SCTE ISBE CABLE-TEC EXPO®'16

SEPTEMBER 26-29 PHILADELPHIA

SCTE Society of Cable Telecommunications Engineers

ISBE International Society of Broadband Experts

Distributed Access Architecture – Goals and Methods of Virtualizing Cable Access

A Technical Paper prepared for SCTE/ISBE by

Nagesh Nandiraju, Ph.D.
Director, Next Generation Network Architecture
Comcast
1701 JFK Blvd, Philadelphia, PA 19103
215-286-1941
Nagesh_nandiraju@comcast.com

Table of Contents

Title **Page Number**

Introduction 3
Distributed Access Architectures 3
NFV and SDN in Edge and Access Networks 4
Home Run for NFV 5
High Level Architecture of Virtualized Headend 7
Virtual Network Function Architectural Options 9
Lab evaluations 11
Summary 14
Abbreviations 15
References 15

List of Figures

Title **Page Number**

Figure 1 - Centralized Access Network Architecture 5
Figure 2 - Distributed Access Network Architecture 5
Figure 3 - High Level Archicture of Access Network with NFV 7
Figure 4 – Orchestration Scopes in Distributed Access Network 8
Figure 5 – Side by Side View of Bare-Metal, Virtual Machine and Containers 10
Figure 6 – (a) Bare-Metal – VM scenario (b) VM-VM to scenario 11
Figure 7 – TCP Throughput between two Virtual Machines on different Physical Hosts 12
Figure 8 – Bare-Metal and VM scenario 13
Figure 9 – TCP Traffic Performance metrics in a VM with and without Traffic Control 14

Introduction

Web-scale designs are innovating at fast pace and the separation between a traditional Telecommunications Networking industry and software-controlled world is blurring day by day. In a new world where the frequency of a finger touch on every Mobile and Web apps is increasing every minute, change is no longer a dreaded dream. Users of the technology are constantly seeking for the flexibility with any services they are associating with. The expectation is to have any and every service, be it live video streaming or virtual reality hangouts, be mobile and ubiquitous. Mobility implies not just geographical mobility but also mobility across devices and software platforms. Service providers are now challenged with addressing these needs, creating the immersive experiences for customers in addition to the chores of maintaining the network and increasing data pipe capacities [1][2].

Distributed Access Architecture (DAA) [1] is a method by which some part of the access technology is put into a fiber node, and this can include DOCSIS, PON, WiFi or Ethernet. With DAA, the Fiber Node now becomes an active element in the IP network, with instrumentation and reporting capabilities.

Network Function Virtualization (NFV) and Software Defined Networking (SDN) are two powerful concepts that enable the transformation of networking from purpose-built close knit platforms to an open and loosely coupled platforms [3]. This decoupling of traditional network functions from the underlying hardware brings in opportunities for new innovative architectures, and enables agility. Specifically, NFV at a high-level virtualizes conventional network functions into software running on Commercial Off-the-Shelf (COTS) Hardware. The resulting software functions are also called as Virtual Network Functions (VNFs). SDN programmatically manages and controls these VNFs, taking advantage of the powerful abstractions enabled by NFV and making use of any APIs exposed by the VNF.

In this paper, we will first walk through the “why” and “how” of DAA. We will then discuss some design principles such as disaggregation and loosely coupled architecture that facilities the introduction of new technologies seamlessly. We then present the virtualization architecture for the headends, explain some of the design and architecture choices for virtualized network functions (VNFs) and present some preliminary results from our evaluations in the lab.

Distributed Access Architectures

It is by no means surprising to hear the nature of data growth and how it is driving the capacity increases and evolution of the networks. Up until now the existing access terminating systems (CMTS) have been aggregating the traffic and exploiting the statistical multiplexing concept. As a large number of customers are aggregated, the overall capacity is efficiently shared among those users. This is possible because the peak usage of various types of traffic are not always aligned and even if they do align, the nature of IP networks is to spread the load over time. When capacity increases were required, merely adding more RF spectrum for Data was the solution. This was manageable for two key reasons: a) legacy video has been the relatively dominant consumer of the RF spectrum even with the 50% CAGR of data and b) peak capacity was still mostly satisfied by a handful of RF channels and the CMTS had enough horse power to aggregate a large number of subscribers. While this increase in capacity meets the needs of a usage based capacity provisioning, the challenge lies with the need for capacity provisioning driven not just by usage but a forward looking peak capacity requirement. Further while there is RF capacity that can be increased in the downstream direction (e.g. from 750MHz/860MHz to 1GHz/1.2GHz), the upstream is capped at 42 MHz due to the HFC network built several years ago. This limits how much longer we can continue to incrementally upgrade the capacity without rebuilding some of the HFC plant.

This lead us to rethink the access network design and strategy. Instead of continuing the age old process of incremental upgrades, we were seeking alternate designs such as Fiber deep with N+0 architecture. This N+0 architecture loosely implies an order of magnitude smaller number of homes passing per node. The key feature of this approach is that it is fairly long lasting in terms of capacity needs. However, the implications of the new design on facilities such as space, power, cooling are fairly substantial. Many Hub or Headends would burst out at seams to implement this architecture using the legacy solutions. In addition, the operational challenges and performance inferiority associated with the analog optics of the HFC compared to Digital optics were quite evident. So a different access architecture that takes advantage of the the substantial change in the outside plant architecture was needed. That is the conception of Distributed Access Architectures.

