Assessing Network and Equipment Failures in the New SDN/NFV Architectures

Marlon Roa
Chris Liou
Vijoy Choyi
Mark Bieberich
Infinera

Abstract

There is little doubt that SDN (Software Defined Networking) and NFV (Network Function Virtualization) are, jointly, a disruption and the breaking point for the telecommunication industry. The promise of lower cost and enhanced functionality of a service provider network is now palpable as we start seeing both technologies mature.

As these technologies converge toward a standard approach, the next step is to look into the details of actual implementation. Among those details, the issue of failure prevention of the network supporting and implementing the said virtualization stands out since a bad implementation in this new control layer approach could, ultimately, create critical failures in the data layer.

This paper explores this topic aiming to raise awareness of the potential complexity of the said network which cannot be omitted from the overall SDN and NFV business case.

INTRODUCTION

SDN and NFV are being implemented in the transport network attempting to follow the success of the software-defined evolution in data centers along with its focused hardware pieces (i.e. controllers and white-boxes). SDN moves the control layer out of data layer equipment or custom management systems and into a central "controller" platform, which now executes the control layer functions for multiple data-layer devices. This lowers the cost of the data-layer devices and enhances the scalability and flexibility of a particular network. It also allows for the easy addition of new services and includes the potential of interoperability among operators' networks for fast multi-provider service turn-up, for example.

In the case of NFV, service providers are seeking solutions that provide a non-purpose-specific, data-layer hardware platform where diverse data-related functions can run either in a central server or in the data-layer equipment as software applications. Again, the objective is to reduce the cost of equipment and enhance scalability of the providers' service offerings.

As a result, the SDN controller hardware concentrates critical transport decisions for its particular set of data-layer devices. The central NFV Management and Orchestration (MANO) solution controls the interworking of specific functions being run by, or controlling, the function execution in a set of data-layer devices within a particular network segment.

Let's approach the analysis of preventing and resolving network failures in SDN/NFV networks from different perspectives:

- Hardware and site failures of the SDN controller, SDN orchestrator or NFV MANO
- Network failures in the interconnectivity between SDN controllers and orchestrators
- Network failures in the interconnectivity between SDN and NFV devices and the data-layer equipment

- The added complexity and cost as a consequence of a resilient SDN/NFV network

For the sake of simplicity, we assume the SDN controller and NFV MANO coexist in the same server under their own VM (Virtual Machine) space. When this is not the case, and they run in their own hardware, one can extrapolate that the complexity for redundancy must be duplicated for each virtualization scheme increasing the cost of the overall deployment.

SDN/NFV HARDWARE REDUNDANCY CONSIDERATIONS

In a traditional network management system (NMS) reliability is necessary to guarantee constant client-side access and to preserve the databases containing configuration and status information. HA (High Availability) schemes have been used to implement redundant NMS systems including geographical diversity in some cases.

As the number of nodes to be managed increased, it was necessary to change the NMS architecture to allow the implementation of hierarchical or distributed approaches, beyond the initial centralized method. The servers, under which the NMS runs, already include redundancy features such as protected hard drives, fans, power supplies, processor redundancy or full hardware redundancy where software can maintain a load balancing scheme as well as monitor the health of the hardware cluster. In addition, modern NMS implementations can be virtualized over common hardware.

The NMS has usually been a proprietary solution, so the vendor included, or recommended, the server hardware to guarantee proper behavior, especially when in a distributed or hierarchical mode. In addition, NMS redundancy methods also add security measures against malicious attacks from the client access perspective and from the network node intercommunication (security is outside the scope of this paper).

The increased number of managed nodes dictated the implementation of a new server or a new load in a distributed system was needed and it also determined if redundancy was needed for this new added server.

As we turn into SDN/NFV, not much changes in terms of server redundancy as the industry was already starting to virtualize the NMS implementations and loading it in common white-boxes.

If the NMS solution in the network has not been virtualized, then there is the extra cost of switching to such an environment for SDN/NFV which involves not only the hardware and interconnect, but the retraining and/or hiring of staff with the correct skills set to maintain the virtual environment.

