CAN DOCSIS NETWORKS LEVERAGE SDN?

Gerry White
Arris

Abstract

Software Defined Networking (SDN) offers the promise of simplified network operation through centralization of functions, automation and programmable interfaces. Although SDN is still in the early phase of its evolution and has initially focused on data center networks it is appropriate to consider whether it could provide any advantages in a DOCSIS infrastructure. This is especially relevant as operators deliver more services from a cloud environment based on virtualized data center technology.

Accordingly this paper looks at the potential advantages of moving components of the DOCSIS ecosystem to an SDN environment. In particular it takes a detailed look at the CMTS and ancillary servers to determine whether a different decomposition of functionality would enable the industry to better leverage this technology. A possible implementation of an SDN based CMTS is proposed and compared to existing implementations As part of this evaluation the DOCSIS protocol itself is considered and potential changes suggested to remove roadblocks and make it more "SDN friendly".

INTRODUCTION

One of the most significant developments in service delivery over the last decade has been the evolution of data centers to deliver cloud computing on a massive scale. Low cost computing and storage platforms coupled with sophisticated virtualization techniques have enabled the delivery of complex services at very low cost points. The deployment of these large scale virtual server farms has illustrated some of the problems with conventional networking equipment, especially in the areas of cost and operational complexity. Software Defined Networking (SDN) has evolved as a potential answer to these problems. This paper looks at some of the basics of SDN, how it might be applied to deliver high speed data services in an MSO network, the benefits which could accrue and the problems to be resolved.

SDN

As workloads change in a data center large numbers of virtual machines must be instantiated, destroyed or moved to optimize use of compute and storage devices. The network must have the capacity to connect all of these devices at high speed with acceptable cost. More critically it must keep up with these changes as they occur which is problematic if new cables must be connected or even if large routing domains must re-converge on each topology change. Software Defined Networking has evolved as an attempt to address these challenges.

In a traditional network each network element such as a switch or router is composed of a data plane and a control plane. The control plane is typically composed of complex software components such as routing protocols executing in an embedded general purpose CPU on the device. The control plane software exchanges messages with other devices in the network (e.g. via a routing protocol) to determine the network topology and state and uses this information to create the forwarding tables used by the data plane to control packet forwarding. Thus each device has its own view of the network and all devices must cooperate to provide the required end to end paths. If forwarding is to work correctly then all of the local views must be coherent so that after a change or failure the devices must converge to a single view. This need for convergence causes a problem for rapidly changing topologies such as those found in data centers.

For high speed devices the data plane is typically built in hardware and performs line rate packet forwarding. When a packet arrives the data plane hardware looks up the address fields in the forwarding data base, selects the outbound port and forwards the packet.

This is of course a somewhat simplistic view of network operation but serves to illustrate the principals under consideration.

Figure 1 shows a sample SDN based network. It is composed of three layers, applications, network control and network infrastructure. The control plane functions which would be embedded in traditional network devices are moved to the control plane where they provide a centralized view of the network rather than the traditional distributed view. This single central view exposes Application Programming Interfaces (APIs) to the application layer. Thus business specific applications can control the network directly via high level interfaces rather than relying on the results of the traditional embedded control plane. The migration of the control plane away from the embedded devices also enables additional functionality to be provided. For example, the control plane can instantiate multiple virtual networks running over a single hardware infrastructure in a direct analogy to the virtual machine architecture used so successfully for data processing. The control plane itself can of course be run in a virtual machine environment to enable redundancy and scaling.

The network infrastructure contains the network devices themselves and provides the data forwarding plane for the network. The network devices in an SDN can be much simpler than traditional routers and switches as the control plane software has been removed and replaced by a much simpler module which updates the local forwarding database based on instructions from the central control plane. Thus the devices can be constructed from simple fast hardware with minimal software.

In order to enable the control plane to operate with network devices from different vendors a standard interface is required. Openflow [OPENF] is an interface specification defining the mechanism by which the control plane modifies the forwarding paths in the network devices. It is supported by multiple switch vendors and enables simple forwarding engines to be controlled by the SDN platform. Thus control of packet forwarding moves away from the network device and into the SDN platform. This enables low cost switches to replace more complex traditional switches and routers.

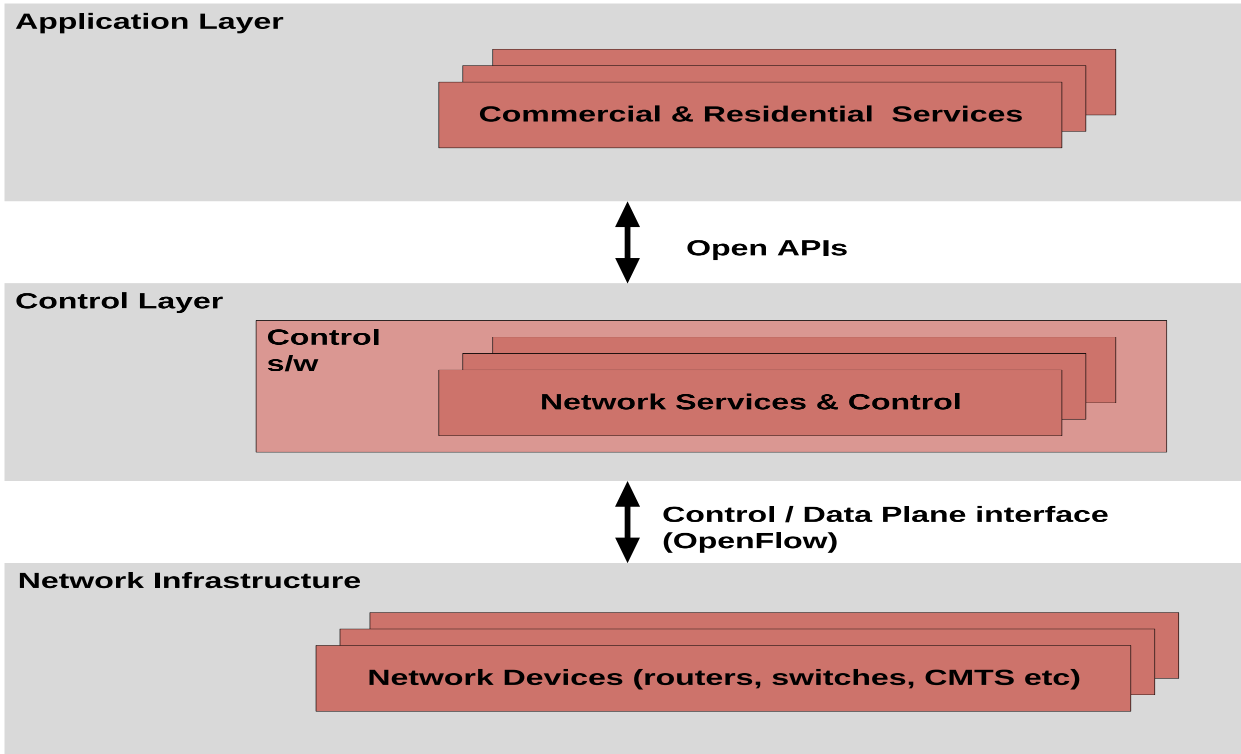

Figure 1 SDN Network Components

To date SDN has been primarily targeted at data center networks. The question to be considered is whether and how it could be used in a DOCSIS based MSO network and what benefits might accrue. The existing cable network is significantly different to those found in data centers. It is however evolving rapidly so that we will look at where it is likely to go and then look at the prospects for using SDN in this evolved network.

