Internet Engineering Task Force
Internet Draft
Intended status: standards track
Expires: April 2013

Florin Balus
Dimitri Stiliadis
Nuage Networks

Nabil Bitar
Verizon

Wim Henderickx
Marc Lasserre
Alcatel-Lucent

Kenichi Ogaki
KDDI

October 22, 2012

Federated SDN-based Controllers for NVO3
draft-sb-nvo3-sdn-federation-01.txt

Status of this Memo

Copyright Notice

Abstract

The familiar toolset of VPN and IP/MPLS protocols defined in IETF has been discussed as a good starting point for addressing several of the problems described in the NVO3 problem statement and requirements drafts. However, there is a significant gap between the VPN technologies and the scale and complexity required when the NVEs are running in server hypervisors.

This draft proposes a solution that bridges the gap between the concepts familiar to the data center IT community with some of the best concepts developed by the networking industry.

The proposed solution is based on the understanding that in a cloud environment NVEs may reside in end-devices that should not be burdened with the complexities of control plane protocols. The complex control plane functionalities are decoupled from the forwarding plane to minimize the required NVE processing, recognizing that major hypervisor distributions already employ Openflow to address this issue.

The solution also defines the mechanisms for scaling data center network control horizontally and interoperates seamlessly with existing L2 and L3 VPN devices.

Table of Contents

1. Introduction

Several data center networking challenges are described in the NVO3 problem statement and requirements drafts. A number of documents propose extensions to or re-use of existing IETF protocols to address these challenges.

The data center environment though, is dominated by the presence of software networking components (vswitches) in server hypervisors, which may outnumber by several orders of magnitude the physical networking nodes. Limited resources are available for software networking as hypervisor software is designed to maximize the revenue generating compute resources rather than expending them in network protocol processing.

More importantly the cloud environment is driven by the need to innovate and bring new IT services fast to market, so network

automation and network flexibility is of paramount importance.

This document proposes for NVO3 control plane a combination between IETF VPN mechanisms and the Software Defined Network (SDN) concepts developed by the Open Networking Foundation and already in use in a number of cloud deployments. Existing routing mechanisms are employed when a scaled out data center deployment is required to federate a number of SDN controllers in a multi-vendor environment or to interoperate with non-NVO3 domains.

The proposed solution can be implemented with minimal extensions to existing protocols and to a certain extent is already operational in several cloud environments. It also proposes a simple mechanism that enables NVO3 domains to seamlessly interoperate with VPN deployments without requiring new functionality in the existing VPN networks.

The focus of this draft is on the NVO3 controller; it describes how the SDN concepts defined in [ONF] can be used and extended to perform the NVO3 control plane functions and how SDN controller domains can be federated using existing routing protocols. The handling and support for different NVO3 encapsulations is described briefly in the later sections. The terminology of NVO3 controller may be subject to further changes in the framework draft [NVO3-FWK].

2. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying RFC-2119 significance.

2.1. General terminology

This document uses the terminology defined in NVO3 framework document [NVO3-FWK].

3. Solution Overview

The concept of a NVO3 generic architecture is discussed in the NVO3 framework draft [NVO3-FWK].

This section describes how NVO3 control plane functions can be implemented starting from the SDN concepts defined in [ONF]. The proposed architecture is depicted in the following diagram.