Hence and up to this point the level of complexity to fit a redundant hardware scheme to prevent single-point of failures for SDN/NFV deployments does not differ much from the classic NMS approach.

However and as a main departure from the traditional NMS scheme, SDN and NFV are actually part of data processing. In the case of SDN, routing, re-routing and cross-connect decisions are part of its responsibility and in the case of NFV, actual data plane functions can take place in a server and/or a fast and reliable transfer of a data-related function from the NFV server to the node is now critically important. This change in network management paradigm affects the hardware and interconnectivity needs, and the latter is covered in the next sections of this paper.

Also, it is important to point out that *"Service continuity is not only a customer* *expectation, but often a regulatory requirement, as telecommunication networks are considered to be part of critical national infrastructure, and respective legal obligations for service assurance/business continuity are in place"* (NFV ETSI [2] specification).

It is then critical that SDN controller functions have a high degree of availability to avoid long periods where the network does not respond to new connections, new labels (Ex: IP addresses, MPLS labels, Ethernet VLANs or MAC addresses) or lack of reconfiguration due to failures or traffic congestion, for example.

To this end, Openflow switch specification 1.2 added the support of multiple controllers, namely slave/master/same controller objects, for redundancy issues, and it leaves it up to the industry the design of the topology to implement the protection.

In the case of NFV, it is even more critical that the server with the NFV MANO function has a high availability as a failure in this server would cause the lack of loading customer requested functions which can violate SLAs or, worse, create a situation where the network assumes an NFV function has been implemented and opens it to customer traffic, which worst case could become a legal matter (Ex: Failed firewall or encryption VNF execution and customer traffic to be protected is enabled to be transported anyway).

In both cases (SDN and NFV) and due to the amount of traffic being processed, latency concerns and/or the number of nodes being managed, the controller scheme is forced to migrate to a hierarchical approach similar to what occurs with the traditional NMS case. In NMS, we could have a slave NMS server that did not need to be backed up since its data was periodically stored in the HA master NMS server cluster and it was seen as not critical for operations since traffic continuity was always maintained by the underlying network elements. Instead and in the case of SDN/NFV, the servers handle portions of data processing which, in most cases, will call for redundancy in these slave locations.

Hence, an increased expense is incurred (vs traditional NMS model) by adding HA schemes to the slave location which can call for cluster structures requiring not only protection servers, but HA monitor solutions where all this hardware may have to be located in non-temperature control environments requiring the extra expense of hardened hardware.

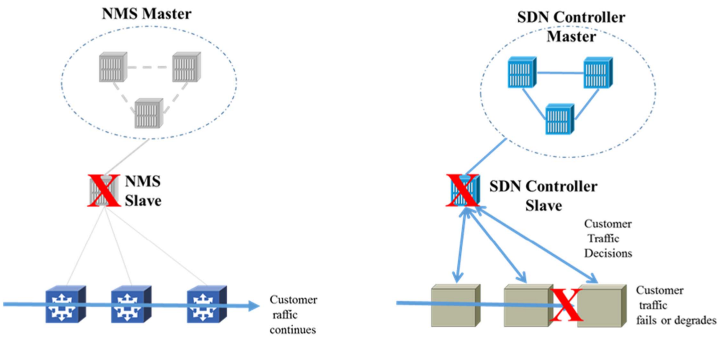


Figure 1 – Management layer failure and traffic effect

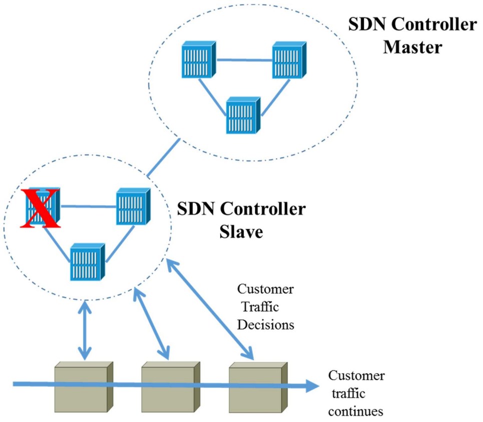


Figure 2 – Required Server redundancy to guarantee traffic continuity

Differing from SDN, NFV requires a level of redundancy on the host of the VNF whether it is a white-box or the network element itself. In the traditional method, the network functions resided in the same hardware processing the data plane without the need to be downloaded, instead they remained idle until required to execute.