CHANGES IN CABLE INFRASTRUCTURE

Changes in cable infrastructure will be required to match the evolution of the services which must be delivered.

The following major trends are driving cable service evolution:

- Rapid expansion of the data rates which must be provided for high speed data services
- A move from broadcast to narrowcast video
- A move to IP delivery for video using Adaptive Bit Rate (ABR) protocols

The major impact of these changes to cable infrastructure is that they will require a significant expansion of the IP capacity currently in place. The existing core and regional networks are IP based but will need expansion to cope with the additional load. The access network is currently a hybrid of analog, MPEG and IP delivery and it is here that more dramatic changes will be required. As more services move to an IP base the capacity of the existing CMTS network used to deliver IP over DOCSIS will need to expand to accommodate this.

IP Expansion In the Access Network

Current CMTS ports are significantly more expensive than their video only MPEG equivalents. If this does not change then the dramatic expansion of IP capacity which will be needed could represent a significant CAPEX problem. This has been apparent to both operators and vendors for some time and has driven development of next generation DOCSIS platforms such as those based on the CCAP specifications [CCAP]. These platforms leverage high density silicon components to offer higher capacity and lower per channel costs in the same footprint as current CMTSs.

As the move to IP delivery continues it is appropriate to investigate whether additional changes to the MSO infrastructure would be advantageous and it is in this context that we will examine the potential for SDN.

Move to data center

Centralized data centers provide low cost processing for applications (and for the software components of an SDN). As described in [TRANS] a potential evolution of the cable network to leverage the advantages of a data center environment is a practical proposition. In this architecture the head end in its current form can be replaced by a data center, an Ethernet distribution hub and a simple node as shown in Figure 2. In this model all complex software has been moved to the data center and HFC specific MAC and PHY functions have been moved to the node allowing the use of standard Ethernet optics and switching from data center to node.

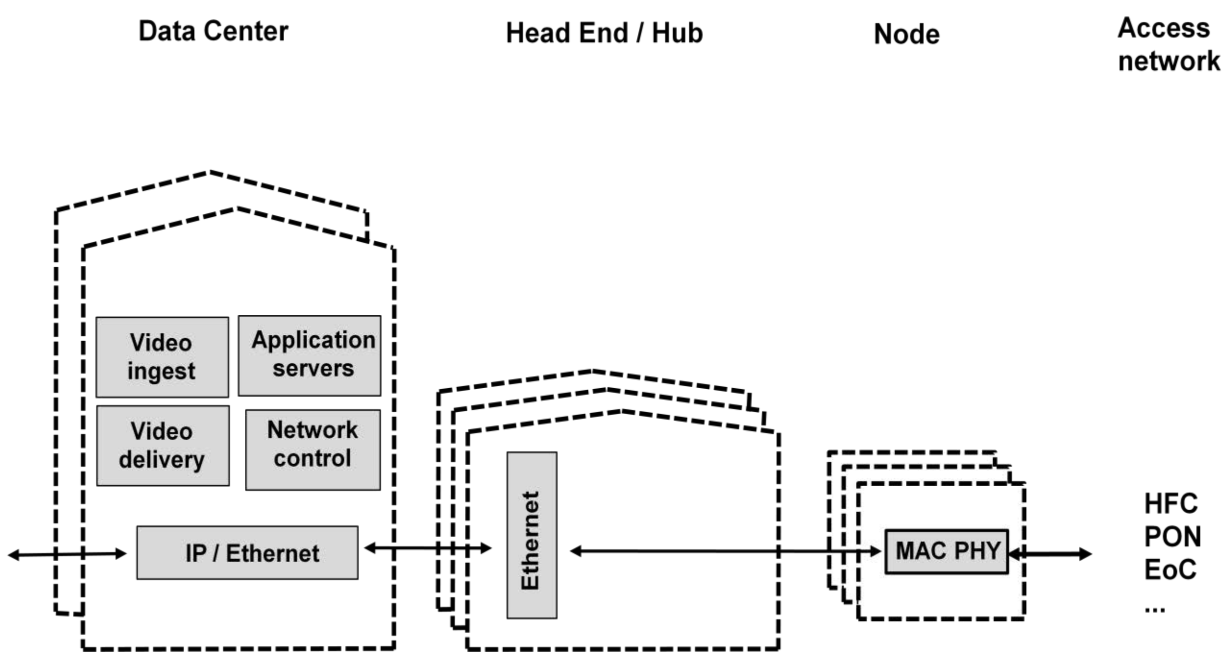

Figure 2: Distributed Cable Architecture

Moving the MAC and PHY functions from the head end to the node creates a more intelligent outside plant architecture. Operators who do not wish to take this step and prefer to keep a simpler outside plant can elect to deploy the MAC-PHY components in the hub rather than the node as shown in Figure 3. They still retain the advantages of the move to the data center and a significant reduction in hub complexity. Readers interested in an in depth comparison of traditional and intelligent HFC architectures are referred to [HFCDFC].

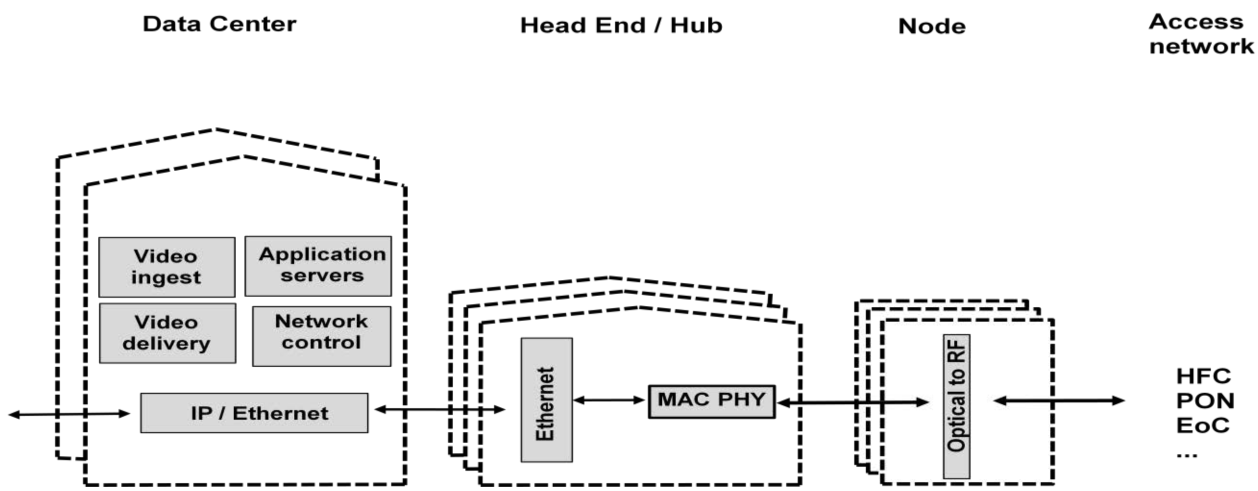

Figure 3: Passive HFC Architecture

SDN can be applied to either of these architectures in essentially the same manner. Thus they will be treated as equivalent for the purposes of this paper.

DOCSIS HEAD END

The previous sections outlined a potential move to a data center based architecture. We will now take a more in depth look at the DOCSIS head end infrastructure, how it could migrate to this type of platform and how SDN could be leveraged.

Figure 4 shows the major components of a PacketCable Multimedia system used to deliver QoS enabled multimedia services over DOCSIS. A detailed description can be found in [PCMM] but is not necessary to follow the paper which simply uses this as an example.