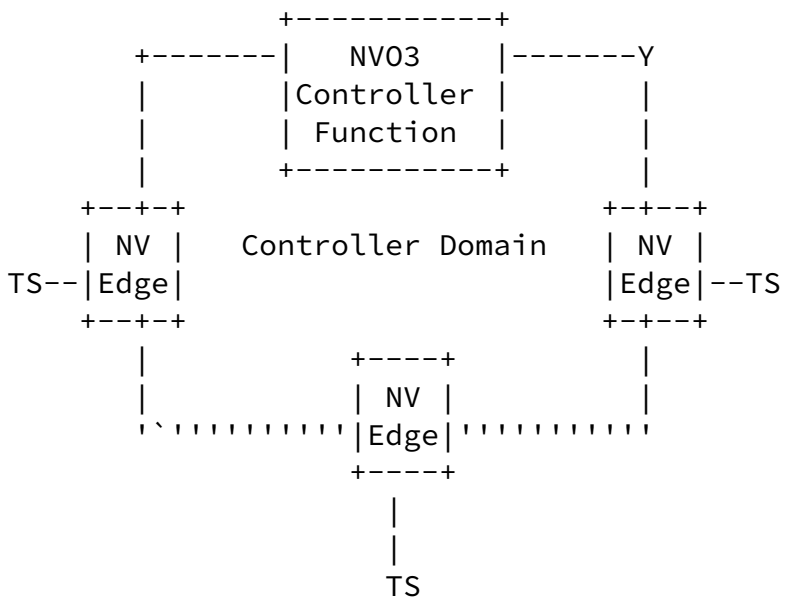

Figure 1 Controller-based architecture for NVO3

NVO3 controller function is implemented using the concept of a SDN controller [ONF]. The Controller is a software component residing in a logically centralized location, responsible for a specific NVE domain [NVO3-FWK]. Each NVE has a control session to the Controller which in turn may run external control plane sessions to communicate to the outside world or to other controllers. All the NVEs sharing a controller represent a controller domain.

The Controller provides a generic API for learning about the Tenant System profile and related events. The mechanism can be through a north bound API of the Controller itself, or through an event tracking mechanism.

As soon as a Tenant System is configured and attached to a Controller domain, the cloud management system informs the Controller about this activation. Existing cloud management systems [Openstack, Cloudstack] assume the Controller provides an API that will enable the cloud management system to notify the Controller about the new service activation. The Quantum plugin in Openstack is an example of an API interface between cloud management and Controller.

Alternative mechanisms can rely on a direct notification of a TS (VM) attachment to the NVE that can be done at the hypervisor layer. The NVE can then pass the TS information to the controller using the procedures described in section 4.1.2.1. For Tenant Systems located in a remote Controller domain, (MP)-BGP is used to advertise and learn the related information to/from the remote controller.

Once the Controller is aware of a new Tenant System attached to a NVE, the TS IP and/or MAC addresses are populated in the controller's routing database. Since the Controller has full visibility of all Tenant Systems belonging to the same tenant within its domain, it will generate the related FIBs and populate the required entries on the NVEs that have a corresponding tenant VNI. The controller can rely on the Openflow protocol [Openflow] for this function, since it is only populating FIB entries, and can use either a pull or a push method or a combination based on the scale and dynamics of the FIB entries that need to be populated in the NVEs, and/or based on a FIB population policy.

When the TS is deleted, the reverse process takes place. The Controller, either from API calls or based on events received from the NVEs, will learn about the TS removal and it will proceed to remove the TS IP and/or MAC from its routing and FIB database. BGP will be used to withdraw these addresses from the remote Controllers to which those addresses were previously advertised and/or managing NVEs with VNIs belonging to the same corresponding tenant VN, if any are present. These changes will also be communicated then to the impacted NVEs by the corresponding remote Controller.

4. Control plane details

The controller is the central component for the NVO3 control plane architecture. In the proposed solution the NVO3 controller utilizes mechanisms to monitor events and create the required connectivity.

This section discusses how the current SDN model [ONF] with appropriate extensions can be used to address the requirements outlined in [NVO3-CPREQ].

To automatically instantiate the required data plane connectivity the NVO3 controller has to perform the following functions:

- Learning of TS profiles and current state (TS State Discovery).
- Auto-instantiation of NVO3 service.
  - o Address advertisement and associated tunnel encapsulation mapping
  - o FIB population.
- Underlay aware routing.

4.1. Tenant System State Discovery

This section sets the stage for a generic set of events and actions that MUST be supported by any NVO3 controller if automatic instantiation of the service is desired. The goal is to converge

first on a common information model between the cloud management system and the NVO3 controller. It is also highly desirable to converge to a common protocol and information model that will allow cloud management systems and Controllers from different vendors to interoperate. At the current time the integration of a controller with different cloud management systems requires customization work and this will lead to interoperability problems and duplicated development efforts to interoperate a matrix of cloud management systems and Controllers.

4.1.1. Tenant System states and related information

There is a large variety of Tenant Systems that may need to receive service from a NVO3 domain. For example in the Cloud networking space a TS may be a VM or an appliance instance (Firewall, LB) assigned to a particular tenant. There are a number of possible

states for the Tenant Systems that need to be signaled to and interpreted by the Controller. The following is an initial set of TS states that is mainly derived by mirroring the model of virtual machine lifecycle management encountered in several cloud management tools:

- Not-deployed: TS exists in management system but is not instantiated. The NVE does not need to know about these instances.
- Running: TS is instantiated, active and ready to send traffic. The appropriate NVEs with instances of the corresponding tenant VNs, must have all required functionality to send and receive traffic for the particular TS.
- Suspended: TS is instantiated, but currently in a suspended state. Traffic from/to the TS can be ignored. Routes to this TS may be withdrawn from the corresponding tenant VN.
- Shutdown: TS is in the process of shutting down. A complete shutdown is not known though, and it will depend on the capabilities of the TS. Traffic from/to the TS must be forwarded.
- Shut off: TS is in the power-off mode and not attached to the NVE. This is similar to the suspend state. Traffic from/to the TS can be ignored. Routes corresponding to the TS must be withdrawn from corresponding tenant VN and the forwarding state at the local NVE must be removed.
- Moving: TS is active but a TS Move command was originated. The Controller must participate in any state transfer functions. The goal is to directly forward traffic to the TS at the new location and possibly tunnel traffic in transit to the old location from the old location to the new one.

- Other: Opaque state that refers to additional states defined by a specialized TS.

Even though, the states above are often related to virtual machines, the model or a subset can cover the physical appliance states as well. Depending on the TS, some of these states might not be easily identifiable (additional mechanisms, liveliness check, are required to detect a crashed or shutdown physical machine).

4.1.2. Tracking local TS events

The controller must have full information about the TS state. This can be achieved in one of two ways:

1. The cloud management system utilizes an API exposed by the Controller to update the TS state. This is the model deployed by the Openstack Quantum API or the Cloudstack Network Guru API.
2. The NVE tracks the above events when it is co-located with the hypervisor using internal mechanisms and reports it to the Controller using a signaling mechanism. When the NVE is not implemented in the hypervisor hosting the TS, a tracking protocol between the hypervisor and NVE can allow the tracking of TS state events. A standard protocol for this tracking function can significantly assist in the interoperability between different hypervisors and NVEs. One such mechanism is discussed in [Server2NVE].

The following section discusses the NVE to controller signaling procedure for the second scenario.

4.1.2.1. NVE to Controller signaling of TS events

In the case where the NVE directly tracks VM events, there is also a need for a standard signaling mechanism between the NVE and the Controller. This proposal utilizes extensions to Openflow protocol [Openflow] in order to accommodate this signaling. Openflow is supported by a number of hypervisor distributions and is already active in some large cloud deployments. Moreover it has already the messaging base required to perform this function.

In the current Openflow specification, the first packet of an unknown flow can be forwarded to the Controller when no match is found in the local openflow switch. One possible method to implement TS event signaling is to extend this functionality and use the TS event as a trigger for a generic "flow request" from the NVE that will carry sufficient information to the controller to allow for service initialization. However, the flow request is extended here

as it does not contain parts of a TS packet, but information of a TS event. Alternatively a new request type can be defined. Details of this procedure will be added in a future revision.

4.2. Address advertisement and FIB population

Once the controller learns about the TS state event from North bound API or from the NVE it performs the following actions:

- Identify the required NVO3 service attributes. Service attributes could include acces lists and policies for certain actions.
- Populate the VN routing and FIB tables with the TS address(es).
- If a push-model is used, it downloads the required FIB updates and service attributes to the NVEs that participate in the related VN (pre-population).
- If a pull-model is selected, it waits for the first packets of the corresponding flows or potentially other requests triggered by some events before establishing flow state, as per the Openflow specification, or some FIB state in the NVE
- A combination of push and pull models may be beneficial. For example flows that require consistent latency and/or no packet loss may be pre-populated using a push model while other less important flows may be populated using a pull model. Similarly, ARP entries may be pre-populated or pulled-in when the NVE sees an ARP request with no corresponding ARP entry in its local cache.

4.2.1. Push versus Pull

The Openflow model described in the previous section enables both a push and a pull model. The selection of a model will depend on the controller and NVE capabilities and deployment requirements. Different implementations might choose alternative methods or a combination of pull/push. This is though an implementation detail that is supported by the basic solution framework.

4.3. Underlay awareness

The Underlay network consists of a core often running IP routing protocols to control the topology and to provide reachability among NVEs. A routing module in the Controller may participate in the underlay IP routing to maintain a view of the network underlay. The routing information exchanged may be used to control path selection or to make forward/drop decisions in the ingress NVEs, so that network bandwidth resources are not unnecessarily wasted.

4.4. Controllers federation

To address a scale-out NVO3 deployment multiple Controllers may be required. The Controllers need to exchange their routing information to allow for NVO3 services to extend across NVO3 domains managed by individual Controllers. The following diagram depicts this scenario.