This meant there was no need to flag its existence or its correct deployment except for an after-reboot/reset integrity check that protected all logic in the hardware. Under the new NFV scheme, the hardware processing the data plane needs to add a method to check the integrity of the VNF to be run, then exchange a control protocol with the VNF MANO and add cache and logic that can restore the executing VNFs in case of reboots or resets events.

Enhanced complexity is needed to ensure that after a reboot/reset event, the NFV MANO was not in the middle of a VNF transfer along with the resync of the hardware with the MANO to ensure proper operation moving forward and, in the worst case, re-transmit many VNFs from MANO to the network device making the outage even longer for the customer traffic. This prolonged outage could be reduced by adding full hardware protection for the hardware running the VNFs.

Yet, the recommendation would be to establish SLAs for networks with VNFs to include longer outage time allotment after a reboot/reset event than a traditional network. Additionally, extra-testing of the NFV devices reaction to this type of events must be included in qualification testing to avoid possible long-term outages in production due to sub-par implementations.

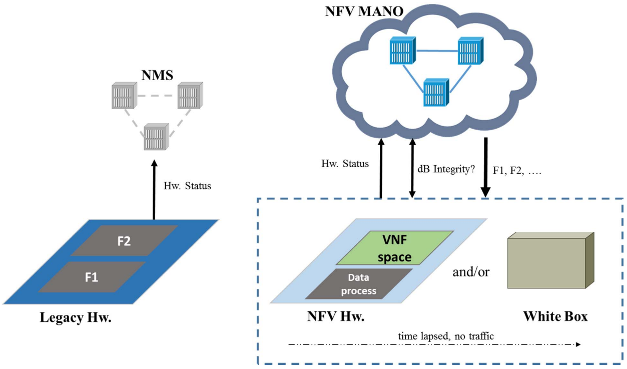
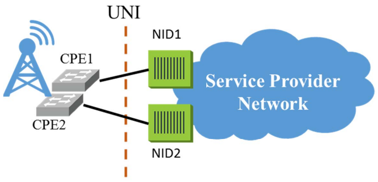


Figure 3 – Reboot/reset scenario – Legacy and NFV network element

In general, NFV ETSI recognizes the need for HA of the new solution and so it requires support for

- *"Service continuity and failure containment*
- *Automated recovery from failures*
- *Eliminating single points of failure in the underlying architecture*
- *Multi-vendor environment*
- *Hybrid Infrastructure*" NFV ETSI

However, it does not enforce a common approach to solving these issues. Hence, the selection among the different vendors of controller software, server hardware, network elements and their interconnectivity to guarantee carrier grade service becomes a new item for operators to evaluate vendors and solutions for.



Legacy Case

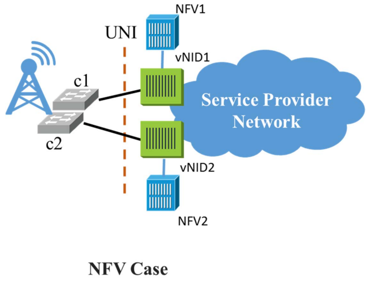


Figure 4 – Customer request for separate hardware at UNI

There is also the very real possibility that customers of a service provider will be unwilling to share the same hardware for their VNFs with other customers VNFs as it is common in demarcation deployments, for example. In this case and if VNFs are running in a common white box near the CPE site, the service provider would be pushed to add a white box and node element for that customer, possibly not being able to charge for the multiple hardware pieces since it is common today to provide the dedicated CPE device at no extra cost.