- An Application Manager /Server hosts the QoS-enabled applications and coordinates policy and QoS decisions.
- A Policy Server implements MSO-defined authorization and resource-management procedures which are enforced by the CMTS.
- The Record Keeping Server (RKS) tracks the usage of access-network QoS resources via message exchanges with the CMTS.
- A security server provides an authentication and security infrastructure support.
- The Operational Support system provides operations and management support.

- The CMTS provides data forwarding and DOCSIS control functions
- A managed IP network connects these devices together

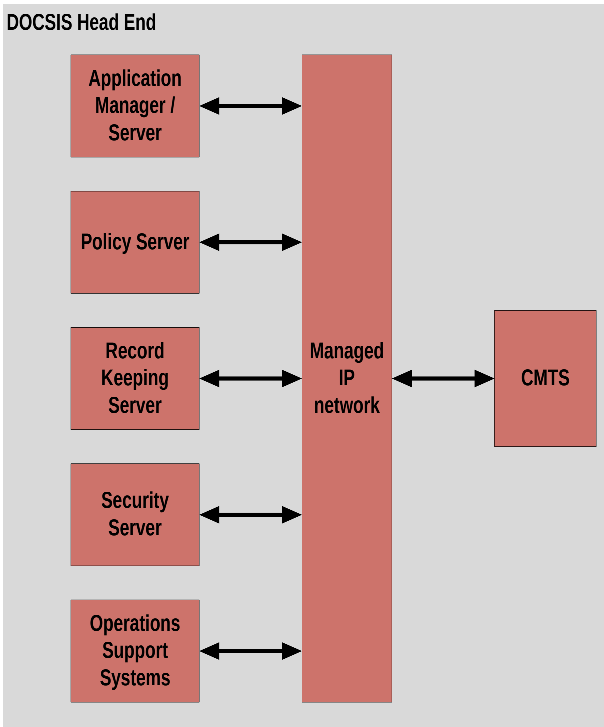

Figure 4 DOCSIS Head End

All of these components other than the CMTS and the IP network can be implemented as software running on standard server platforms and would be part of the application layer in an SDN based infrastructure as shown in Figure 5. The interfaces between the components would be modified for the centralized control layer but the functionality would remain largely intact.

The managed IP infrastructure could be evolved into an SDN architecture by moving the control plane into the SDN control layer and using OpenFlow switches in the same way that this has been done for data center networks.

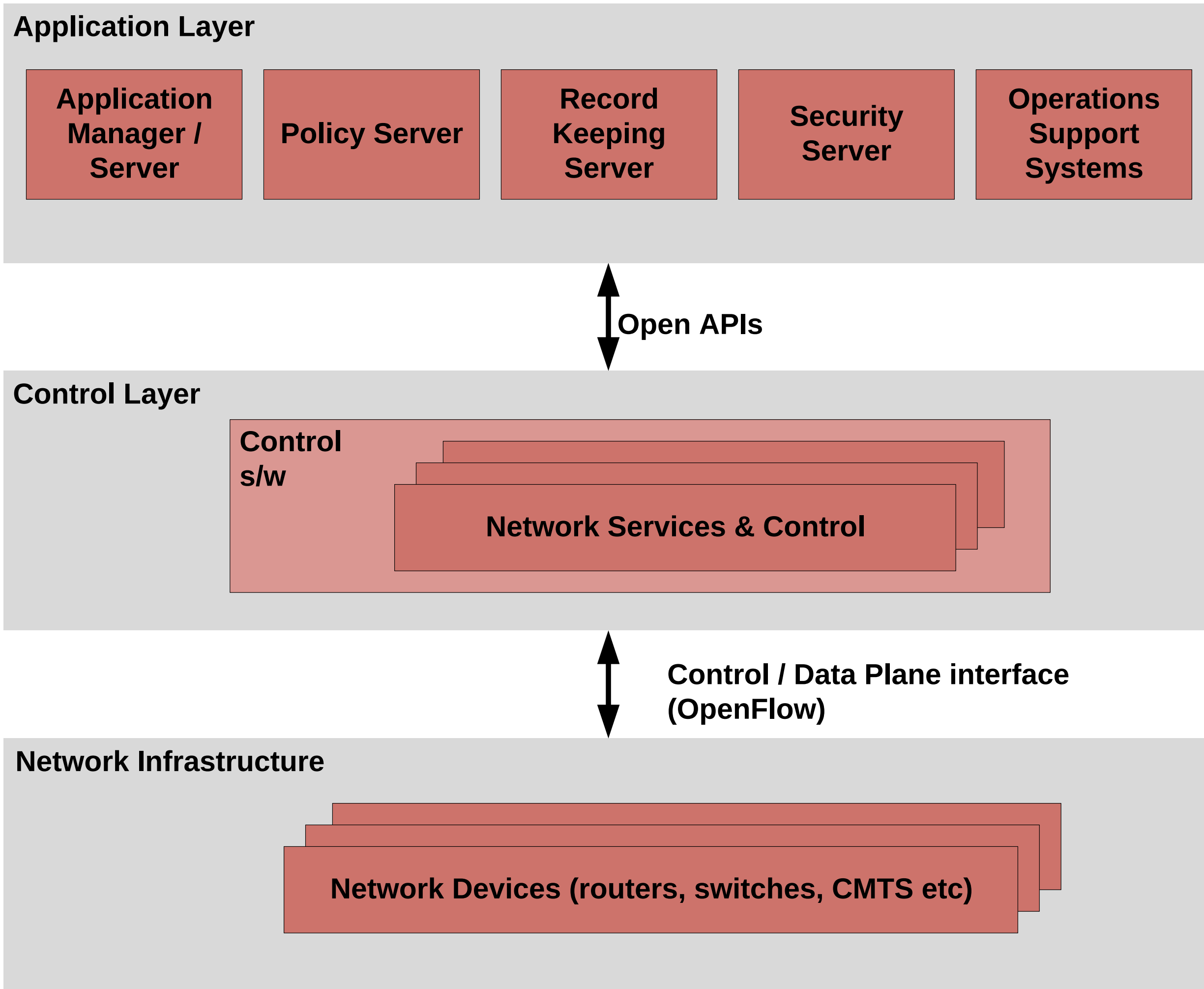

Figure 5 SDN DOCSIS Head End

CMTS

Moving the CMTS into an SDN architecture is a more complex undertaking which we will now examine in more detail.

Figure 6 shows the major components of an integrated CMTS system. These can be divided into three areas the control plane, the digital data plane and the RF data plane.

Control Plane

The control plane is typically implemented as software running on one or more general purpose CPUs and provides the following functions:

- Execution of the routing and layer 2 protocols used to create the forwarding data base
- Execution of the DOCSIS finite state machines used to control the interactions between the CMTS and the cable modems.
- Execution of the BPI+ security finite state machines used to manage the

security association between the CMTS and the cable modems.
- Interfaces to the control systems for PacketCable [PKCB] and PCMM used to establish QoS enabled sessions (shown in the application layer in Figure 5).
- Upstream bandwidth allocation and scheduling to manage the shared upstream resources
- DOCSIS specific control functions to handle CM operation and policy
- An operations and management component used for CMTS system provisioning and control.
- Additional application specific services such as subscriber management which may be present in some cases
- Slow path forwarding to handle exception packets which the data plane cannot forward e.g. ARP, DHCP.

