Implement Closed-Loop Network Decisioning Now with Big Data Analytics and Fuel Future-State SDN Use Cases Through a Common Platform Deployment

Brennen Lynch and Anukool Lakhina
Guavus, Inc.

Abstract

Communication Service Providers (CSPs) are finding an increased reliance on their IP networks by consumers from a service perspective. As consumers become more dependent on IP services and thus bandwidth demand grows, CSPs require dynamic, stateful tools to optimize network operation/QoS and monetize the streaming data generated by the network. CSPs also have the opportunity to incorporate potential consumer-based revenue streams by targeting elastic, innovative, and potentially disruptive services and technologies.

Operational virtualization can occur with systems that are present within the network now. CSPs have the opportunity to become early adopters of these closed-loop use cases and roll out incrementally, starting with solutions that can be supported by current network conditions and placing their business in a strategic position to exploit SDN in a timely manner. CSPs that implement a standardized data ingest/processing/fusion platform can fuel use cases that are supported by current network conditions (such as real-time routing decisions). As the industry and backbone/access networks move towards complete virtualization through SDN and NFV techniques, the common platform can ingest additional, relevant data sources (such as hypervisor logs) to power coherent virtualized, automated, closed-loop decision processes.

This paper explores platform integration and techniques that can be implemented in the network now, which place operators in a prime position to exploit fully virtualized control systems as they become incorporated. Use cases from an operator's network perspective (traffic optimization) as well as a consumer's perspective (dynamic service delivery) are explored and demonstrated through experimental investigation of both current network and fully SDN-enabled test-environment scenarios.

Transition towards dynamic resource allocation

Code-line mediation of network infrastructure generally requires physical intervention at the hardware site. This highly inflexible way of mediating the network is not congruent with the way that current services are becoming virtualized. No longer does the ISP or content provider connect directly to the subscriber to deliver media. Intermediary content delivery networks (CDNs), placed near to the subscriber, cache and relay content to reduce latency and increase the quality experienced by the subscriber. Datacenter operations are in many cases have transient requirements; significant hardware/CPU optimization can occur if these operations are virtualized and implemented, when and where they are required.

Network analytics can extract highly valuable information and insights. However, there must be integration within the network to allow for these

insights to drive operations. Devices currently make stateless decisions based on the traffic they carry, and not based on the network's entire state. Resource allocation is based on fixed presets, rather than network conditions and demands. Large and complex networks simply compound the problem of manual intervention within network settings – a tedious and error-prone process.

Virtualization of networks is at odds with centralized functions/services, yet does centralize control functions. As physical servers are virtualized and operate under dynamic provisioning, the innovation bottleneck is now placed within network provisioning. New protocols and services require tedious manual changes of the software within network devices. Enabling a centralized control plane and providing separation from the forwarding functions offers the ability to dynamically program the behavior of the network using well-defined interfaces. This controller can be deployed as a cluster for high availability and scalability. These are the core facets of Software-Defined Networking (SDN) implementation from a functional perspective. Network Function Virtualization (NFV), developed by service providers, is a complementary concept allowing for relocation of network functions (such as NAT, firewalling, DNS, intrusion detection, caching, etc.) from dedicated appliances to generic servers, enabling dynamic resource allocation.

Analytics is the brain behind a highly optimized and efficient network. Traditional 'store-then-analyze' paradigms are simply too latent to fuel real-time decisioning in the face of petabyte-scale streaming data. By implementing a distributed-compute model, where known end-actions are translated into pre-processing algorithms at the collector state, an operator can effectively power real-time processes that are dependent on such fusion. Effective integration of SDN will require a specialized, *big data fabric* to provide stateful decision-making, as these inveterate & transient functions are conditional on real-time processing of network data (as well as fusion to appropriate reference datasets).

Architecture & State of Integration

Software Defined Networking

At its core, SDN provides a construct for centralizing the control plane (typically layer 2/3 functions). This controller provides a logical representation of the network by abstracting the network topology and the network elements. It 'replaces element-based, box-at-a-time configuration with network control; that is, instead of hop-by-hop control, it provides end-to-end control' [17]. By providing a centralized stateful controller, applications can then manipulate control functions dynamically via northbound APIs. Through such orchestration applications, SDN can automate service chaining for service functions in higher-order layers (3+) – a boon for minimizing deployment disruptions when implementing SDN functions.

Southbound protocols (such as standards-based OpenFlow) provide the controller access to the data plane devices - whether these devices are physical or virtual. By standardizing the control protocols, different vendor devices can be easily rendered as

interoperable. This ease of integration is hugely supportive of industry innovation, as the control plane can ensure interoperability on a virtual basis – no longer do entire software stacks need to be updated in support of a new service (imagine if every time you wanted to download another app on your smartphone, a new operating system update was required!).

A non-disruptive integration of SDN towards a wholly device-based implementation (requiring SDN-compatible devices) must support virtual switches to communicate with devices, as well as provide support for bare-metal hosts such as datacenter servers. The figure below details this hybrid architecture.