In all redundancy and load balancing cases, server hardware redundancy is a key part of any protection scheme (including HA, 1+1 or 1:N). SDN and NFV transactions and objects for operation are being standardized, but no effort has been made to normalize the multi-vendor space of server hardware. Hence, an extra operational cost is added to the service provider reflected in extra qualification testing to ensure that the selected vendors servers run the SDN and NFV at a similar processing and connectivity performance level, so if deployed and executing protection switching, the overall data-processing functions will not suffer by a mismatch in performance from one vendor server to another.

Besides the SDN controller, there exists the SDN Orchestrator that is not necessarily part of the traffic processing portion of the network; hence, the approach to redundancy can follow the NMS or datacenter HA concepts without any special provisions.

Then, looking at cost and complexity for the hardware portion of SDN/NFV, the SDN Orchestrator protection is similar to the traditional datacenter and NMS protection schemes, also mixed with OSS. The SDN controller is a new layer of protection where the hardware redundancy requirements are similar to the said NMS and Datacenter schemes, except that these instantiations of carrier-grade redundancy for SDN/NFV controllers can be expected to be far more numerous than in the legacy case and with the probability of requiring hardened server solutions in some locations.

In addition, the operational costs can increase as a network element solution now requires that the SDN management entity software, its host hardware, the NFV MANO, its host hardware and the network element, all of which can be from different vendors, work in fully synchronized and in a manner that their individual execution do not disrupt service continuity or perform below traditional networks. This performance has to be added as new testing by the service provider vendor or solution qualification process.

A way to mitigate this cost increase is to embrace the traditional selection of a handful of vendors where each owns the set of software and hardware of network elements and the management. This basically reduces the service providers cost savings by locking them again to a particular vendor for the network elements, management (SDN controller, NFV MANO) and the necessary software.

Alternatively and as a better option, the standards bodies can define a certification for hardware and software in a multi-vendor space where performance details are included in the test plan as processing performance, ports speeds, protection topologies and protection switching performance parameters.

HIGH AVAILABILITY CONSIDERATIONS BETWEEN SDN/NFV LAYERS

When considering an SDN-centric solution that involves distributed orchestrators, controllers, and physical network elements, it is necessary that an architecture is put in place that ensures survivability and a proper level of availability under multiple potential failure or communication degradation scenarios. The distributed system as a whole needs to be tolerant to faults and conditions that include loss of communications, loss of data, and loss of synchronization, as well as potential overload conditions where specific application instances may be subject to higher loads than anticipated.

This section describes some of the key issues and considerations regarding the communications network between the orchestration and SDN control layer, as well as between the SDN control layer and the physical network infrastructure.

Resiliency Considerations: Orchestrator & SDN Control Layer

The general commonly-accepted multi-tiered architecture for a scalable & distributable SDN solution involves service orchestration systems interacting with typically multiple SDN controllers. Service definition and orchestration is generally handled by the orchestrator, and the configuration of the service into the often multi-domain, multi-vendor network infrastructure is achieved via programmatic interaction with generally different vendors' SDN controllers, generally through an Application Programming Interface (API). This separation of functionality allows the orchestrator to focus on service abstraction and service lifecycle functionality on an end-to-end basis, while the SDN controllers can focus on network-specific configuration, management, and control for implementing its specific portion of the end-to-end service.

The functional separation of roles and responsibilities helps to facilitate the distribution of software modules running on different compute platforms, interacting via some form of client/server information exchange inherent to a well-defined API. This distributed software building block approach not only helps minimize the impact from changes and independent evolution of each individual software module, but also helps to facilitate software scalability, using techniques described in a later section of this paper.

However, distribution of software functions and the API's between the orchestration and SDN control layers present a potential failure point which requires attention in order to maintain system integrity and resiliency. Examples of failure or degradation modes that an HA solution should address include:

- Loss of data communications between client and server
- Failure of client- or server-side application
- Performance degradation due to insufficient server-side resources or excessive information flooding the client application
- Indeterminate state of server application

These issues are not dissimilar from Data Communications Network (DCN) resiliency issues that exist today between OSS/BSS systems and vendor network management systems. Techniques used in recovery from failures or degradation in these scenarios can be leveraged, with the added benefit of new open-source technologies that may make the implementation more standardized or easier.