<u>Digital Data Plane</u>

The digital data plane is the hardware based packet forwarding system and provides the following functions:

- Ethernet interfaces to connect to the regional and core networks
- The fast path forwarding engines which transfer packets between ingress and egress interfaces. Forwarding decisions are made by matching data fields in the packet header against entries in the forwarding database which has been created by the control plane. Packet header manipulation is also performed in this engine.
- The downstream QoS component implements the per flow QoS model defined in the DOCSIS specification.
- The DOCSIS MAC implements the lower layers of the DOCSIS MAC protocol in conjunction with the DOCSIS software components in the control plane. This includes DOCSIS header creation and removal and encryption / decryption functions.

<u>RF Data Plane</u>

The RF data plane provides the interface to the HFC plant:

- The downstream PHY provides conversion of digital input signals into QAM analog output, frequency conversion and RF transmission.
- The upstream PHY provides pre processing and demodulation of the received analog signals.

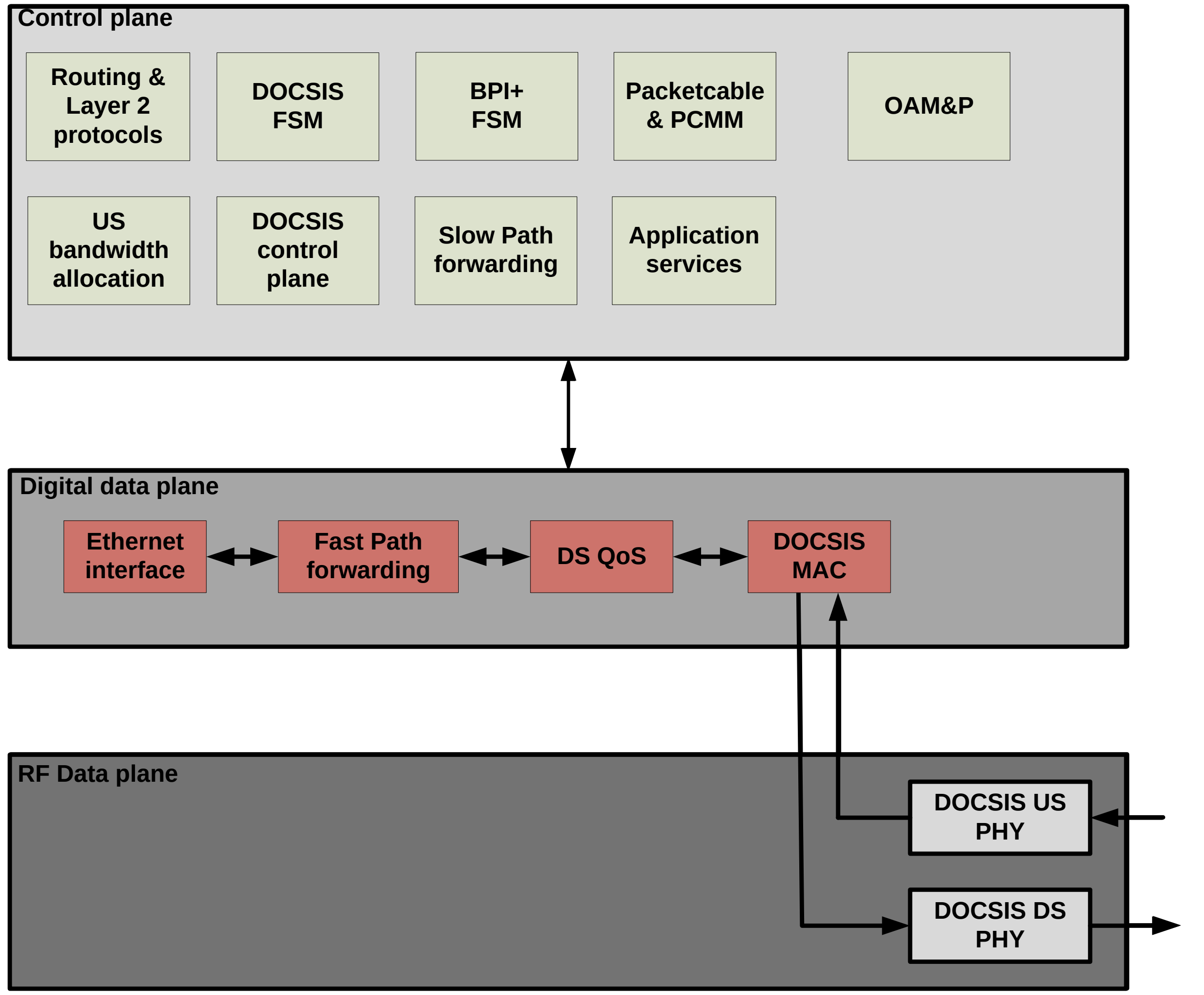

Figure 6 Integrated CMTS

A modular CMTS architecture [M-CMTS] as shown in Figure 7 is very similar except that the downstream PHY is implemented in a separate universal edge QAM and a DEPI [DEPI] control module is added to manage the interface between CMTS and UEQAM..