The distribution of some parts of the Access Network technology is quite clear, but what and how to split, and what would be the resulting architecture inside the headend and outside plant were the next set of explorations.

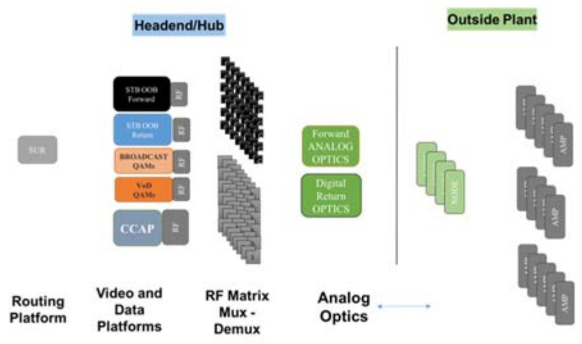
NFV and SDN in Edge and Access Networks

As we were working on the transformational need in the access architecture, in parallel, there were many significant technological transitions happening in the broader networking industry that are distantly related to our challenges. In particular, we are witnessing major transformations in optical network bandwidth, switching, server, and storage platforms. These are primarily driven by the hyper scale data center deployments. For example, Facebook notes [4] their 200x increase in the network bandwidth per rack of servers from 2Gbps per rack to 400Gbps in just 6 years.

There is at least one key pattern enabling these dramatic changes– **Disaggregation and Abstraction** – in a cost effective way. **Disaggregation** is the concept of first breaking down a function or a system (represented as a single unit) into multiple smaller core components and then organizing them back together. The principle of **Abstraction** is to define clear interfaces for each component. As the core components are combined back together, each component's interfaces are defined. Any interaction with a component will be via the interfaces specified. Each component is responsible to scale and maintain its interfaces. This breaks the bigger problem to be solved into multiple smaller problems, agility and simplicity is achieved. Best of breed components can be mated together. For instance any network function may be broken down into multiple sub components. Each sub component may expose interfaces also known as Application Program Interface (API). Components interacting with other component's APIs can change its implementation logic without being too dependent on other components as long as the APIs are not violating the interface definitions.

The concepts of NFV and SDN, no longer just buzz words, originated from the unique needs of Hyper Scale Data centers and are based on these above defined concepts. Nearly a decade ago, the exponential growth trend of networking bandwidths, driven by data storage and compute (CPU processing) needs in inter and intra datacenter environments, couldn't be met with closed-monolithic business and technological solutions. To solve this problem, classical computer science principles - disaggregation and abstraction - were applied. Hardware and Software were decoupled, programmatic APIs using fungible code were developed minimizing the need of standards based protocols, Hardware was well defined to be based on Merchant silicon (high volume application specific chips that have well defined programmatically accessible interfaces). This changed fundamental economic, technological and operational models of networking inside data center. Similar changes occurred in the storage and compute space. Distributed Network Software systems that are decoupled from hardware is the basis for Network Function Virtualization (NFV). Software Defined Networking (SDN) is the result of a need for a framework to manage the decoupled, disaggregated system components.

Access Networks also had similar challenges. Multiple Access Network technologies such as HFC, PON, Ethernet, Wireless etc. that have common goals, such as providing near identical set of services to the end customers, increase the provisioned capacity exponentially. This is a good candidate problem to be solved by abstraction. The first set of decoupling is the distribution layer between the Headend and physical media conversion point at the Optical Node. Digital optics using 10G Ethernet clearly bring a lot of benefits and simplicity. For HFC, placing the media conversion and the RF processing in the node closer to the physical demarcation between Fiber and Coax seemed attractive and simplified the technological design, albeit this presents a radical change in operations. Other Access technologies can also reuse these common set of abstract interfaces (physical and logical) to interconnect between the node and headends. As shown in the Figure 1, all the remote access devices (RADs) are aggregated by high density switches. For convenience we refer to this as Distributed Access Aggregation Switch (DAAS).



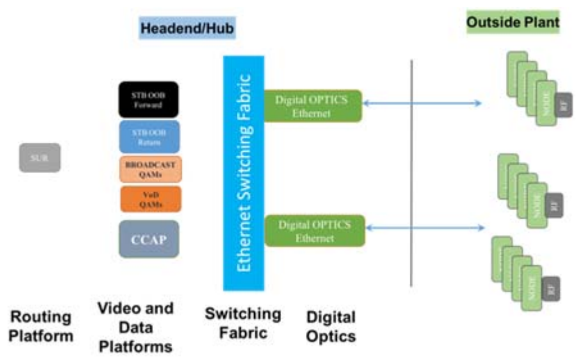


Figure 1 - Centralized Access Network Architecture

Figure 2 - Distributed Access Network Architecture

Home Run for NFV

Now that the physical interconnection between the headend and Node is established, the scope of remaining functionality in the headend deals with packet processing. This is very close to the scope of the data center networking challenges. In HFC, and in particular DOCSIS, the macro block (CCAP-Core) that is placed in the headend includes the upper layers of the DOCSIS. The functionality includes features such as DOCSIS scheduling, encapsulation, decapsulation, encryption, routing etc. Advances in general purpose compute platforms and software mechanisms on top of these platforms (e.g. DPDK, SR-IOV, VPP) make it viable to implement these features on the compute platforms in a manner that is similar to that being used in web-scale data centers. Many routing vendors have publicly presented the performance of several VNFs such as virtualized routing solutions achieving 80 Gbps and/or higher.