Remedies for detection and correction of such issues include:

- Client/server heartbeat monitoring
- Performance & health monitoring of SDN controller application for detection of performance degradation
- Methods for both deep data synchronization as well as rapid re-synchronization to recover from communication lapse
- Active/active or active/standby SDN controller application architecture for redundancy and rapid failover
- Dynamic instantiation and load balancing across multiple server application instances
- Networking resiliency between orchestration and SDN control layer (generally employing mesh networking with multiple communication paths and IP networking techniques)

In scenarios where the orchestration and SDN control layer functions may span a sufficiently large geography, it is necessary to additionally take into consideration the added complexity of increased communications latency, and potentially reevaluation of the use of synchronous or asynchronous client/server communications, as well as whether strong consistency or eventual consistency of state is needed for the specific use-case scenario.

Intra-Orchestrator & Intra-SDN-Controller Resiliency

Within each of the software architectural layers are additional failure points that must be guarded against. Resiliency from internal software or database failures must also be accounted for – watchdog monitoring and general software lifecycle management (including restart, resynchronization, and state recovery) are typical methods employed by the vendor of the solution.

Software functionality within the layers may themselves be distributed across multiple servers, which again presents similar resiliency challenges as mentioned in the previous section, with the exception that the interaction between software modules within the orchestration or SDN control layer is addressable by the vendor's own design options, whereas the resiliency between the orchestration and SDN control layer is generally integral to the formalized API that defines the interaction between layers.

Depending on the implementation of the SDN control layer, additional challenges in maintaining resiliency may exist. SDN control solutions that span large geographies or large numbers of systems, for example, may internally support distributed "root/leaf" architecture to co-locate some functionality closer to the physical infrastructure layer, while still maintaining an aggregated, logically centralized view of resources within the root controller. In addition to survivability of communication lapses between the root/leaf sub-layers of the SDN control layer, database resiliency and state consistency between the root/leaf controllers may present additional challenges, but these would generally be addressed by the vendor of the SDN control layer.

The NFV case

As it was stated in the introduction, this paper assumes SDN and NFV share environment following in the work of ONF and ETSI [8]. Within the NFV MANO, the communication between the different control and monitor parts (orchestrators and managers) follows a similar scheme as in the SDN controller/orchestrator case.

The communication between NFV and SDN will also need to be protected using normal cluster/HA methods. A more delicate interconnectivity situation is present between the NFV control layer and the data plane device.

Resiliency Considerations: SDN Control Layer & Forwarding Layer

The communication layer between the SDN controller and the physical network presents another important potential failure point that needs to be protected against. SDN configuration and control data, along with associated faults, events, and performance monitoring data, are typically passed between these layers. Here, router-based DCN resiliency to the forwarding elements plays an important role, ideally with some form of low-latency and reliable transport with sufficient QoS to help mitigate communication degradation situations.

In scenarios where full DCN redundancy is not available or cost-effective, such as in some WAN environments, additional communication channels via in-band paths may be required. In these environments, Gateway Network Element (GNE) functionality can be utilized to proxy control plane communications to the target system, ensuring an alternate communication path to the DCN path. This is generally required independent of the device-specific protocol used to communicate forwarding instructions and information.

In scenarios where control plane communications to the end forwarding device is proxied through a pair of redundant system controllers, additional provisions are needed to handle system controller failover and replacement scenarios. Depending on the way system controller redundancy is implemented, this may involve custom recovery schemes.

HIGH AVAILABILIY ARCHITECTURE

The requirements for High Availability can vary depending upon the specific component of the SDN or NFV controller. This is the case when it comes to the connectivity layer at the northbound and at the southbound interfaces of the SDN or NFV controller.