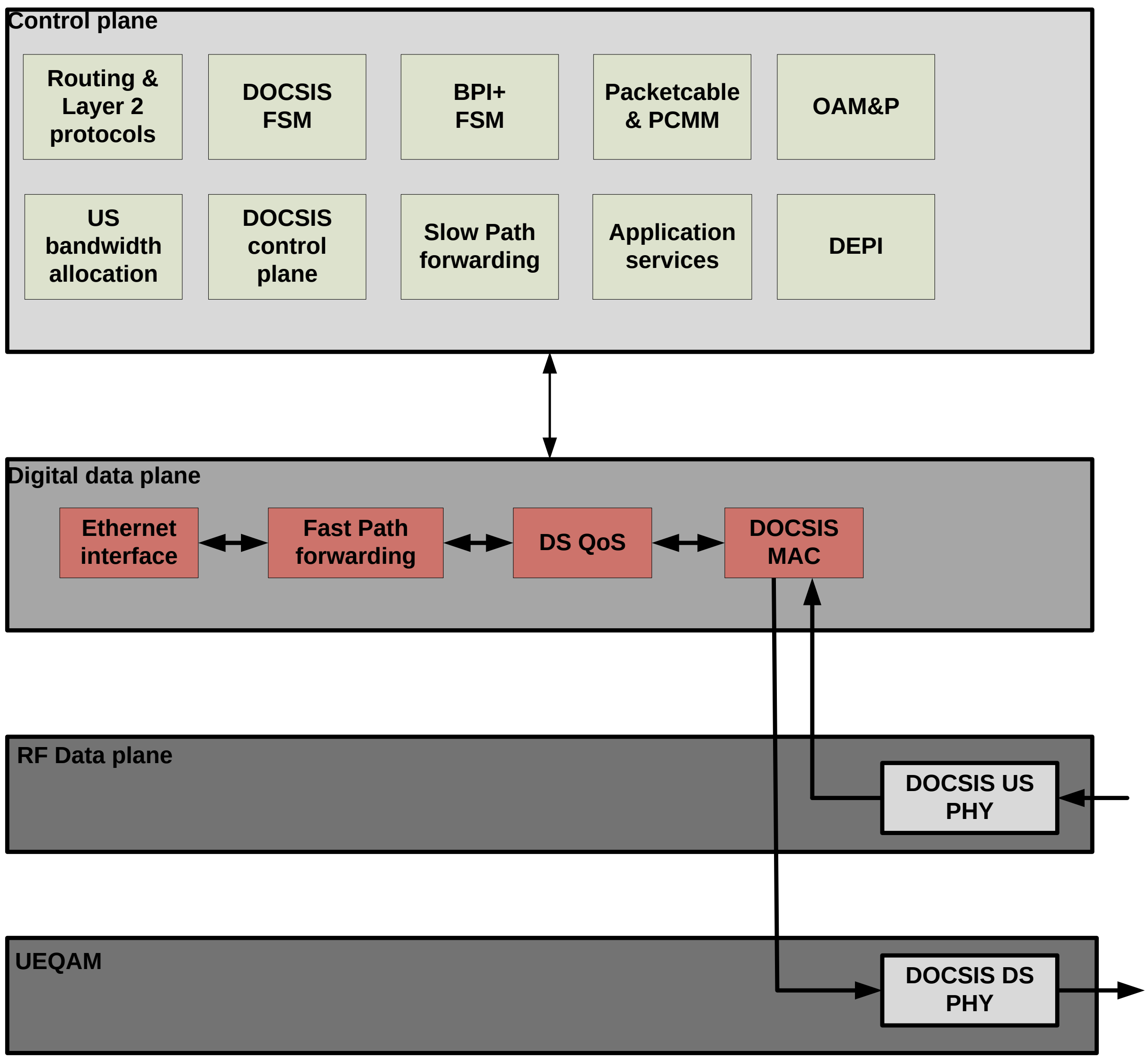

Figure 7 Modular CMTS

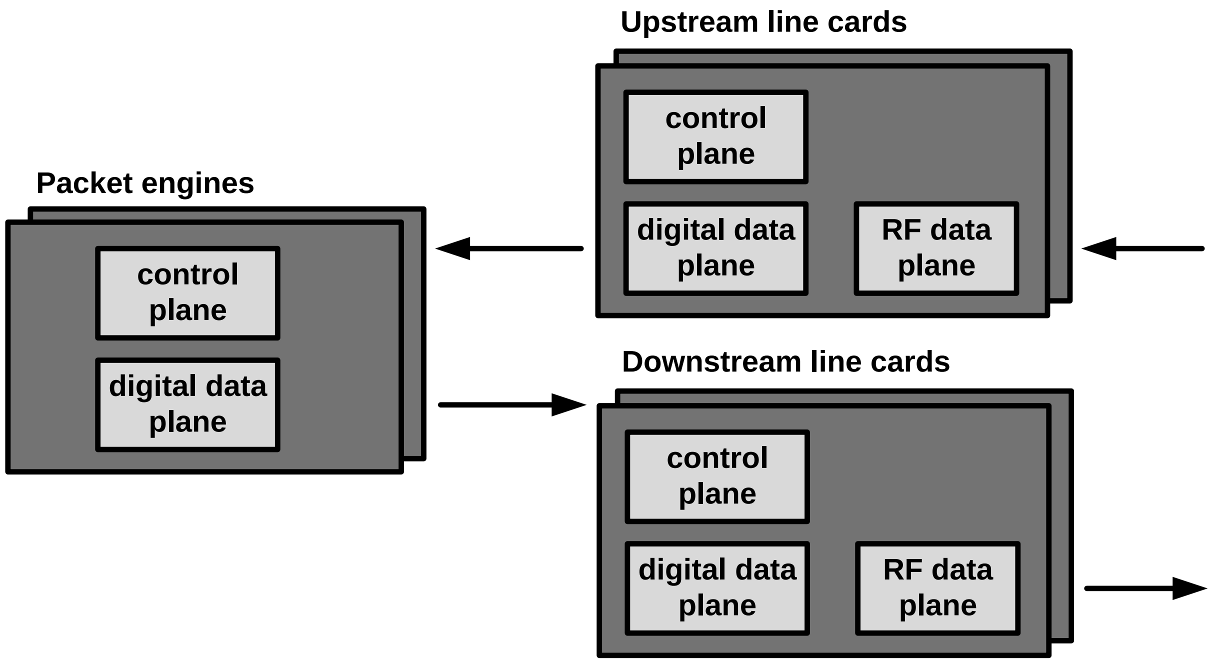

Figure 8 Typical Current CMTS Implementation

Figure 8 shows a typical CMTS implementation with the control plane distributed across all of the line cards in the system. In practice this distribution will be very uneven with the majority of the control plane software typically running in the packet engines.

SDN BASED CMTS

Figure 9 shows one option for how the CMTS can be moved to an SDN model. The control plane software is removed from the embedded control plane processors in the CMTS to the SDN control plane while the data plane of the CMTS remains essentially intact.

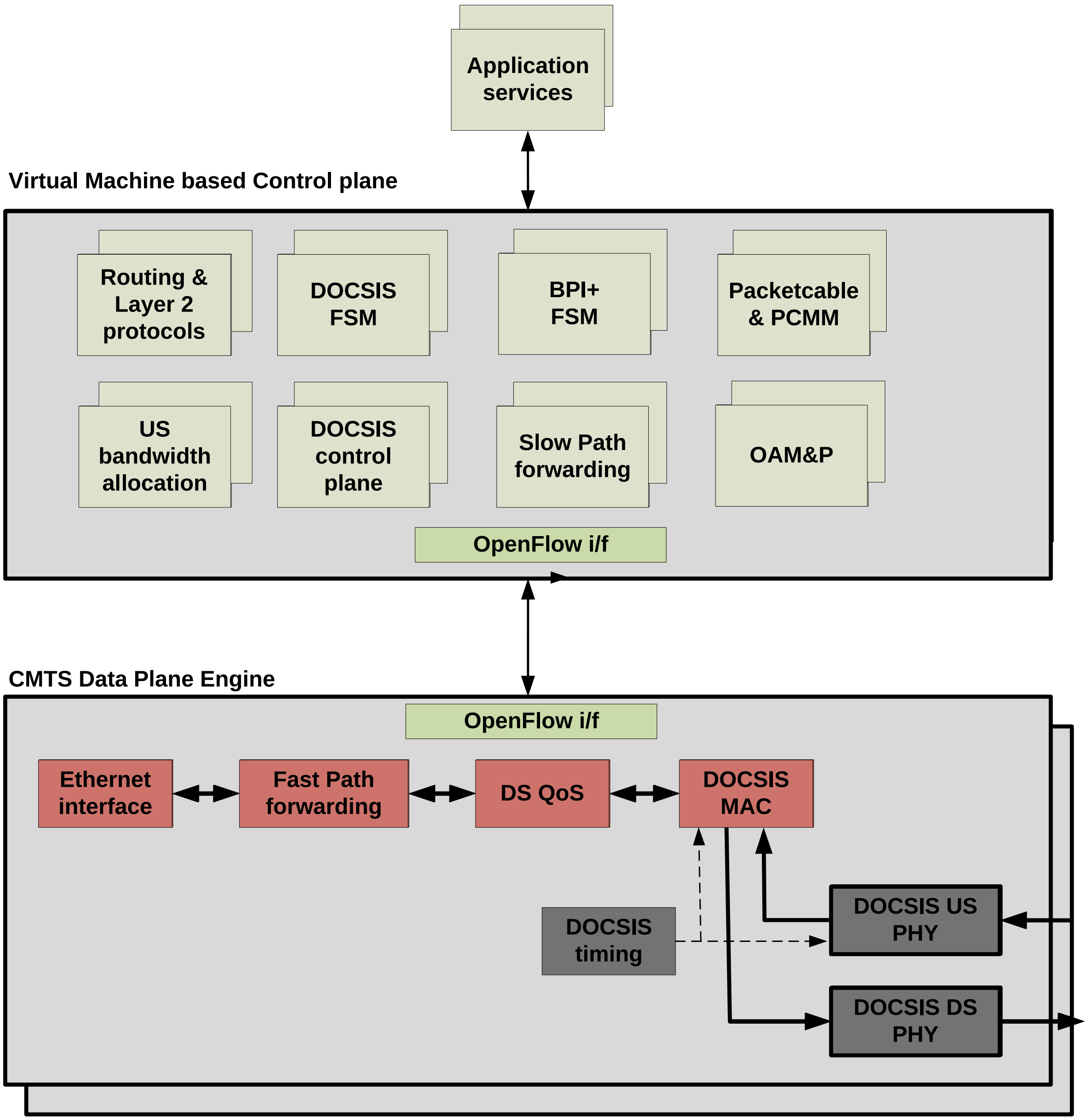

Figure 9 SDN based CMTS Phase 1

The control plane modules continue to provide the same functionality except that internal interfaces within the CMTS are replaced by messaging between the control and data planes. This would be based on the OpenFlow protocol.