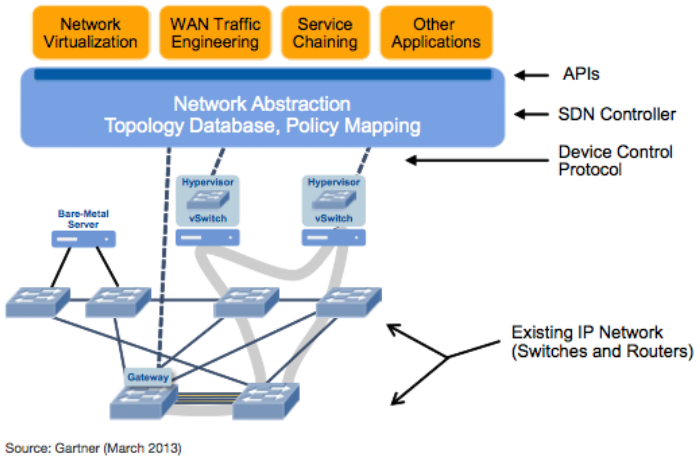

Figure 1: hybrid architecture of SDN + bare metal [17]

Network Function Virtualization

Since standard hardware – multicore processers – have made exponential gains in their processing power, network functions can be embedded as software agents in these standard processors, as opposed to the traditional implementation of a unique appliance (consuming a fixed amount of resources at a fixed location). These traditional implementations could span from centralized-service data centers, or highly distributed appliances (such as video caching). Virtualizing these functions with standardized hardware (i.e. x86) transitions from the traditionally unique (& costly) implementations and supports one single, highly efficient investment of an operational model. NFV breaks the traditional linkage between IP location and identity. This is a very similar model to how modern data centers operate, and is highly complementary to SDN techniques that abstract the control plane from the data plane.

State of Integration within Service Providers

Hybrid architectures will be implemented in order to maintain a seamless transition over time to a fully SDN architecture. Traffic forwarding must be differentiated into centralized controller forwarding (i.e. via OpenFlow) or forwarding via conventional means. The Converged

Cable Access Platform (CCAP) provides a transitional model towards virtualized control, by combining the eQAM for digital video with the CMTS for data into one device (or logical entity, if virtualized). The key CCAP feature towards SDN techniques is that it provides a control/management plane (for IPDR, DOCSIS, RF, QAM MIBS PCMM, & edge resource management). Decentralizing control from the headend in a virtualized CCAP environment would allow these compute processes to occur in the most efficient location as determined by the controller. The flow table can communicate with the controller to install flow entries based on load balancing, failover, traffic control, service prioritization, etc. Implementing real-time data analysis to support this controller communication with the forwarding plane will ultimately enable dynamic resource allocation and drive key network optimization techniques.

Big Data Fabric; Real-time Processing

Use cases supporting SDN techniques will require an underlying data fabric that supports real-time analysis and supports closed-loop, automated processes. A loosely coupled, component based architecture is the basis to this platform. Furthermore, the collection architecture should be intelligently structured with edge-based processing components. At the petabyte-scale of high velocity network data, collection must have the ability to pull data from the edge through the backbone network in real time without taxing mission-critical infrastructure. Furthermore, the network data needs to be dynamically fused and correlated with static & reference data sets to produce key causal relations. Thus, a departure from the traditional 'store-then-analyze' approach must be explored in order to move from reporting-based BI platforms to one that supports real time, closed loop processes. Although this paper is not a detailed investigation of the specific platform components, a high level view will be presented of the overall structure.

Basing the platform on open-source architecture is important to maintain interoperability standards. However, radical extensions of traditional open-source platforms are required to optimize for specific use cases (low latency, high cardinality, compute requirements, etc.). A highly available, 'compute-first' architecture that is based on the open source Apache Spark project is the cornerstone of this *new data fabric* that supports this paper's use case explorations. By integrating batch analytics, real-time analytics, and iterative machine learning into a single stack, significant latency gains can be made over the Apache Hadoop's Map-Reduce implementation (also significant development primitives over map-reduce). At the point of collection, data can be normalized and fused by using agent-based software components that are optimized for low-latency, high-throughput transactions.

Using Apache Spark as the base, extensions can be incorporated into a single stack to support high cardinality use cases, complex event processing and machine learning iterative algorithms, as well as fast query processing (outside of structure workflows) and mutable data-store integration. Furthermore, some of these platform components *must be virtualized* in order to effectively support SDN use cases. The compute engine supports stream processing via a rules

engine & analytics based on state variations, batch processing through cubes representation, & iterative processes through machine learning engines. Apache Spark has great support for machine learning through its Mlib library (which leverages Scalanlp's Breeze and JBlas), and graph analysis through its GraphX library. Shark, a distributed-query engine implemented over Spark, can support low-latency discovery use cases over data that resides in Hadoop's Distributed File System. Through cluster management (ie YARN), multiple applications and workloads can run on a single cluster, optimizing resource allocation and providing cluster resilience. The processed data can then be presented for visualization via in-memory caching & disk storage using columnar compression, or can be used as an outbound action trigger through a message broker framework with standardized adaptors. This could easily communicate with any variety of SDN controllers such as the open-source OpenDaylight, or vendor-specific such as Juniper's Contrail. The virtualization of this platform to support efficient SDN use cases could leverage QEMU/Openstack. [20]

Use Cases for Fixed-line Networks

Access Network

By deploying a platform with the ability to provide real-time correlated outputs, an operator can begin to drive virtual functions that support closed-loop use cases around optimization, routing/caching, security, and more. This also allows for continual visibility of VNF (virtual network function) & VRF (virtual routing & forwarding) implementations.