Controller to Data plane connectivity

The HA requirement on the connectivity between the controller and the data plane is quite stringent. Firstly, the network-wide state maintained by the controller will get out-of-sync with the network, and the control plane functions in the controller now makes incorrect or poor decisions based on this data. This would be the case when the packets that were not delivered were routing updates for some link state based protocol. Secondly, the loss of connectivity could result in down time for end users, as would be the case when the network element needed the controller to decide on how to handle some data traffic for which a rule was not yet programmed.

To ensure resiliency in the connectivity between the controller and the data plane, a common technique is to maintain multiple connections between the controller and each network element. Along with this, the network connecting them needs to have multiple diverse paths that the above mentioned connections can traverse.

Figure 5 depicts the redundancy model wherein a connection exists between every node of the controller cluster and a network element. Also, by using techniques such as subnets we can ensure that the path taken by each of the connections does not overlap. In this case, let’s say a link gets severed, then we still have ‘n-1’ connections available between that NE and the controller, assuming there are ‘n’ nodes in the controller cluster. The connection itself can be established by either side initiating it. It is quite common these days for network elements supporting

Openflow to allow configuring multiple controllers to concurrently communicate with.

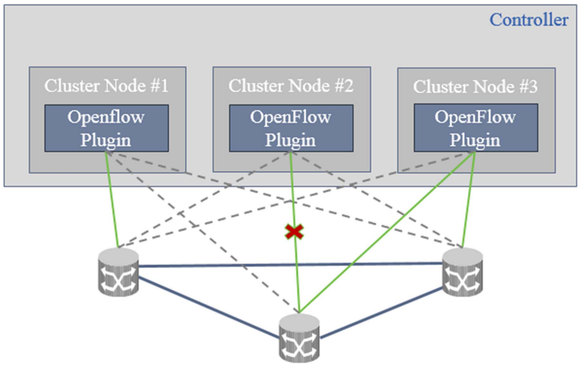


Figure 5 – Connection redundancy between controller nodes and network nodes

Having put in place redundant connections to handle connection failures does take care of making the connectivity resilient, but it does introduce some level of complexity within the controller. When the controller needs to program a set of rules or send down other commands to the NE, it now has the option of distributing the requests across the connections concurrently. Although this may seem like an attractive option w.r.t scalability, it can lead to issues of ordering and dependencies between the requests may break. In order to prevent this, the controller will need to consider just one of the connections active at any given time and the others as standby. This way the write operations from the controller always go on the active connection, while operational status updates coming from the NE can still be accepted over every connection.

The controller cluster can have an election process between the cluster nodes in order to decide which node's connection is to be given the active status. The election can occur on first discovery of the NE and also whenever the active connection fails.

With NEs supporting southbound protocols like Openflow, the default role of each connected controller is equal. However, this can be overridden such that the role can be set to slave. Once the active connection election occurs in the controller cluster, each of the controller cluster nodes can then request the Openflow NE to switch the role accordingly.

In the case of NFV MANO, we must assume that the connectivity is direct between the MANO and the network device. A low latency link is a must as the NFV aims to provide the same performance as legacy carrier-grade data-plane solutions. In addition, the bandwidth must be high enough to handle control layer traffic and any data-layer traffic transported over the same link. A QoS scheme can guarantee coexistence of both low latency data transport and control traffic continuity.

The link can be protected in a scheme that provides 1+1 protection or, in some non-critical cases, an in-band alternate path can be implemented, but the provider must be aware that non-network-element resident VNF execution could degrade its performance to levels that may violate the stipulated SLA.

As it can be concluded, sharing the same access path to the network element by NFV and SDN can hinder scalability of both SDN and NFV or it can impact the performance of VNF execution due to the higher requirement for latency and speed in the VNF case. Then, as VNFs are loaded in a particular network element location, it can be expected that as a worst case, the VNF load and the SDN control tasks can grow to a point where a network element can be required to handle a physical connection for SDN and another physical connection for NFV, both of which can be protected. Another alternative is to upgrade the management link to a high speed, optical-base connection. Yet and in either case, the cost and complexity increase from that of an equivalent legacy implementation.