- The routing and layer 2 protocols are likely be retained in an SDN system as it must interoperate with existing network devices such as the core routers. In this case they would use OpenFlow messages to update the forwarding data base in the data plane. With the move to a centralized control system it would of course be possible to construct a system in which the forwarding tables were constructed using a totally different mechanism without the need to change the OpenFlow based data plane devices.

- DOCSIS management messages continue to be exchanged between the finite state machines of the CMTS control plane and the cable modems via the data plane of the CMTS. They now incur an additional network hop between the CMTS data plane and the control plane so that some management of the added latency would be required.
- The BPI+ security finite state machines used to manage the security association between the CMTS and the cable modems will run in virtual machines in the control plane. In the SDN model the keying updates need to be passed to the encryption engines in both the CMTS and the CMs.
- PacketCable [PKCB] and PCMM systems running in the control plane VMS are used to establish QoS enabled sessions. They can continue to use the existing interfaces to the servers but could also move to more modern REST based interfaces.
- Upstream bandwidth allocation and scheduling are typically very CPU intensive. They can take advantage of the additional low cost CPU cycles available in the workstation hosted control plane. They can also be scaled up and down by adding or removing VMs as needed.
- The operations and management component used for system provisioning and control would run in the control plane. The configuration data for the data plane would need to be passed over the interface. Monitoring and alarm data need to flow in the opposite direction.
- Adding application specific services such as subscriber management becomes much more practical in the SDN environment. Additional CPU cycles are readily available on demand and the server based development environment is much friendlier than those for embedded systems.
- Packets which cannot be forwarded by the data path for any reason are handed off to software for processing. In a current CMTS this will run in one of the on board control plane processors. In the SDN case these packets will be handed off to the remote control plane for processing. The two inter system hops (data plane -> control plane -> data plane) added to the path will of course add latency but by definition 'slow path' packets, e.g. ARP or DHCP, need additional processing which adds latency so that this should not be a problem.

With this move we have taken a major step towards SDN, separation of control and data planes and centralization of the control plane functions.

Figure 10 shows The CMTS decomposition taken a step further. In this model the DOCSIS specific hardware has been separated from the IP/Ethernet packet forwarding engines and moved to a DOCSIS MAC-PHY shelf. Following this move the Ethernet interface, fast path forwarding and DS QoS modules are a very close approximation to an off the shelf Ethernet switch and we have taken a further step along the SDN path by enabling the use of COTS forwarding engines.

The DOCSIS MAC-PHY shelf is still needed to implement the DOCSIS framing and security functions at the MAC layer and for the DOCSIS PHY layer and RF functions.

DOCSIS QoS is based on a per flow model which is typically not supported in Ethernet switches. Thus if this must be supported an additional QoS module may be needed in the MAC-PHY shelf

The DOCSIS upstream and downstream links are tightly coupled so that a timing module is also required to ensure synchronization.

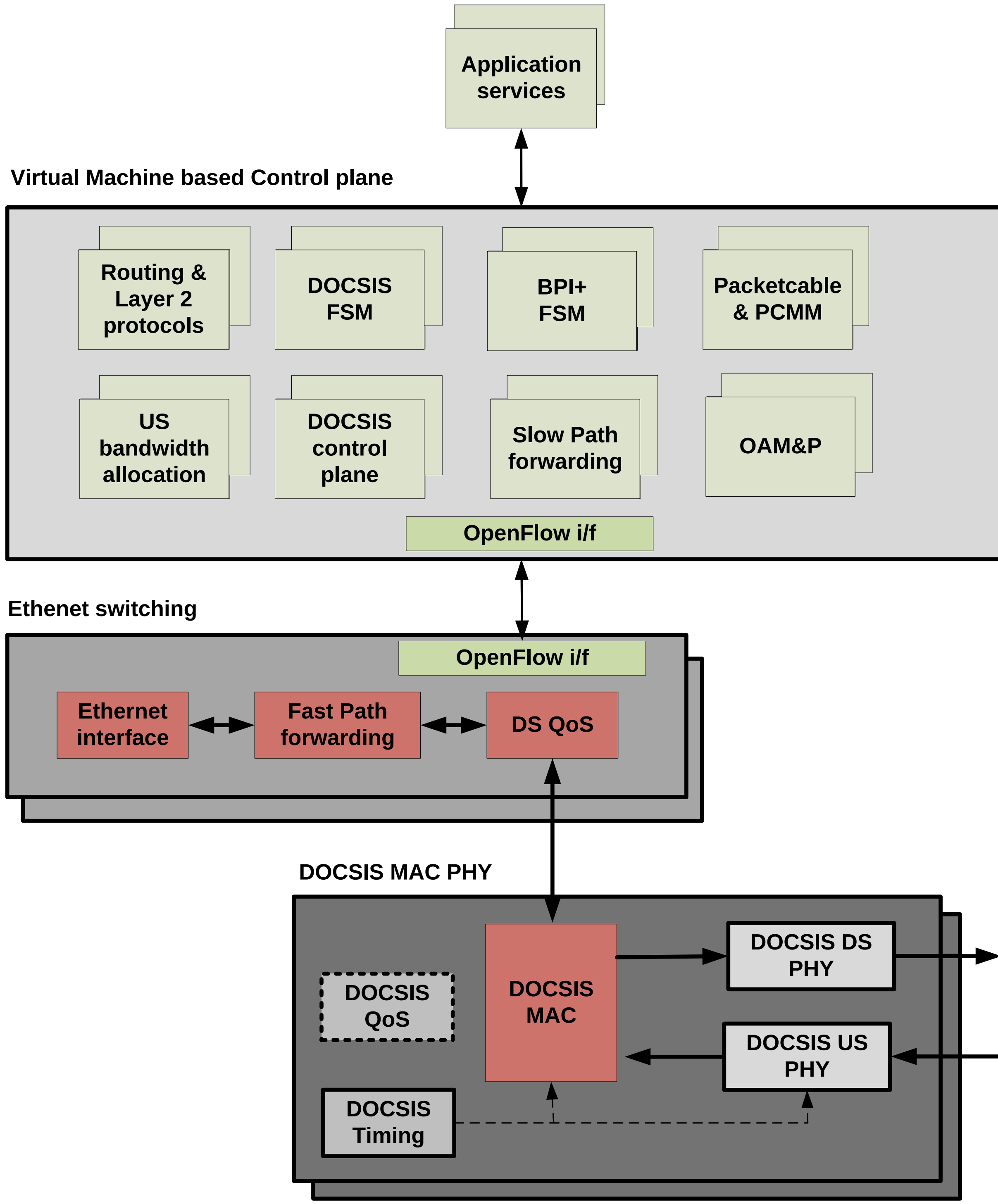

Figure 10 SDN Based CMTS Phase 2

SDN BENEFITS

As the cable infrastructure migrates to a data center architecture Software Defined Networking (SDN) can offer significant advantage in both network architecture and operation.

