardware-based solution, with finite on-board storage for a small service area. Once the storage is exhausted, the older data is replaced with fresh data, which leads to less samples of data being stored or a short time span of data analytics or logging. Additionally, correlating data is a labor-intensive activity, with a person logging into each platform remotely. In a virtual cable

---

[16] https://aws.amazon.com/what-is-cloud-computing/

hub, a continuous stream of data is sent to the cloud, with machine analytics performed on a much larger service area. The cloud service expands, as necessary, to support longer time spans. Moreover, instead of data taken at a few instants in time, the data is nearly continuous and provides a complete picture of the virtual cable hub health.

7. Conclusion

As was predicted over the past few years, virtualization has arrived in the cable edge. The virtual cable hub will leverage the numerous and substantial virtualization benefits from other industries and apply these tried-and-true virtualization concepts for the first time. These concepts are the keys to unlocking the path to sustainably growing capacity, adapting quickly to customer demands, and a solution which is flexible and elastic enough to dynamically augment and shift resources to the most in-demand applications.

Abbreviations

AAA	Authentication, Authorization and Accounting
ASIC	Application-Specific Integrated Circuit
CapEx	Capital Expenditure
CCAP	Converged Cable Access Platform
CMTS	Cable Modem Termination System
COTS	Commercial Off-The-Shelf
CPE	Customer Premise Equipment
CPU	Central Processing Unit
DAA	Distributed Access Architecture
DEPI	Downstream External-PHY Interface
DOCSIS	Data Over Cable Service Interface Specification
FTTH	Fiber To The Home
FPGA	Field-Programmable Gate Array
Gbps	Gigabits Per Second
GCP	Generic Control Plane
HFC	Hybrid Fiber-Coaxial
HW	Hardware
I/O	Input/Output
IoT	Internet Of Things
ISSU	In Service Software upgrade
MAC	Media Access Control
NFV	Network Function Virtualization
NIC	Network Interface Controller
OOB	Out-of-band
OpEx	Operating Expenditure
OS	Operating System
PHY	Physical
PNM	Proactive Network Maintenance
RF	Radio Frequency
RU	Rack Unit
SCTE	Society of Cable Telecommunications Engineers
SDN	Software Defined Networking
SDV	Switched Digital Video
SLA	Service Level Agreement
STB	Set-Top Box
SW	Software
TCO	Total Cost of Ownership
TTM	Time To Market
UEPI	Upstream External-PHY Interface
vCMTS	Virtual CMTS
vCPE	Virtual CPE
VOD	Video On Demand
VPN	Virtual Private Network

Bibliography & References

Branson, Tony. "Database Scalability: Vertical Scaling vs Horizontal Scaling." *vcloudnews.com*. VCloud News, 5 Dec. 2016. Web. 31 July 2017. <http://www.vcloudnews.com/database-scalability-vertical-scaling-vs-horizontal-scaling/>

"Cloud Computing with Amazon Web Services." *aws.amazon.com*. Amazon Web Services, Inc, 2017. Web. 31 July 2017. <https://aws.amazon.com/what-is-aws/>

DOCSIS 3.1 MAC and Upper Layer Protocols Interface Specification, CM-SP-MULPIv3.1-I11-170510, May 10, 2017, Cable Television Laboratories, Inc.

Dwivedi, Vijay. "Container Orchestration Tools: Compare Kubernetes vs Mesos." *platform9.com*. Platform9, 11 Aug 2016. Web. 31 July 2017. <https://platform9.com/blog/compare-kubernetes-vs-mesos/>

Ellingwood, Justin. "The Docker Ecosystem: An Overview of Containerization." *digitalocean.com*. DigitalOcean Inc., 1 Feb. 2015. Web. 31 July 2017. <https://www.digitalocean.com/community/tutorials/the-docker-ecosystem-an-overview-of-containerization>

Luther King Jr., Martin.: April 26, 1967, Cleveland speech

Matatyaou, Asaf. Transforming the HFC Access Network with a Software-Based CCAP. Publication. San Jose: Harmonic, 2015. Web.

Modular Headend Architecture v2 Technical Report, CM-TR-MHAv2-V01-150615, June 15, 2015, Cable Television Laboratories, Inc.

Popek, G. J.; Goldberg, R. P. (July 1974). "Formal requirements for virtualizable third generation architectures". Communications of the ACM. 17 (7): 412–421. doi:10.1145/361011.361073.

"Popek and Goldberg virtualization requirements." *en.wikipedia.org*. Wikipedia Foundation, Inc., 25 June 2017. Web. 31 July 2017. <https://en.wikipedia.org/wiki/Popek_and_Goldberg_virtualization_requirements>

"Production-Grade Container Orchestration." kubernetes.io. The Kubernetes Authors, 2017. Web. 31 July 2017. <https://kubernetes.io>

Radio Frequency Interface Specifications, SP-RFII01-970326, Mar. 26, 1997, Cable Television Laboratories, Inc.

Remote Downstream External PHY Interface Specification, CM-SP-R-DEPI-I07-170524, May 24, 2017, Cable Television Laboratories, Inc.

Remote PHY Specification, CM-SP-R-PHY-I07-170524, May 24, 2017, Cable Television Laboratories, Inc.

Tholeti, Bhanu P. "Learn about hypervisors, system virtualization, and how it works in a cloud environment." *ibm.com*. IBM, 23 Sept. 2011. Web. 31 July 2017. <https://www.ibm.com/developerworks/cloud/library/cl-hypervisorcompare/index.html>

"Virtualization." *en.wikipedia.org*. Wikipedia Foundation, Inc., 11 July 2017. Web. 31 July 2017. <https://en.wikipedia.org/wiki/Virtualization>

"Virtualization and the Internet of Things." *windriver.com*. Wind River Systems, Inc., Jan. 2016. Web. 31 July 2017. <https://www.windriver.com/whitepapers/iot-virtualization/1436-IoT-Virtualization-White-Paper.pdf>

"Virtualization: The Promise vs. The Reality." *westconference.org*. Tintri, Apr. 9 2015. Web. 31 July 2017. <http://westconference.org/WEST17/CUSTOM/pdf/WEST_Tintri_Thoughtleadership.pdf>

"What are microservices?" *opensource.com*. Opensource.com, n.d. Web. 31 July 2017. <https://opensource.com/resources/what-are-microservices>

"What are Microservices?" *pivotal.io*. Pivotal Software, Inc., 2017. Web. 31 July 2017. <https://pivotal.io/microservices>

"What is Cloud Native?" *cncf.io*. The Linux Foundation, 2017. Web. 31 July 2017. <https://www.cncf.io/>

"What is a virtual appliance?" *turnkeylinux.org*. Turnkey Linux, n.d. Web. 31 July 2017. <https://www.turnkeylinux.org/virtual-appliance>

Yadav, Rishi. "What real cloud-native apps will look like." *techcrunch.com*. TechCrunch, 3 Aug 2016. Web. 31 July 2017. <https://techcrunch.com/2016/08/03/what-real-cloud-native-apps-will-look-like/>