Controller to Orchestrator connectivity

Although we need High Availability for the connectivity between the controller and the orchestrator, the requirements are not as high as that at the southbound interface. Since the orchestrator is not directly dealing with packet-in messages coming from the dataplane and not in charge of maintaining the global network state, it can afford to sustain occasional glitches in connectivity with the controller. Moreover, with automation and retry logic that orchestrators are typically built to handle, it should be quite trivial for the orchestrator to re-try any provisioning tasks that got impacted while the connectivity was down with the controller.

Given that a typical SDN controller in itself is comprised of a cluster of nodes where each node is active, it is not going to be feasible and is not recommended for the orchestrator to directly be exposed to the cluster nodes. This is because the controller can at any point in time decide to spin up or bring down nodes in its cluster based on the load and this should in no way impact the interaction with external systems. By fronting the controller cluster nodes with a load balancer, we not only achieve this decoupling with external systems but also get the ability to distribute the request load across the nodes of the controller. With this, the orchestrator then only interacts with a virtual ip address that is serviced by the load balancer. The load balancer will appropriately distribute the incoming requests to one of the cluster nodes depending on the load balancing policy configured by the network administrator.

To ensure HA in the connectivity between the controller and the orchestrator, we now need to ensure not only that there are redundant paths deployed in the network to reach between them, but also that there is a redundant load balancer.

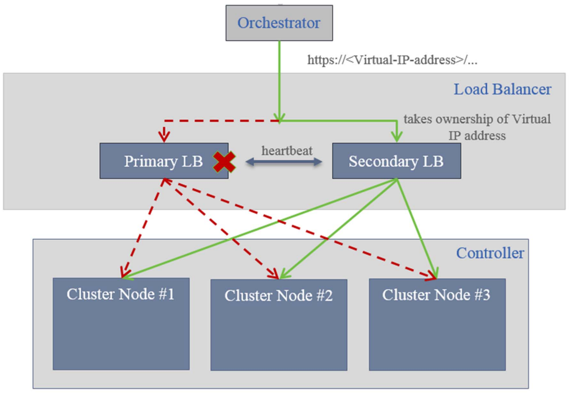


Figure 6 – Redundant load balancing

As can be seen from Figure 6, the secondary load balancer detects that the primary has failed based on loss of heartbeat. The secondary then takes ownership of the virtual IP address and updates the ARP cache. From that point on, any new request from the orchestrator will get routed to the secondary load balancer, which in turn will distribute the requests to one of the cluster nodes of the controller.

This model also handles scenarios where one of the controller cluster node fails. In such cases, the load balancer will detect the controller cluster node failure and then ensure that future requests from the orchestrator are forwarded only to one of the remaining cluster nodes.

ADDED COST AND COMPLEXITY OF A RESILIENT SDN/NFV NETWORK

From the data plane to the controller and orchestration layers, the migration to SDN and NFV requires operators to embrace new approaches to network resiliency. A resilient SDN and NFV network architecture is critical to maintaining and enforcing SLA's; however, there are important cost and complexity implications that must be identified, evaluated and planned for as part of the process to validate the network virtualization business case.

In comparison to more simplistic resiliency measures in traditional NMS and OSS architectures, where data exchanges with the network are administrative in nature, SDN and NFV execute control plane operations and network function instantiation—critical functions that have a direct impact on the ability of the network to respond to end-user demand. As a result, SDN and NFV require more sophistication in network resiliency planning, design and testing to ensure SLA's are met and operations can scale, reliably, over the long term.

Added cost and complexity that result from SDN and NFV resiliency can be grouped into the following categories: qualification testing, hardware and software distribution, and physical interconnectivity.

Qualification Testing

One of the central tenets of network virtualization is openness, the concept of enabling integration and interoperability of disparate applications and network functions—across hierarchical boundaries—using standards-based, open API's. With an open network and operational environment, operators can create and deliver services using solutions from multiple vendors as well as their own and partner-developed applications—an increasing number of which will be based on emerging open source software. As openness increases the number of vendors and systems represented in the network, and as new open source solutions evolve, qualification testing is likely to become more complex.