The data sources required vary by use case, but can be segregated into the network-based data, and contextual datasets.

Network data:

- *IPDR/SP*: IP detail record resolving traffic flow, QoS metrics to network topology (cable modem & CMTS). IPDR is already enabled throughout the entire network through DOCSIS.
- *DPI*: deep packet inspection involving probe-based implementations at the router to process each packet & output relevant data, such as the user agent to define device, or the content/application accessed.
- *Flow (i.e. OpenFlow / NetFlow)*: traffic flow structured by the matching packets to the traditional 5-tuple of ingress interface, source/destination IP, IP protocol, source/destination port for TCP/UDP, IP Type of Service
- *DNS*: domain name system provides IP-to-URL translation and, with a given IP, provides visibility into content & application.
- *Router configs/SNMP*: gives interface level details for capacity, verification, etc.
- *Hypervisors*: such virtual topology information is contained within the hypervisor logs, such as virtual-to-physical mapping, routing paths, and bandwidth reservation state. This could be in a distributed environment to achieve scalability [3].
- Other configuration data: application IPs/ports, site IP ranges, etc.

QoE-driven Optimization; Content Routing & Caching; CDN

As consumers are increasingly reliant upon IP delivery of content, QoE-driven mechanisms are important to ensure a high quality to the end user, as well as optimize the use of the access network. Content caching through CDNs have been implemented for some time now as a method to reduce core traffic and to increase the quality of service to the consumer. Traditional engineering would build fixed-appliance caches on expected peaks, which ultimately means that much of the caching capacity would be unused for a majority of the time. These typical deployments are in-line in order to capture and respond to http requests (thereby forced to examine all traffic, whether it can be served by the cached server or not). SDN techniques allow CSPs to deploy servers out-of-line, directing specific flows to the caching servers. This results in an increase of the overall scalability and reliability of the solution. In addition, specific targeting of sites based on caching can occur.

Figure 2: example visualization of services usage, classified by content category. This gives visibility towards any caching VNFs implemented on a per-content basis, and parameters such as by utilization or WAN cost allow for operational oversight. Furthermore the sites/site-pairs could be exposed to view top demand generators for each service.

When services can be deployed close to the end user, latency and overall network efficiency benefit. Stateful decisions must be made; in other words, considering not just distance, but also latency, jitter, and available bandwidth. By correlating streaming inputs of link performance, latency of services, delay, & jitter, an SDN controller function can spin up or spin down services by directing flows to the caching servers. Although this can be structured on the bit-level, it is much more valuable to resolve these flows into content, in order to aggregate and optimize by content or service. This level of granularity can be obtained in two ways: (1) through some level of packet discovery via DPI, which can either be full-census or via intelligent extrapolation of flows based on sample-census DPI, or (2) by resolving NetFlow logs to DNS queries on a real-time basis (requiring low-latency compute operations). Controllers can then obtain latency information from the routers to ultimately build quantitative analytics and even predictive; to ultimately program the network itself in response to routing in certain conditions or *anticipated* conditions. Although the mechanisms of controller-based VRF is out of scope of

this paper, the controller could act at the router level by dynamically implementing distributed, virtual publish/subscribe systems in the Ternary Content-Addressable Memory (TCAM) thus dynamically programming the access control lists (ACLs) [2]. Financial information can also be incorporated to bring peering costs via leased links/routers into context [12]. An MSO could use GSLB (global server load balancing) to direct traffic according to these criteria, and interface with PCMM PS (the PacketCable Multimedia Policy Server) to provide some level of DOCSIS QoS assurance. From a video optimization standpoint, business-contextual data around weather, large events i.e. Superbowl, etc. can be integrated into the dynamic caching/routing algorithm. The operational metrics associated with demand generators and VNF functions are shown in figure 2 through an example user interface.

Traffic Optimization

Current implementations of traffic forwarding (in DOCSIS networks) see the CMTS performing routing functions. through routing protocols and by providing a default subscriber gateway, the CMTS routes traffic from subscriber devices to aggregation routers, which in turn route traffic across core network to the Internet or CSP servers. With the converged cable access platform (CCAP) providing the transitional bridge from physical hard-wired appliances to virtualized network functions, controller interfaces that rely upon real-time analysis of data can perform optimization-based functions. The CMTS/CCAP can serve as a layer 2/3 device by interfacing its Flow Table with the controller, providing dynamic forwarding function as well as aggregate flow processing (matching multiple traffic types and/or destinations). Once these distributed-compute algorithms are virtually implemented, they can operate on a closed-loop basis to support such VRF functions (see figure 3). Implementation of local DNS caching can also take advantage of this analytics engine to drive traffic optimization, reducing response time to the subscriber as well as reducing traffic on the core network (minimizing centralized server queries).