Figure 3 shows the high level architecture of NFV based Headend/Hubs and abstracted Access networks. Starting from the bottom of the figure, inside a customer's home, data is being primarily consumed over WiFi. A generic device class (commonly referred as Home Gateway) provides consistent access to the Wide Area Network over any media (e.g. Wireless, CAT6, MoCA) for in-home access (LAN) network. The pipe that is coming from the Access Network into the home can be on media based on any technology. Specifically, it can be DOCSIS on Coax, PON over Fiber or Ethernet over Cat 5e/6 media. These are converted into the Gateway using Cable Modem (CM), Optical Network Unit (ONU) or Ethernet Gateway respectively.

In the Outside plant, the access network technologies such as DOCSIS, PON or Ethernet will be handled by their respective functional boxes. Note here that these access media converting functional boxes can live inside the Headend/Hubs or in the Outside plant (this is illustrated by the "(R-)" in the parenthesis). In DAA, one may assume that these devices live in the outside plant.

From an architecture point of view, the abstraction interfaces are consistent inside or outside. Where it will make a difference is in the operational guidelines and in the specs for the environmental and maintenance aspects.

Inside the Headend/Hub, a pool of compute, storage and networking resources is referred to as the NFV infrastructure and form the basis for any software running to control the access network elements and technologies. This infrastructure can be molded by software to serve multiple services, applications and VNFs.

Application/VNFs are hosted on Compute platforms for performing the control/management/Data plane packet processing functions – a data centric design for Headends with hardware and software of access networks decoupled – a Homerun for NFV!

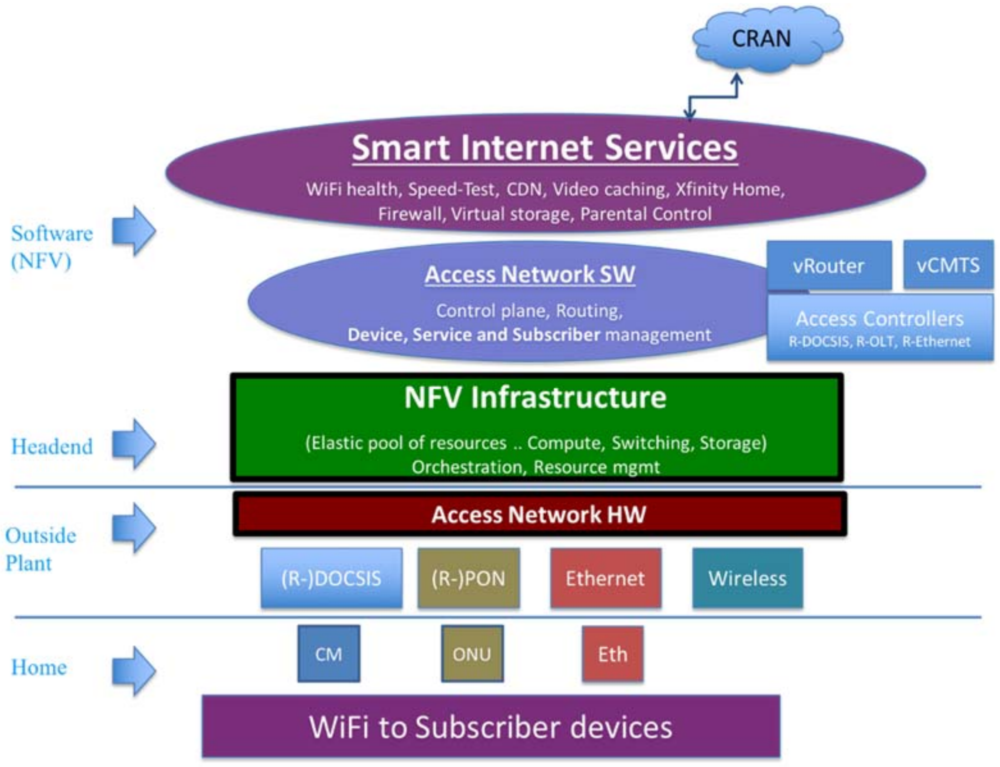

CRAN
Smart Internet Services
WiFi health, Speed-Test, CDN, Video caching, Xfinity Home, Firewall, Virtual storage, Parental Control
Software (NFV)
Access Network SW
Control plane, Routing, Device, Service and Subscriber management
vRouter
vCMTS
Access Controllers
R-DOCSIS, R-OLT, R-Ethernet
NFV Infrastructure
(Elastic pool of resources .. Compute, Switching, Storage) Orchestration, Resource mgmt
Headend
Access Network HW
Outside Plant
(R-)DOCSIS
(R-)PON
Ethernet
Wireless
Home
CM
ONU
Eth
WiFi to Subscriber devices

Figure 3 - High Level Archicture of Access Network with NFV

The implementation and software architecture of the virtual network functions can vary a lot now due to the myriad of options for implementation. With more options, management and control become more challenging. In the next section, we will discuss the operational aspects and the various options in the architecture.