Centralization

The separation of control and forwarding in network devices such as routers, switches and CMTSs allows the control plane to be centralized rather than distributed across multiple devices. This has the benefit that it can provide high level APIs to application services to enable control of the network. This enables the network operation to be driven directly by the policies and needs of the applications rather than using policy to influence the operation of the distributed control plane as in the current model.

The control plane software in modern communications equipment is typically implemented as software running on top of a commercial RTOS such as Linux and executing in general purpose cpu's embedded into the system. This has the advantage of simplicity as the systems are self contained. The embedded nature of the devices does however have the following disadvantages:

- Embedded CPU cycles are very expensive compared to processing costs in general purpose servers. They also have a much slower upgrade cycle. To take advantage of newer (faster, cheaper) CPU versions these must be designed into the next generation of line cards and the deployed systems upgraded. Upgrading to new server hardware is a much simpler operation as the control plane VMs are migrated to the new server platform.
- Embedded software is more complicated to develop and test than workstation software due to inferior tool chains and more expensive test equipment. Thus development cycles are longer and costs higher. It is also more difficult to leverage third party software in an embedded system.
- The distributed control plane entities must interwork to create a coherent view of the network through complex routing protocols which need time to converge following any failures or topology changes

COTS Components

Moving to an SDN architecture enables the use of standard low cost compute platforms for the control plane and standard low cost forwarding engines to provide a significant portion of the data plane. This allows the CMTS hardware costs to benefit from the much larger scale of the enterprise market and allows multiple suppliers to participate.

Load Balancing

Horizontal scaling and redundancy through load balancing techniques is an established tool used in large scale data processing operations. The workload is distributed across multiple virtual machines which can be running on one or multiple physical servers. Figure 11 shows a simple example of multiple RTOS systems running as virtual machines with a VM hypervisor acting as arbiter to the physical hardware. Thus any failure of either software or hardware reduces the overall processing capacity but does not stop a function from executing.

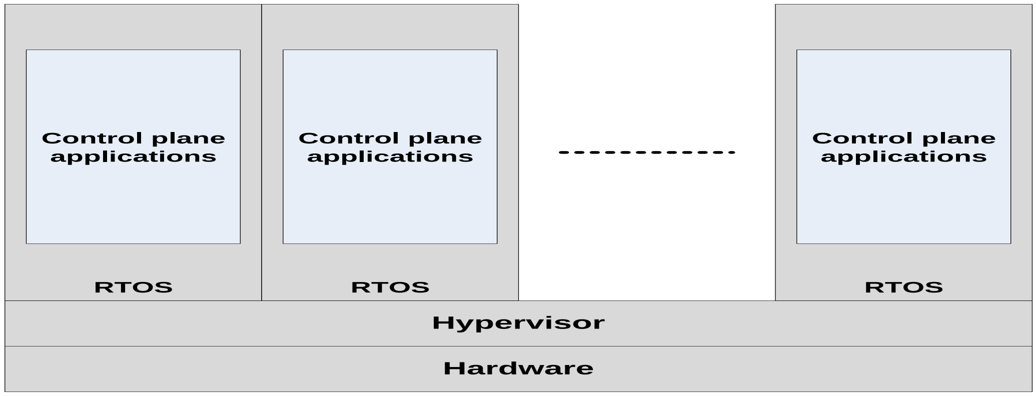

Figure 11 Control Plane Virtual Machines

Moving to this model has potential advantages over an existing CMTS. An RTOS partition can be be assigned to support a much smaller subset of the total user population than would a typical CMTS line card. A software failure would only impact this subset of users rather than all users on the line card. Essentially we use the hypervisor technology to scale the system and reduce the size of the software failure group. As the number of users in the system changes the number of VMs instantiated can be increased or decreased accordingly.

Upgrades to software are significantly simpler in the VM environment. A VM can be spun up with the new software version and tested against a subset of the users. As VMs are removed and restarted over time the upgrade can be managed and deployed from the central location. This is obviously a much simpler operation than upgrading large numbers of embedded systems distributed over multiple field locations.

Virtual Networks

SDN enables the creation of virtual networks which overlay the physical network infrastructure. The virtual network can be targeted at specific applications and manipulated via software changes rather than moving boxes and cables. It is a close analogy to the virtualization which has occurred in the server farms in data centers. The virtual network operating system is analogous to a virtual machine system such as VMWare. In a cable environment virtual networks could be created for residential high speed data, business services and managed IP video. Figure 12 shows a possible virtual network system based on OpenFlow interfaces to the Ethernet switches and DOCSIS MAC-PHY shelves