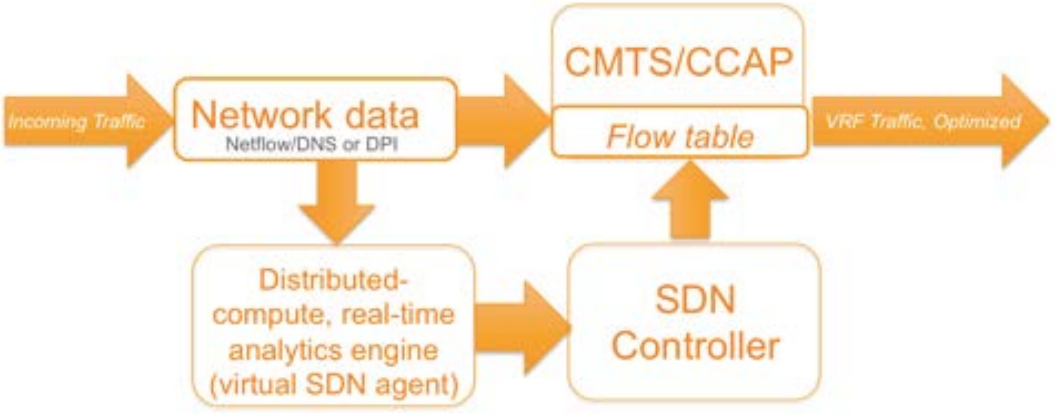

Figure 3: example representation of closed-loop VRF driven by a distributed-compute, real-time analytics engine

By learning the network topology

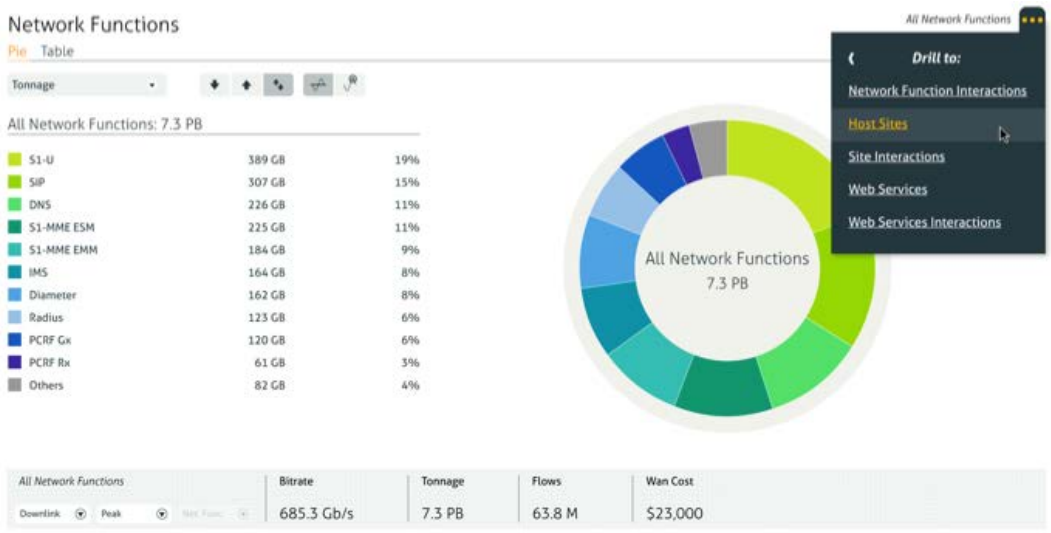

Figure 4: example of a centralized UI that provides visibility towards how network functions are operating. Executive oversight is enabled through key drill downs to host sites, site interactions, & web services/interactions.

Network Function Visibility; SLA Monitoring

Visibility towards how VNFs are operating is a key executive use case to allow for ongoing monitoring of their interactions with the network (see figure 4 for example UI). A site-to-site breakdown of traffic supports site visibility. For network functions, one could centralize in a UI the app usage for the network by data center (to satellite site,..), the applications by bitrate/tonnage/flows, WAN cost, & utilization. Furthermore one could drill down to view top consuming sites for a given application, or drill down to sites and site-pairs to see top demand generators. SLA monitoring can occur by visibility towards performance metrics; views by links (by delay/packet loss/jitter), or drill into top applications and cloud services on impacted links. Alerts can be configured via application performance to inform the stakeholder of any anomalistic behavior. This type of executive oversight enables operational assurances.

Framework for other use cases

As diverse as software functions are, there are many use cases that can be enabled through the real-time analysis of data. This paper will mention a few more use cases at high level, but this is not meant to be exhaustive. Security use cases can support Lawful Intercept (virtualized vs. at the CMTS), and enable stateful firewall decisions that can interact with packet flow through the Flow Table and can be instantiated on a virtual basis. Or, support for distributed CG-NAT or dynamic encapsulation of VPNs directly from the cable modem [5]. Furthermore, targeted billing services can be implemented on a virtual basis with a high level of accuracy.

Datacenter Operations

Datacenters are lacking an integrated view across infrastructure, applications, and users – such an integrated view is required on a continuous basis to trigger timely operational decisions. User traffic will generally span many datacenters & network functions. Traditionally these must be constructed via prediction – calculating the expected traffic at peak and engineering to support that. This leads to inefficiencies during non-peak hours, as well as potential bottlenecks if traffic is exceedingly high. By understanding how these network functions are operating in real-time, a provider can route traffic and optimize these functions via a software-based controller. This capability is highly dependent on analyzing these network logs and fusing them with contextual reference data (for example, fusing with graph-based representations of topology via hypervisor logs), and having the ability to do so in real-time.