High Level Architecture of Virtualized Headend

In this section we will discuss the high level architecture of a virtualized headend and briefly present the options and their implications. Figure 4 shows the high level architecture of a headend. Since the compute resources meet the basic needs for any application, the challenge is how to effectively manage them and plan for deployments. Typically, compute resources are deployed and operated in large pools. This helps in optimizing operational aspects such as better resource utilization by scheduling, maintenance, spares for replacements, technical expertise to operate and manage, power and resiliency of facilities. Operationally for smaller facilities (aka Hubs), one option would be to simply have an aggregation and forwarding plane, i.e. a DAAS, that folds into a spine at a central facility. This facilitates concentrating compute/memory resources pools in fewer locations. However, there are still a large number of facilities that need to be coordinated and operated. The work flows that are designed over the years for building the network with incumbent vendor driven designs may introduce large delays in building services over the top. This can be either because of the lack of features from a very old hardware designs or it can simply be the natural progression of work transition and knowledge gaps between multiple teams that are not necessarily living and breathing the same concepts. This is where orchestration and controllers come into the picture. The

options for Virtualized infrastructure management and Virtual Network functions and how they are controlled and orchestrated will be discussed below.
The architecture can be split into multiple logical domains as shown in Figure 4:

- Underlay network
- Overlay network
- Access Network
- Services

- **Underlay network** comprises of the physical connection between the servers, switches and the RADs. Setting this physical portion of the architecture via a controller is the orchestration of the Underlay network.

- **Overlay Network** is a broadly used term used in the networking industry. For our context, by Overlay network, we imply the network that needs to be established to facilitate the interaction of the Access network software, subscriber management with the remote access devices. For example, the L2TPv3 tunnels to the Remote PHY device.

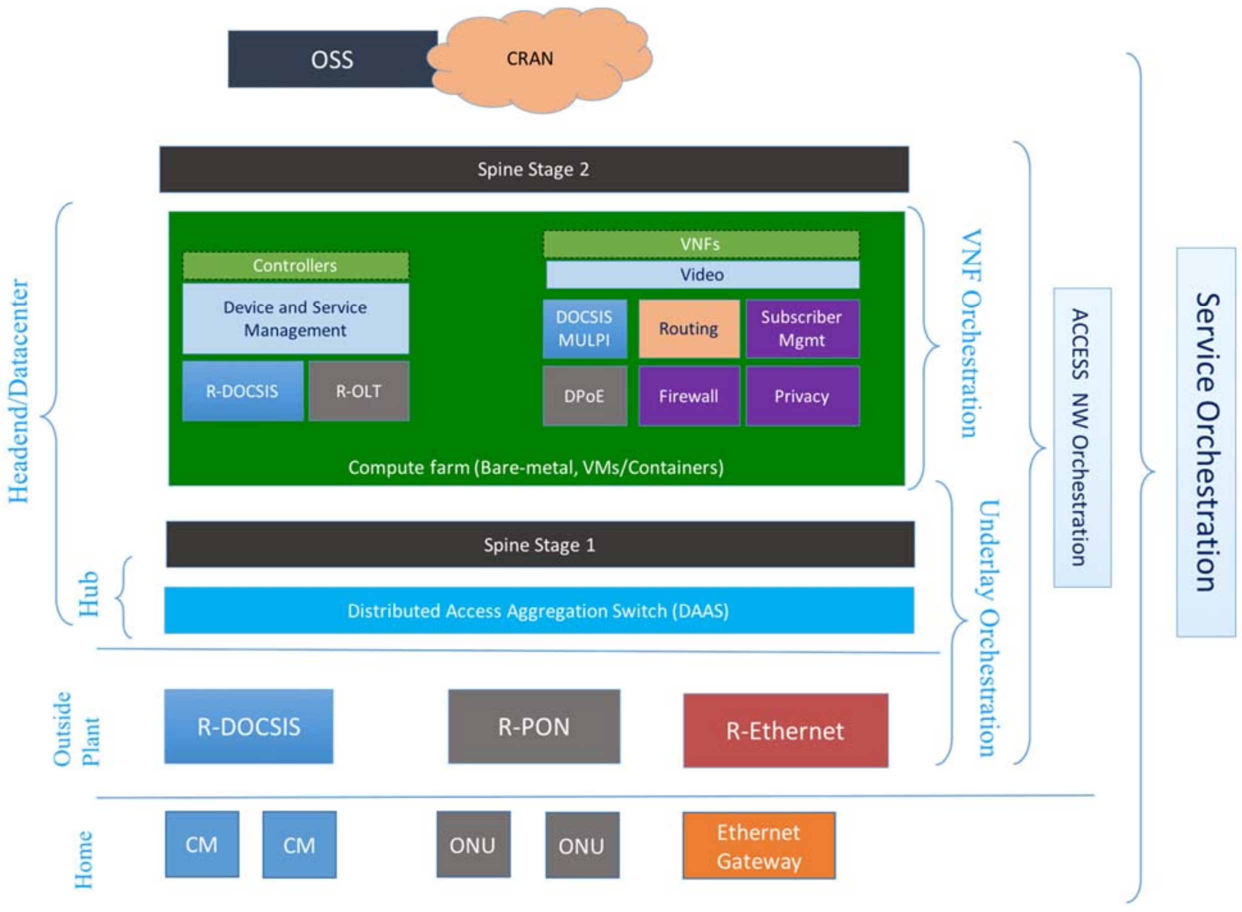


Figure 4 – Orchestration Scopes in Distributed Access Network

- **Virtualized Edge Components and VNF Orchestration**
  - Besides Access Network functionality, several other VNFs can co-exist in the same infrastructure. This presents a major advantage – new applications and services can be seamlessly deployed. This also brings up the challenge we initially discussed of capacity management. Now that the underlay network based on the new access network design is not the limitation, capacity management of the compute infrastructure is the new process that needs to be engineered. Instead of just looking at the data usage needs, the capacity management workflows need to take into account compute resources requirements for various applications.