In addition, in the near term, while vendor hardware and software solutions vary considerably in performance, scale, resiliency schemes supported, and the extent to which they offer open solutions, operators will likely incur higher pre-production qualification testing costs. Nevertheless, as standards-based API's, open source software solutions, and general purpose servers become more mature and commercially validated over the mid to long term, qualification testing should become easier and expenses should decrease relative to total cost of operations.

Hardware and Software Distribution

To achieve scale in SDN and NFV networks and operations, service providers will embrace increasingly distributed architectures of the controller and orchestration layers. As a result, resiliency of key software components and the server infrastructures on which they operate—often spanning multiple, geographically dispersed data centers—will become a more complex challenge than it was in a traditional NMS and OSS environment.

Many of the hardware and software resiliency measures discussed in previous sections are unique to SDN and NFV, and represent added capital and operating expenses to the migration effort. Because SDN controllers play an active, central role in traffic management, operators must consider a wide range of new and proven resiliency measures to employ across network and operational layers. For example, mesh networking between SDN controllers and orchestrators represents a degree of resiliency largely unnecessary in the traditional NMS and operational domains of physical architectures. The use of more containerized, microservices-based software architectures, which allow for more distributed and flexible implementation of SDN control and orchestration functions, is another driver of new resiliency requirements that can increase complexity and cost.

Physical Interconnectivity

As deployments of SDN and NFV scale, the number of servers, clusters and data centers—and the optical paths interconnecting them across an expanding geographic landscape—will grow steadily. With connectivity among nodes and sites increasing and requiring optimized performance and high availability to assure network services, cost and complexity of implementing resiliency measures is likely to increase.

The previous sections outline physical layer considerations and requirements for connectivity between the network and controller layer and controller to orchestration layer. Connectivity between controller clusters and orchestrators may require additional nodes, such as load balancers, to optimize controller performance. However, complexity between these layers is relatively low; there are more complexity and cost considerations from the controller to the underlying network. Fundamentally, there are more potential failure points, more network performance attributes to control (such as latency and loss), and a larger number of constraints posed to the network planner.

SUMMARY AND CONCLUSIONS

SDN and NFV are changing the way we approach network design. They expand the market for new vendors to compete in this arena while promising reduction of costs for service providers in a similar fashion as in the case of data centers. However, these technologies require the inclusion of known resiliency methods to locations and devices in the network that were not designed to support them. Hence, there is an expense to be included in the business case in order to support new demands like

- The implementation of HA clusters in new non-headend or datacenter locations,
- New low-latency redundant management paths or
- Changes of qualification and testing of network equipment, etc.

In addition, the actual separation of functions among multiple devices forces the protection of the integrity of the information being exchanged. This includes cost and complexity increases due to encryption and firewall schemes. This is an extensive topic, not covered by this paper, but still a key expense and complexity item.

As it is always the case with new technology life cycles, the expectation is that as the hardware and software mature, the cost of implementation will decrease over time. However, and in the near term, protection and redundancy could condition the extent to which some service providers can commit to deploy SDN and NFV in portions or even all of their networks.

REFERENCES

1. ETSI GS NFV 001 v1.1.1. “Network Functions Virtualisation Use Cases”. 2013
2. ETSI GS NFV-REL 001 v1.1.1. “Network Functions Virtualisation Resiliency Requirements”. 2015
3. ONF, “OpenFlow Switch Specification V1.3.0”. 2012
4. Software Defined Networks: A Comprehensive Approach, Paul Goransson, Chuck Black June 5, 2014
5. Kretz, F. Ramos, and P. Verissimo. “Towards Secure and Dependable Software-Defined Networks”.
6. Bernstein. “Availability For Network Function Virtualization”. 2015 Spring Technical Forum
7. A Basta, M Hoffman et al.“Applying NFV and SDN to LTE Mobile Core Gateways; The Functions Placement Problem”. 2014
8. ONF. “Openflow-enabled SDN and NFV solution brief”. 2014