The interface of the physical and virtual domains is key to survivable, scalable, operable data centers. In today's datacenters, physical and virtual assets are largely silo-ed and very cumbersome to fuse even on a reporting/historical basis. By integrating a seamless end-to-end view, previously un-targeted use cases can be exploited such as real-time security or root cause analysis, as well as

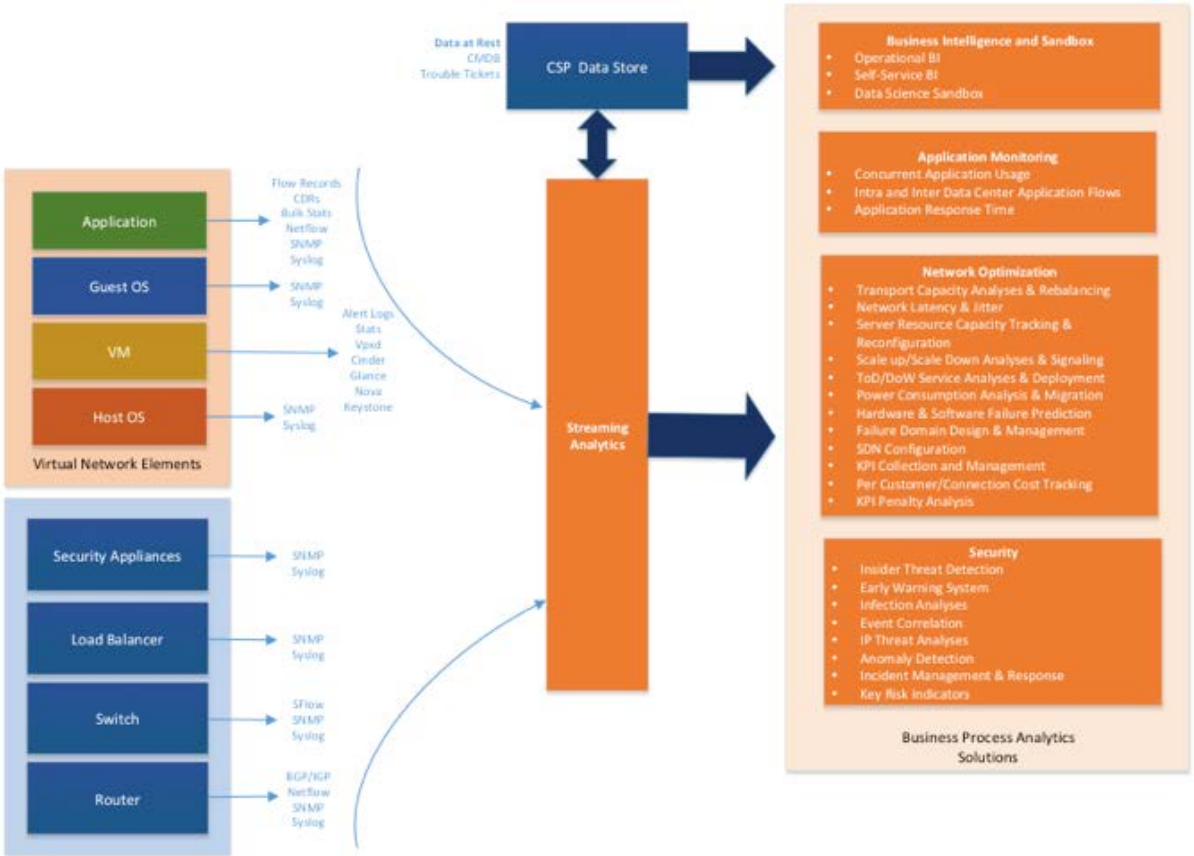

Figure 5: reference architecture for DCOI. Logs are collected directly; VM data pulled directly from orchestration systems. Flow, BGP collected from transport infrastructure.

power VNFs around dynamic resource allocation. Closed-loop signaling is used to automate critical response to issues that are discovered. From an executive standpoint, business leaders can implement assurances around security and operational optimization, as well as identifying shadow IT.

By integrating closed-loop architecture, intelligent edge processing and fusion supports highly valuable use cases dependent on real-time analysis. A sample architecture would collect the necessary data from their points of origin; illustrated in figure 5**.** Network elements forward traffic & polling data such as syslog from the application or OS; SNMP, and flow. VM data is collected from the elements and orchestration via SDKs (software development kits; API plug-in libraries), and server data is collected from the management systems (iLO, SEL,..). Static data such as customer information and semi-static such as DNS and trouble ticketing is collected. As additional data centers are brought online, they can be easily integrated within the analytics framework to create a single, unified view.