- **VNF orchestration** is comprised of the Scheduling, Deployment and Management of the life cycle of the Applications/VNFs.

- **Access Network Orchestration** is comprised of the automation of provisioning an access network using the underlay, overlay and the VNF orchestration. For example, consider deploying a new DOCSIS service group in a DAA. The remote devices need to be provisioned, the virtual resources need to be carved out, the VNFs for DOCSIS, Routing, and Subscriber management need to instantiated of the services that are built using one or more application VNFs. These are comprised of several workflows and are orchestrated and controlled by the access network controller.

- **Service Orchestration** is comprised of the end to end orchestration of the services that are built using one or more application VNFs. This provides a service to an end customer.

Virtual Network Function Architectural Options

In this section, we will briefly cover the various form factors of VNFs, differences in the environments and implications of how these VNFs can be deployed. There are three major ways VNFs can be deployed: Bare-metal, in a Virtual Machine (VM) or inside a Container. The definition and differences are elaborated below.

Bare-metal vs VMs vs Containers

- This is one of the most common questions many people have on their mind. Figure 5 illustrates the three environments. The three hosts (servers) have the same hardware configuration but the internal operating system and software that guides the application environment are configured differently. In a Bare-metal implementation, the application/VNF is running as a native process using the host operating system, Kernel, and libraries. The dependencies of the application are completely tied to the host operating system. This can be challenging when managing a large pool of server infrastructure. The operator needs to ensure the dependencies of the myriad of applications are somehow met. This often results in a bloated host system that is challenging to maintain Any mismatch in dependencies can lead to application failure during subsequent roll outs. Another drawback of this approach is that the common pool of resources will likely end up partitioned for some applications, resulting in a tightly coupled system architecture.

- In a Virtual Machine environment, there is a hypervisor that can host multiple operating systems. Each operating system represents a Virtual Machine (VM). The VMs can communicate with each other via a local soft forwarding fabric. This can be accomplished via the host kernel bridging or using a software such as a virtual switch or virtual router. A commonly used software is Open vSwitch (OVS). OVS as loosely implied by name is a switch and each VM can attach to one or more ports on the OVS. Packets between VMs intra host or inter host is facilitated by overlay networks.

The framework of VMs enables the application/VNF provider to completely package their software with all the dependencies. The operator/Service provider (representing the infrastructure provider) merely deploys them as a VNF on top their pool of hosts (compute infrastructure). The operator/service provider maintains the host operating system in their pool of resources and is independent of the application/VNF provider who wants to deploy the application and provide service using this VNF. The demarcation and abstraction are very clear and hence the deployment can be very smooth. This helps us gain the much required operational agility and loosely coupled systems.

- A container environment derives all the goodness of a VM environment but with less overhead. The fundamental difference between a VM and a container is the overhead. Typically, the host machine runs a kernel, hypervisor and a switch. The VM runs on top of a hypervisor using its own operating system. A virtual switch (or router) bridges packets between the physical ports of the host and logical interfaces of the VMs. In contrast Containers are packages that run on top of the host operating system but they reuse some parts of the host kernel. They are segregated with separate namespaces (linux cgroups). The application dependencies are all packaged into the containers and only very generic needs are met by the host operating system.