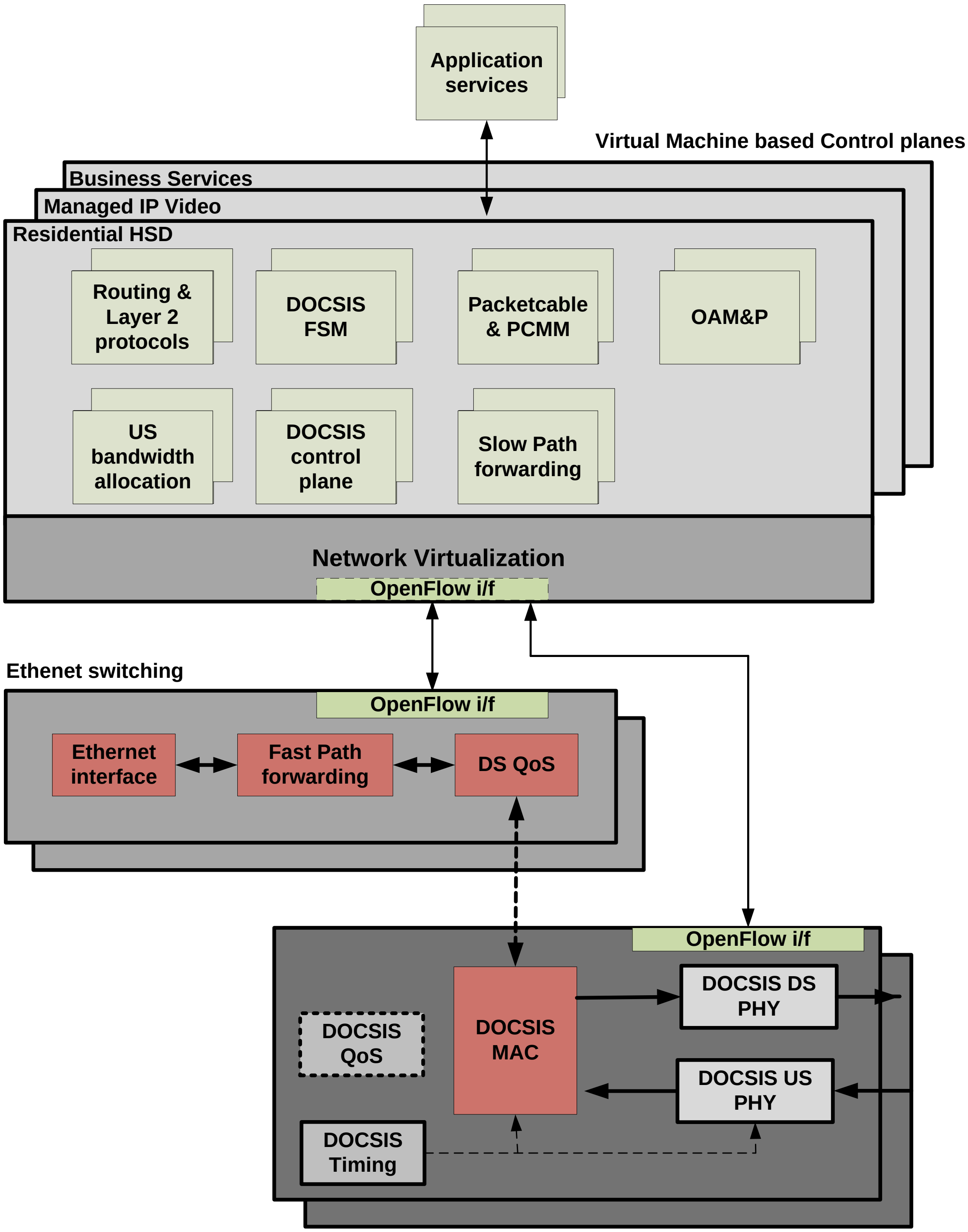

Figure 12 Virtual DOCSIS Networks

SDN PROBLEMS

Not all DOCSIS functions can be performed using off the shelf hardware components. The two suggested implementations of an SDN based CMTS Figure 9 and Figure 10 both require DOCSIS specific PHY, MAC and QoS hardware components. Thus a DOCSIS shelf will be required in addition to the servers and switches. The reasons for this are described below.

DOCSIS PHY

In the downstream direction the DOCSIS PHY hardware modulates the digital data stream into QAM signals, up-converts these to the desired frequency, conditions and delivers the RF signal to the network. The upstream PHY receives the RF signals from the cable modems and demodulates them to produce a digital byte stream. It is also responsible for power and timing measurements used in the ranging process [DOCSIS].

DOCSIS MAC

In the downstream direction the DOCSIS MAC hardware implements the lowest layers of the MAC protocol including DOCSIS header creation, packet encapsulation and content encryption. In the upstream direction decryption and header removal are performed.

DOCSIS QoS & Bonding

DOCSIS provides a very sophisticated per service flow QoS model enabling services such as voice and video to be delivered with guaranteed quality. The downstream QoS engine is more complex than the QoS available in a standard Ethernet switch so must be applied in the DOCSIS MAC-PHY shelf. With DOCSIS 3.0 channel bonding downstream QoS is even more complex as a packet may be sent on any one of multiple channels.

In the upstream QoS is provided by the upstream scheduling software running in the SDN control plane which allocates upstream bandwidth between requesting cable modems. For this to operate effectively it should not introduce significant added latency to the process, so that in practice this function may need to be split between entities in the DOCSIS shelf and the VM control plane.

DOCSIS Timing

DOCSIS requires that upstream and downstream links are synchronized to a common clock to enable efficient demodulation of the upstream traffic bursts. This timing function is restricted to the components in the DOCSIS MAC-PHY shelf and does not propagate to Ethernet switches or the control plane.

DOCSIS CHANGES

As shown above DOCSIS can be implemented in an SDN architecture but will still require some specialized hardware. Clearly some of this hardware such as the PHY components will always be needed but it is worth examining the protocol itself to see if it could be made more SDN friendly.

QoS

DOCSIS is primarily used to deliver Ethernet packets from servers to clients in a home network via HTTP connections. These packets travel over best effort (or possibly priority based) networks from the source to the CMTS and from the cable modem to the client with DOCSIS QoS being applied over the CMTS to CM link. Moving to a much simpler priority based QoS model would enable standard Ethernet switches to be used and remove the need for the downstream QoS hardware. This would potentially reduce the quality of PSTN style voice services but would still support VoIP and streaming video delivery, both of which currently operate successfully over cable networks as best effort services from OTT providers.

Bonding

DOCSIS 3.0 channel bonding allows multiple DOCSIS channels to be combined to offer a higher speed downstream service to a cable modem. Packets sent over the bonded link can arrive out of order so that sequence numbers must be added and checked. A bonding component is required on the MAC-PHY shelf to implement this. A DOCSIS bonding group can be created from any number of channels (up to the maximum supported) and multiple bonding groups can overlap almost arbitrarily. This makes packet scheduling a complex operation. If bonding groups were restricted to set numbers of channels e.g. 2,4,8,…) and constrained to be hierarchical this could simplify the bonding operation without a major loss of functionality and reduce the complexity of the MAC-PHY shelf.

Payload Header Suppression

PHS is a compression scheme introduced for the low bandwidth channels available in early DOCSIS deployments. With the move to much wider channels in the post DOCSIS 3.0 era this is rarely used and could be removed to simplify both embedded and SDN based implementations with minimal impact.

CONCLUSION

As the cable infrastructure migrates to a data center architecture Software Defined Networking (SDN) has the potential to have a major impact on network architecture and operation.

It can provide the following advantages over a traditional CMTS deployment:

- Separation of control and forwarding in network devices such as routers, switches and CMTS.
- The use of standard low cost compute platforms for the control plane
- The use of standard low cost forwarding engines to provide the data plane
- Centralization of control decisions as opposed to the decentralized model used by current network hardware
- Orchestration of resources across layers and domains for optimal performance
- Horizontal scaling and redundancy through load balancing techniques
- Provide general purpose interfaces to hide details of network elements from upper layer protocols and applications enabling simpler application development.
- The ability to provide virtual networks under the control of centralized applications.
- Remote operation with the control plane running in a cloud environment or super head end complex.

A number of DOCSIS protocol changes could make an SDN implementation simpler and lower cost. These should be considered for a future version of the protocol.

These advantages combine to offer the potential for lower capital and operational costs.

<u>REFERENCES</u>

[CCAP]	J. Salinger, "Proposed Next Generation Cable Access Network Architecture", SCTE Conference on Emerging Technology, 2009.
[TRANS]	Gerry White Transforming Cable Infrastructure Into a Cloud Environment, NCTA 2012 Spring Technical Forum
[HFCDFC]	M. Emmendorfer, S. Shape, T. Cloonan & Z. Maricevic Examining HFC and DFC (Digital Fiber Coax) Access Architectures, SCTE Cable -TEC Expo 2011
[M-CMTS]	Cablelabs DOCSIS® Specifications — Modular Headend Architecture (MHA)
[OPENF]	www.openflow.org
[DOCSIS]	Cablelabs DOCSIS® Specifications - www.cablelabs.com/cablemodem/specifications/index.html
[PCMM]	Cablelabs DOCSIS® Specifications - www.cablelabs.com/packetcable/specifications/multimedia.html
[PKCB]	Cablelabs DOCSIS® Specifications - www.cablelabs.com/packetcable/specifications/index.html l
[DEPI]	Cablelabs DOCSIS® Specifications - www.cablelabs.com/cablemodem/specifications/index.html

ABBREVIATIONS AND ACRONYMS

API	Application Programming Interface
CCAP	Converged Cable Access Platform
CDN	Content Delivery Network
CM	Cable Modem
CMTS	DOCSIS Cable Modem Termination System
COTS	Commercial Off The Shelf
CPE	Customer Premise Equipment
CPU	Central Processing Unit
DOCSIS	Data over Cable Service Interface Specification
FSM	Finite State Machine
Gbps	Gigabit per second
HFC	Hybrid Fiber Coaxial system
HSD	High Speed Data; broadband data service
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
MAC	Media Access Control (layer)
Mbps	Megabit per second
MSO	Multiple System Operator
OTT	Over The Top
PHY	Physical (layer)
PMD	Physical Medium Dependent (layer)
REST	Representational State Transfer
RF	Radio Frequency
RTOS	Real Time Operating System
SDN	Software Defined Networking
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VM	Virtual Machine
VoIP	Voice over IP