Resource Utilization - Capacity Planning; Reduce WAN Costs

Business users are always searching for ways to optimize capacity planning, such as reducing WAN costs, deferring data center link upgrades, etc. By integrating NFV, the optimal data center can be identified and the service deployed in a virtual instance. Data center sites can be identified through ingesting data provided by SDKs; in a virtualized instance. Fusing this utilization-based data with the context of WAN costs, along with link-based metrics from the access network (latency, hop count, etc.), will drive a closed-loop optimization application (a VNF). This specialized VNF can furthermore integrate with forwarding & routing decisions to optimize the access network flows dynamically (via software controller interacting directly with the Flow Table in the CMTS/CCAP). This VNF could be monitored by centralizing the real-time operational metrics into a user interface (see figure 6). Redeployment of services reduces the WAN costs. Rehoming of traffic also defers link upgrades and application licenses could be reduced.

Figure 6: example visualization of site-based traffic. Site-to-site interactions give visibility towards how DCOI-based resource allocation VNFs are operating, such as those driving WAN-based network optimization.

The ability to forecast resource consumption as well as respond in real-

time to utilization conditions will drive efficiency-based processes. By collecting flow data from the physical domain that fuses with server, VM, & storage data, one can create support resource optimization VNFs on a real-time basis and present a seamless view of resource utilization from the data center gateway to the application (see figure 7). Furthermore, data center resources can be forecasted per application based on historical trends and rules-based engines (operating via key indicators). By including financial-based data around energy, VMs can be instantiated based on minimizing cost as well as delivered performance [1]. This allocation algorithm interacts with the centralized controller to manage the servers and switches across the logically virtualized data centers [3]. Ultimately this will reduce the network downtime, radically improve root cause analysis, and improve end-user QoS.

Application Optimization
Similar to resource optimization, but extended to manage L3 and above application traffic. This could involve applying application-based policies (including QoS assurance), as well as defining the deployment, user, and server contexts. For high-value applications, specific SLAs around QoS might be in place, involving a cost model of both revenue and penalty [21]. KPIs around CPU/memory utilization, disk i/o, storage latency, switch/firewall/link/load balancer performance & utilization, etc. can be implemented to alert or trigger automated actions once they are surpassed. Or, based on application on-demand (i.e. data backup or replication scenario), network bandwidth can be dynamically allocated for backhaul connectivity or inter-datacenter links.

Efficient utilization of resources to support these applications must occur under a fluctuating demand and unpredictable failures. This requires a dynamic adaptive mechanism in the form of a VNF that is driven by real-time analysis of the physical- and virtual-domain data.

Cognitive Computing
With growing data-center demands at over 60% per year, the infrastructure needs to adapt dynamically. Real-time granular detail on response times, arrival rates, & concurrency for capacity planning prevents performance-sapping

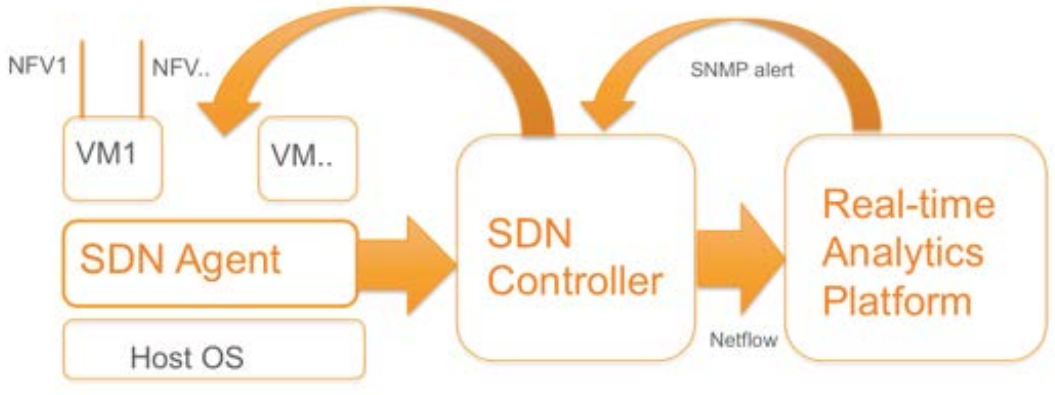

Figure 7: dynamic resource allocation allows for resources and functions to be optimized across data centers

measurement code in core applications. This will maximize server utilization while maintaining user performance. From a business standpoint, an analytics solution could accurately predict scale points to manage investment.

Business-powered use cases (closed-loop trigger or visual discovery) around assessment of performance-related issues. Real-time browsing and purchasing data can be streamed into analytics and forecasting applications. As business & performance conditions change, the IT systems must follow suit & adapt; interacting with the network functions to eliminate inadvertent but costly errors that are driven by static (poor) system configuration. Or, from a product standpoint, applications with different QoS requirements can be sold with differentiated pricing – bulk applications that are not critical to revenue can be transported when the network is not being heavily utilized.

Security & Root Cause

Data center security is traditionally a linear firewall implementation. The current service-deployment implementation strategies allow for large vulnerabilities, as security is generally added on after deployment, as an afterthought. The solution is to distribute security systems that are built in from the beginning - systems that carry out regular automated tests to track compliance.

By tracking and characterizing malware threats / fraudulent transactions etc. in real-time (for neutralization), machine learning algorithms can be implemented to effectively characterize new as-yet unknown threats. By providing a seamless correlation between the physical and virtual domains, obscure patterns (such as those that span datacenters) can be identified to characterize and neutralize such threats. A virtualized firewall instance can be then implemented, or the centralized controller can interact directly with the routing/switching mechanism to drop offending packets. Service chaining based on such virtual instances could occur – having a flow pass through a network monitoring VNF, followed by a load balancing VNF, to ultimately a firewall VNF. Furthermore, the underlying analytics platform could provide predictions based on network behavior (using data science techniques such as application of Granger causality analysis).