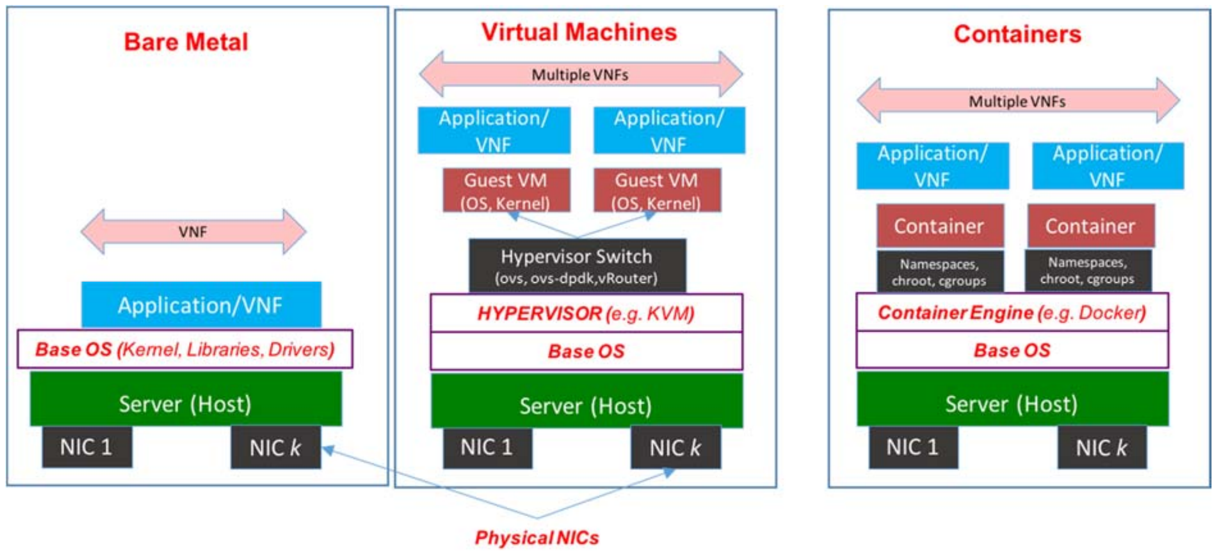


Figure 5 – Side by Side View of Bare-Metal, Virtual Machine and Containers

Implementation choices for CCAP-Core

In this section we will cover a brief overview of the implementation options of CCAP to adopt to the Distributed Access Architecture (DAA). Recall that in a DAA, some functionality of the CCAP is disaggregated and the core components can be implemented in different ways. For example, consider the case where RF functionality is implemented in a remote device. This leaves the core DOCSIS functionality to be implemented in the headend – this subset is termed as "CCAP-Core". There are several possible implementations of a CCAP-Core feature set:

a) CCAP-Core in a Chassis

b) CCAP-core as a VM or
c) Micro-services architecture.

In the first option, current CMTS/CCAP device may continue to host the functions in the current chassis. The existing hardware (line cards) requires some modifications. However, since the hardware and software in these chassis were designed without envisioning the DAA, they tend to carry over the scaling limits that existed in the original legacy centralized architecture.

As an alternative design, in the second option, the CCAP-Core software can be ported onto a virtual platform, either as a bare-metal implementation or in a Virtual Machine. As a third option, one can take this opportunity to embrace a micro services design and disaggregate the monolithic CCAP-core into multiple smaller pieces. An elaborated description of these options are out of scope for this paper but we hope to cover them in the future.

Lab evaluations

We have engaged with several vendors to work on multitude of proofs of concepts (PoCs) to close the gap between "what if" and "what is". At the time of this writing a variety of VNFs are being evaluated, including virtual routers, virtual firewalls, speed-test servers, virtual CPE functions, and data traffic analysis. We specifically started with hypervisor environments, Openstack based networks and bare-metal solutions. As with any technology, the spectrum of options is pretty wide and the options contain several knobs to tune the performance. For something as simple as a Virtual TCP traffic generator we ended up with over 100 combinations. We investigated network intensive workloads on common off the shelf (COTS) platform by enabling Data Plane Development Kit (DPDK), Single Root IO Vector (SR-IOV). DPDK provides a powerful abstraction of underlying hardware for enabling data plane intensive work loads on COTS hardware. SR-IOV enables the traffic from a NIC card to directly reach a VM bypassing the soft-switching layer. OVS-DPDK is an augmentation of OVS using DPDK and has presents significant performance improvements (as high as 17x [5]). The performance of TCP over OVS-DPDK on top of a KVM is compared with other alternate combinations.