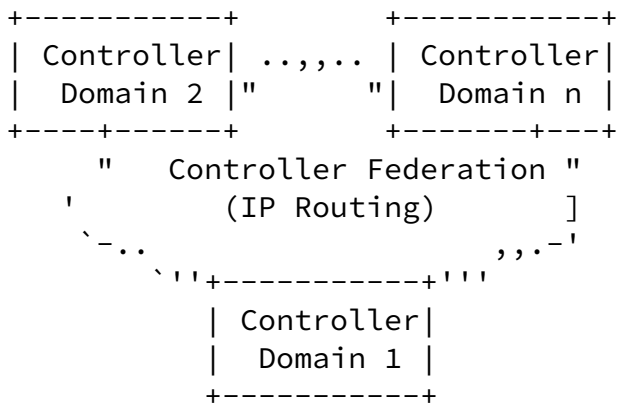

Figure 2 Controller Federation

The fundamental requirement in any such function is a system that enables state distribution between the different instances. When Controllers from the same vendors are used this can be achieved through a pub/sub mechanism or through a distributed database. For example an ActiveMQ mechanism such as the one deployed by Openstack [Openstack] or a DHT based database like Cassandra.

When Controllers from multiple vendors are in place or when interoperability with existing WAN services is required a standard way of distributing TS information is required.

One can envision that existing VPN mechanisms can be utilized for this function, distributing private reachability information pertaining to the tenant VNs. A Routing module implementing the related MP-BGP procedures can be used as follows:

- If L3 (IP) services need to be implemented across domains, the procedures described in [BGP-VPN], specifically the IP VPN SAFI and related NLRI can be used to exchange the Tenant IP addresses.
- For L2 services the procedures described in [BGP-EVPN], specifically the EVPN NLRI(s) can be employed to advertise Tenant L2 addresses and related information among multiple Controllers.
- For both service types, mechanisms like Route Distinguisher and Route Targets provide support for overlapping addressing space across VPNs (tenant VNs), and respectively for controlled

tenant service topology and route distribution only to the NVEs that have at least one TS in that VPN (tenant VN)

Each Controller can then consolidate the external and internal routing tables and generate required FIB entries that can be downloaded as required to the NVEs.

The advantage of utilizing (MP)-BGP to federate Controllers is that it enables interoperability between Controllers of different vendors as well as interoperability with existing WAN services, including Internet and L2/L3 VPNs. BGP is a proven mechanism that can be used to achieve the required scale and secure isolation for multi-tenant, multi-domain and multi-provider environments.

5. Data plane considerations

The Controller can be used to control different data planes for different solution options.

5.1. L2 and L3 services

MP-BGP VPN and Openflow can be used to program the required NVE data plane for both Layer 2 and Layer 3 services: Openflow is able/can be extended to handle both L2 and L3 FIB entries and multiple tunnel encapsulations while MP-BGP support for multiple address families ([BGP-VPN] and [BGP-EVPN]) allows the extension of both L2 and L3 services across Controller domains.

After the local TS discovery and MP-BGP exchanges the L2 and/or L3 forwarding entries are computed first in the Controller and mapped to different types of tunnel encapsulations based on the type of core network, addressing type and the negotiated service encapsulation type.

The resulting FIB updates can be downloaded using Openflow to NVEs that have VNIs corresponding to tenant VNs associated with FIB entries. The NVEs make use of these entries to forward packets at L2 or L3 depending on the type of service and the desired processing sequence.

5.2. NVO3 encapsulations

A number of vSwitch distributions already provide support for some of the encapsulations that had been proposed in some IETF drafts and allow the mapping of FIB entries to tunnel encapsulations based on these protocols. Opensource code for these encapsulations is also available. FIB entries with associated tunneling encapsulations can

be communicated from the Controller to the NVE using Openflow where supported or via Openflow extensions as required. Openflow supports also the MPLS VPN encapsulations easing the way for interoperability between NVE and VPN domains. Moreover the use of BGP for MAC and IP advertisement for different NVO3 encapsulations has been proposed in [EVPN-NVO3].

6. Resiliency considerations

This section will discuss resiliency for a Controller domain implementing the control framework in this document.

6.1. Controller resiliency

For a large domain, controller resiliency may be required. Non Stop control-plane schemes may be extended to cover the Controller Openflow component in addition to the BGP component. Alternatively the controller resiliency schemes proposed in [Openflow] may be employed in conjunction with Graceful Restart (for example [BGP Graceful-Restart]) for the routing modules.