This data must be structured in an appropriate visual/closed-loop architecture to target the described use cases. For example, a Security solution

Figure 8: example user interface of threat identification and propagation analysis

might tackle its use case set through the following three modules:

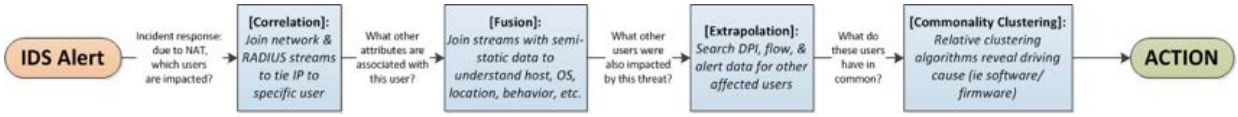

Figure 9: flow chart detailing security use case from Alert, Correlation, Fusion, Extrapolation, Root Cause Analysis, & Action

Security dashboards: by exposing the top threats, correlating the events to the user, and overlaying the context (device, location, OS etc.), a security stakeholder can implement assurances and ongoing monitoring.

Security investigations: once events have been correlated, their attributes and network behavior can be extrapolated to understand threat propagation and threat impact (see figure 8). This provides required insight in to the threat proliferation and monitors the results of any incident containment effort.

Advanced threat detection: to target the driver of any threat, a compute module can be implemented that not only detects anomalies, but also defines the root cause. Through relative commonality calculations, attributes of the subscribers who visit malicious URLs/IP addresses can be correlated as a driving cause. Furthermore, baseline users exhibit various behavior features (connection ratios, destination entropy, fan-out, etc.), from which a risk score can be calculated based on anomalous behavior (that has been previously shown to precede a threat). To understand impact (now & potential), the Most Likely Origin system (ground zero) can associate with the subscriber attributes to drive a proliferation algorithm that extrapolates the potential impact across the entire subscriber base and systems architecture. For an example of the threat investigation flow, see figure 9.

Once the threat is identified, an increase in sampling rate could be triggered to increase operational awareness. Further, SDN techniques to interrupt the packet flow (as previously-described) can be implemented or the element could be instructed to configure the span port to collect all packets and decode to PCAP.

Conclusion

Implementing real-time datacenter operational intelligence is key to an effective hybrid datacenter strategy. Holistic intelligence of physical, virtual or cloud functions (across multiple data centers) powers the type of scalable analytics that drives automation through structured, end-to-end use cases. The network can be dynamically allocated via functions, storage, compute assets that require deeply business contextual data. Spend can be optimized through data-driven resource allocation. From a product standpoint, deployment of new services can be radically accelerated and SLAs can be more fortified.

The access network can be targeted with a number of data-driven use cases. As CCAP becomes integrated the distribution of flows can be optimized via demand, latency, priority, or other metrics. Content can be dynamically cached as CDN networks move from traditionally purpose-build appliances towards a software-controlled cache on standard hardware. Other virtualized

functions such as security, CG-NAT, VPN, or more can leverage the same underlying data and processing capabilities inherent to an operational intelligence platform.

A distributed-compute, real-time analytics platform is the key enabler to these closed-loop use cases around data center & access network control virtualization. Key virtual network functions (such as VNF) are dependent on compute components that are fed pre-processed data on an extremely timely basis. With many modern datacenters already operated under virtual control, access networks are following suit with a slow transition towards a fully virtualized and centralized control scheme. With this in mind, implementing the underlying analytics (operational intelligence) platform will position an operator with the ability to implement virtualized control when and where it becomes available. This platform inherently operates on a 'collect-once, analyze-many' paradigm that supports extremely quick deployment of specialized VNFs.

REFERENCES

[1] Beloglazov, Anton, and Rajkumar Buyya. "Energy Efficient Resource Management in Virtualized Cloud Data Centers." *Cloud Computing & Distributed Systems (CLOUDS) Laboratory, Univ. Melbourne* (2010): n. pag. Print.

[2] Bhowmik, Sukanya. "Distributed Control Algorithms for Adapting Publish/Subscribe in Software Defined Networks." *University of Stuttgart* (2013): n. pag. Print.

[3] Chuanxiong, Guo, Guohan Lu, Helen Wang, Shuang Yang, Chao Kong, Peng Sun, Wenfei Wu, and Yongguang Zhang. "SecondNet: A Data Center Network Virtualization Architecture with Bandwidth Guarantees." *ACM CoNext* (2010): n. pag. Print.

[4] Dong, Yaozu, Xiaowei Yang, Jianhui Li, Guangdeng Liao, Kun Tian, and Haibing Guan. "High Performance Network Virtualization with SR-IOV." *Journal of Parallel and Distributed Computing* 72.11 (2012): 1471-480. Print.

[5] Donley / CableLabs, Chris. "Leveraging Openflow in DOCSIS Networks." *NCTA Spring Technical 2013* (2013): 59-76. Print.