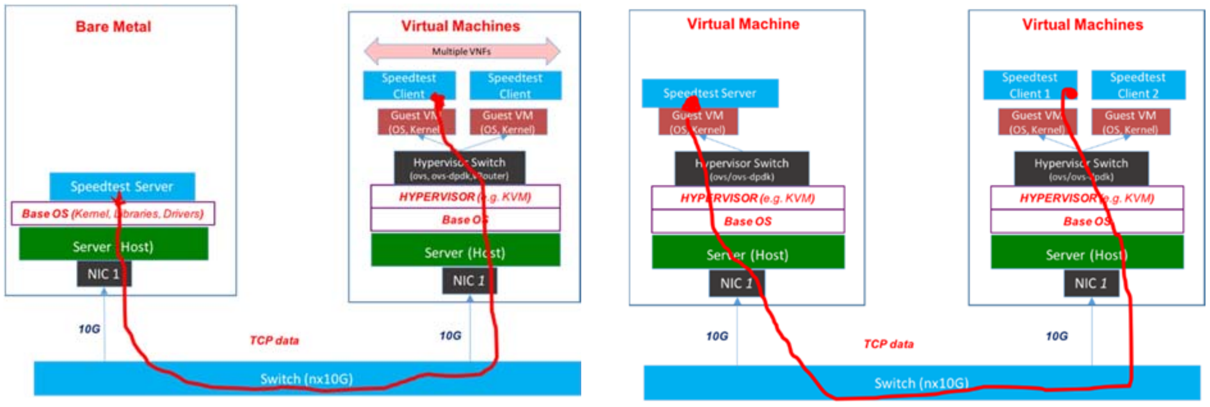
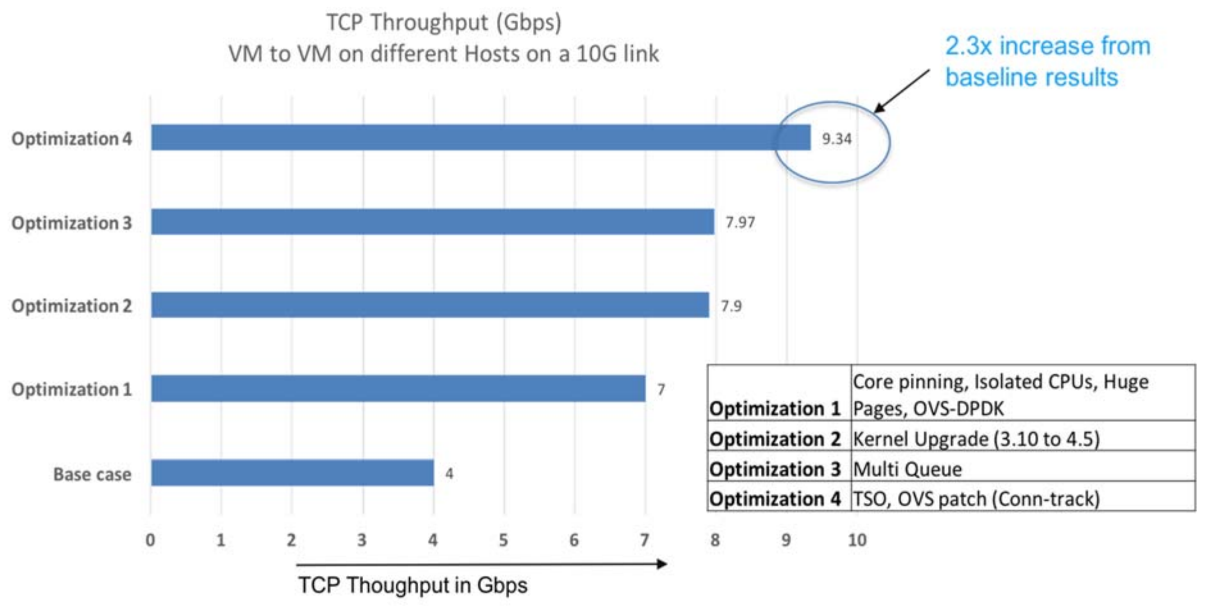


Figure 6 – (a) Bare-Metal – VM scenario (b) VM-VM to scenario

Figure 6 (a) and (b) illustrates the setups for evaluating the TCP performance on both bare-metal and VM environments.

We quantified the TCP throughput between virtual machines running on two different hosts and compared it to a bare-metal equivalent setup. It turns out that there are several factors that impact the TCP performance, such as the linux kernel, the CPU isolation, and the operating system.

Figure 7 shows the TCP throughput between two VMs running on different hosts attached to a same 10G switch. When we started out, the TCP performance was around 4Gbps. This was after we had already upgraded the OVS-DPDK (which moves packets between the Physical NIC card and the VMs). OVS-DPDK public published performance results using UDP were much higher. Albeit those results were using UDP, we were still puzzled by the inferior performance and started investigated the different knobs. Working closely with Intel, we identified the optimizations and helped fine tune the system performance. We tried over 100 combinations of optimizations. Figure 7 demonstrates a summarized version of all optimizations that resulted significantly improved the performance.



Optimization 1	Core pinning, Isolated CPUs, Huge Pages, OVS-DPDK
Optimization 2	Kernel Upgrade (3.10 to 4.5)
Optimization 3	Multi Queue
Optimization 4	TSO, OVS patch (Conn-track)

Figure 7 – TCP Throughput between two Virtual Machines on different Physical Hosts

We also carried out similar tests of TCP Performance between a CM and a virtualized Speed test server. Specifically, we started out by characterizing the performance differences of a network service from a customer point of view. Figure 8 shows the two scenarios where a hypothetical customer may run the speed test and reach out to either a bare-metal based server or a virtual machine based server.

- CM to Speed test server on Baremetal
- CM to Speed test server on VM

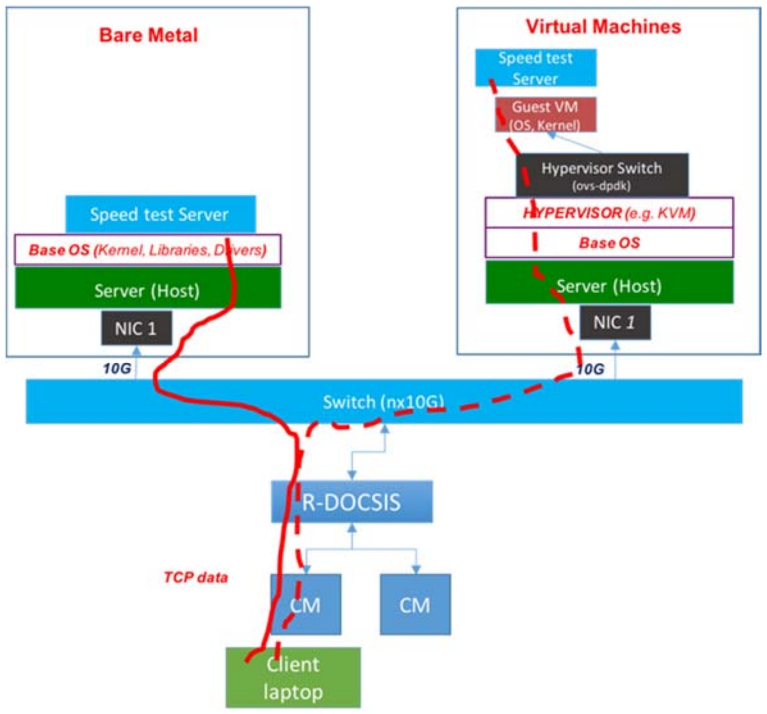


Figure 8 – Bare-Metal and VM scenario

Figure 9 shows the results of a TCP server sending traffic to CMs. The two scenarios in Figure 8 are: a) “VM without TC” and b) “VM with TC”. In scenario (a) the traffic is passed unrestricted without any traffic control (TC) using queuing mechanisms. In scenario (b) traffic is passed through a Hierarchical Token Bucket (HTB) and traffic is regulated into the network.