[6] Durr, Frank. "Towards Cloud-assisted Software-defined Networking." *University of Stuttgart / Institute of Parallel & Distributed Systems* (2012): n. pag. Print.

[7] Haider / NICT Tokyo, Aun, Richard Potter / NICT Tokyo, and Akihiro Nakao / NICT Tokyo. "Challenges in Resource Allocation in Network Virtualization." *ITC Specialist Seminar* (2009): n. pag. Print.

[8] Jain, Raj, and Subharthi Paul. "Network Virtualization and Software Defined Networking for Cloud Computing: A Survey." *IEEE Communications Magazine* 51.11 (2013): 24-31. Print.

[9] Koldehofe, Boris, Frank Durr, Muhammad Tariq, and Kurt Rothermel. "The Power of Software-defined Networking: Line-rate Content-based Routing Using OpenFlow." *University of Stuttgart / Institute of Parallel & Distributed Systems* (2012): n. pag. Print.

[10] Kulkhani, Shridhar. "Software Defined Networking (SDN) for Cable Access Networks." *SCTE Technical Forum 2013* (2013): n. pag. Print.

[11] Liou / Infinera, Chris. "Optimizing Multi-Layer Networks with Transport SDN." *SCTE Technical Forum 2013* (2013): n. pag. Print.

[12] Lively / Cisco, Dave. "Software-Defined Networking and Cloud – Enabling Greater Flexibility for Cable Operators." *NCTA Spring Technical 2013* (2013): 77-84. Print.

[13] Nandiraju / Arris, Nagesh, and Sebnem Ozer / Arris. "Applying the Software Defined Networking Paradigm to MSO Commercial Networks." *NCTA Spring Technical 2013* (2013): 85-98. Print.

[14] "OpenFlow-enabled SDN and Network Functions Virtualization." *Open Networking Foundation, ONF Solutions Brief* (2014): n. pag. Print.

[15] Rahman, M. R., and R. Boutaba. "SVNE: Survivable Virtual Network Embedding Algorithms for Network Virtualization." *IEEE Transactions on Network and Service Management* 10.2 (2013): 105-18. Print.

[16] Ram / Rice University, Kaushik, Jose Santos / HP Labs, Yoshio Turner / HP Labs, Alan Cox / Rice University, and Scott Rixner / Rice University. "Achieving 10 Gb/s Using Safe and Transparent Network Interface Virtualization." *VEE* (2009): n. pag. Print.

[17] Sharma, Akshay, Deborah Kish, Sree Chadalavada, Laura Flowerree, and Sean Moore. "Forecast Overview: SDN and NFV in Carrier Infrastructure, Worldwide, 2013." *Gartner* (2013): n. pag. Print.
Skorupa, Joe, Mark Fabbi, and Akshay Sharma. "Ending the Confusion About Software-Defined Networking: A Taxonomy." *Gartner* (2013): n. pag. Print.

[18] Smith / Juniper Networks, Andrew, Colby Barth / Juniper Networks, and Saif Rahman / Comcast. "Enhancing DOCSIS Services through Network Functions Virtualization." *NCTA Spring Technical 2013* (2013): 99-105. Print.

[19] Somohano / Juniper Networks, Pilar, Brian Mutcherson / Comcast, and Boon Thau Loo / University of Pennsylvania. "Unleashing Network Potential through SDN in Virtual Networking Lab Environments." *SCTE Technical Forum 2013* (2013): n. pag. Print.

[20] "Spark Documentation & Overview." *Spark 0.9.0 Documentation*. Apache, 2014. Web. <http://spark.apache.org/docs/latest/>.

[21] Wang, Xiaoying, Zhihui Du, Yinong Chen, and Sanli Li. "Virtualization-based Autonomic Resource Management for Multi-tier Web Applications in Shared Data Center." *Journal of Systems and Software* 81.9 (2008): 1591-608. Print.

[22] White / Arris, Gerry. "Can DOCSIS Networks Leverage SDN?" *NCTA Spring Technical 2013* (2013): 105-25. Print.

KEY ACRONYMS

- ACL: access control list
- BGP: border gateway protocol
- CCAP: converged cable access platform
- CDN: content delivery network
- CMTS: cable modem termination system
- CSP: communications service provider
- CG-NAT: carrier-grade network address translation
- DCOI: data center operational intelligence
- DNS: domain name system
- DOCSIS: data over cable service interface specification
- DPI: deep packet inspection
- eQAM: edge QAM
- GSLB: global server load balancer
- iLO: HP Integrated Lights-Out
- IPDR: internet protocol detail record
- ISP: internet service provider
- MIB: management information base
- MLlib: Spark machine learning library
- MSO: multiple systems operator
- NF: network function
- NFV: network function virtualization
- PCAP: Packet capture
- PCMM: PacketCable Multimedia
- PS: policy server
- QAM: quadrature amplitude modulation
- QEMU: Quick EMUlator
- QoS/QoE: quality of service/experience

- RF: radio frequency
- SDK: software development kit
- SDN: software defined networking
- SLA: service-level agreement
- SNMP: secure network monitoring protocol
- UI: user interface
- VNF: virtual network function
- VNR: virtual network resource
- VPN: virtual private network
- VRF: virtual routing & forwarding
- WAN: wireless access network