The initial tests were very disheartening. While the bare-metal servers we had in the lab were 1Gbps capable, without tuning the DOCSIS layer, we achieved a TCP download throughout of around 880 Mbps. On a VM setup we saw around 60 Mbps, an order of magnitude lower. It turns out that it was an operator architecture error. The poor results were because of the buffer over runs due to interface speed mismatches. The VM setup was pumping traffic greater than 1Gbps but the DOCSIS plant was not capable of those higher speeds. The result was that TCP packets were dropped in every burst of TCP transmission, as clearly indicated by the TCP retires and congestion window size in Figure 8 (referring to “VM without TC”). The Congestion window was repeatedly slashed after realizing the packet drops.

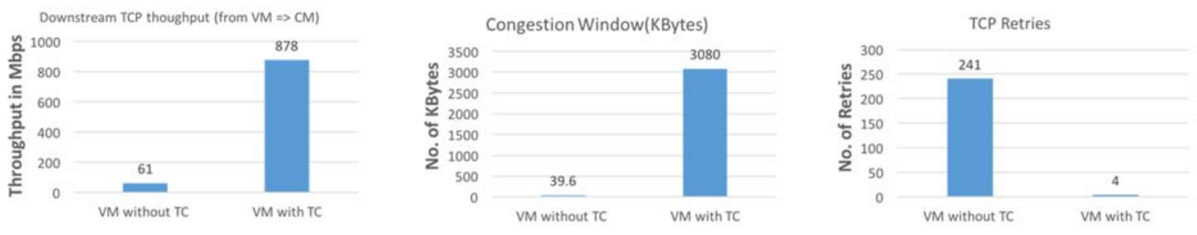


Figure 9 – TCP Traffic Performance metrics in a VM with and without Traffic Control

After applying the traffic control using HTB, the downstream TCP throughout from a VM based speed test server is back to comparable results with the bare metal speed test server. The average Congestion window is jumped up and the retries are negligible. The 4 retries shown only occur in the beginning of the TCP stream start. At the time of the writing we haven't figured out the cause of this behavior.

These performance results are preliminary in the world of virtualizing network functions at the edge and access networks and are far from the perfect results that are supported by the underlying platform. There are many knobs that we haven't fine tuned. We are continuing to iron out the kinks of the platforms, and fine tune those knobs. Throughout our evaluations, we are clearly seeing the benefits of agility in deployments and an automatable workflow that improves the overall resiliency of the network functions and services on top of these.

Summary

Access Networks are undergoing a significant architectural transition by way of distributing and disaggregating the access network functionality. The advantages associated with DAA include cost effective and long reach enabling digital optics, enormous space and power efficiencies, higher performance of DOCSIS, network function virtualization, and abstraction of common physical and logical interfaces.

DAA enables a simpler strategy for transitioning access networks to networks using NFV and SDN. While there are several benefits of NFV and SDN, there are a myriad of options in the architecture and choice of components both from vendors and open source. The challenge is to filter them and make a stable framework that provides the fundamental needs of access network functionality. Efforts such as OPNFV, CORD are just a few examples that are striving to provide a stable framework and encourage the cable and telecom community to start using NFV. To unleash the potential benefits enabled by NFV and SDN, both service providers and vendors need to adapt their development, operational and business models. Agile dev-ops models seem to be quintessential in this new era of networking.

Abbreviations

CM	Cable Modem
bps	bits per second
CMTS	Cable Modem Termination System
CCAP	Converged Cable Access Platform
DPDK	Data Plane Development Kit
HFC	hybrid fiber-coax
HTB	Hierarchical Token Bucket
NFV	Network Function Virtualization
OVS	Open vSwitch
SDN	Software Defined Network
VM	Virtual Machine
VNF	Virtual Network Function

References

[1] The Fiber Frontier, by Dr. Robert L. Howald. Published in Spring Technical Forum, NCTA 2016

[2] Growth Architectures: Built to Last, Built to Launch, by Dr. Robert L. Howald. Published in Spring Technical Forum, NCTA 2014

[3] Delivering Seamless Subscriber Aware Services over Heterogeneous Access Networks using a SDN and NFV framework, by Dr. Nagesh Nandiraju et al., NCTA 2015

[4] https://code.facebook.com/posts/973406756030104/adopting-an-open-approach-to-global-networks-with-the-telecom-infra-project/

[5] Intel® ONP Release 2.1 Performance Test Report https://download.01.org/packet-processing/ONPS2.1/Intel_ONP_Release_2.1_Performance_Test_Report_Rev1.0.pdf