6.2. Data plane resiliency

From a data plane perspective, there are a number of ways to ensure the NVE can be multi-homed towards the IP core. Existing mechanisms like ECMP may be used to ensure load distribution across multiple paths.

The access multi-homing of Tenant System to NVEs on the other hand is applicable only when the NVE and TS are on different physical devices. Several mechanisms can be utilized for this function depending on whether the TS to NVE communication is over L2 or L3. These use cases will be addressed in a future revision of this document.

7. Practical deployment considerations

7.1. Controller distribution

The Controller is providing the control plane functionality for all the NVEs in its domain. The number of NVEs per Controller domain may vary depending on different scaling factors: for example the number

of tenant systems, VAPs, VNs and tunnel endpoints. The concept of Controller federation allows for modular growth enabling a highly distributed deployment where the Controller may be deployed even down to server rack/ToR level in cases where scalable control plane

may be required or if a BGP-based operational environment is the preferred option.

7.2. Hypervisor NVE processing

The proposed solution is designed to minimize the required processing on the hypervisor NVEs. Complex control-plane policy and routing functions are delegated to the Controller that can be deployed in dedicated processors. Hypervisors only run the Openflow agents required to download the FIB entries and process events in some models, as well as perform the NVo3 forwarding function.

7.3. Interoperating with non-NVO3 domains

The routing module of the Controller enables interoperability with existing non-NVO3 VPN domains where the whole Controller domain appears as just another PE to those domains. A VPN interworking function may need, depending on the encapsulation used, to be implemented in the data plane of the gateway NVEs to existing non-NVO3 VPN domains.

7.4. VM Mobility

VM mobility is handled by a tight interaction between the cloud management system and the Controller. When a VM move is instantiated, the cloud management system will notify the Controller about the move. The VM will transition from a running state in one hypervisor to a running state in another hypervisor. The transition of these events will instruct the Controller to properly update the routes in each of the NVEs.

7.5. Openflow and Open vSwitch

Utilizing Openflow as the basic mechanism for controller to NVE communication offers several advantages:

1. Openflow is a binary protocol that is optimized for fast FIB

updates. Since it relies on a binary format it minimizes the amount of data that needs to be transferred and the required processing to provide increased scalability.

2. Openflow is already implemented in multiple hypervisors and is deployed in some large cloud environments. The current specification supports L2, L3 service FIB and flexible flow definition providing a good starting point for future extensions.

From a practical and deployment perspective, the Open vSwitch [OVS] is already part of the latest Linux kernel and most major Linux

distributions for the major hypervisors (KVM and Xen). Minimizing the new protocols that need to be deployed into servers and relying on the existing hypervisor capabilities can significantly simplify and accelerate the adoption of NVO3 technologies.

8. Security Considerations

The tenant to overlay mapping function can introduce significant security risks if appropriate protocols are not used that can support mutual authentication. Proper configuration of Controller and NVEs, and a mutual authentication mechanism is required. The Openflow specification includes a TLS option for the controller to NVE communication that can address the mutual authentication requirement.

No other new security issues are introduced beyond those described already in the related L2VPN and L3VPN RFCs.

9. IANA Considerations

IANA does not need to take any action for this draft.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[BGP-VPN] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private Networks (VPNs)", RFC 4364, February 2006.

[BGP Graceful-Restart] Sangli, S. et al, "Graceful Restart Mechanism for BGP", RFC 4724, January 2007

10.2. Informative References

[NVO3-FWK] Lasserre, M et.al "Framework for DC Network Virtualization", draft-ietf-nvo3-framework (work in progress)

[NVO3-CPREQ] Kreeger, L. et al, "Network Virtualization Overlay Control Protocol Requirements", draft-kreeger-nvo3-overlay-cp (work in progress)

[Server2NVE] Kompella, K. et al, "Using signaling to simplify network virtualization provisioning", draft-kompella-nv03-server2nve (work in progress)

[EVPN-NVO3] Drake, J. et al, "A Control Plane for Network Virtualized Overlays", draft-drake-nvo3-evpn-control-plane (work in progress)

[Openflow] Openflow Switch Specification, http://www.opennetworking.org

[ONF] Open Networking Foundation https://www.opennetworking.org/

[Openstack] Openstack cloud software, http://www.openstack.org

[Cloudstack] Cloudstack, http://www.cloudstack.org

[OVS] Open vSwitch, http://www.openvswitch.org

[BGP-EVPN] Aggarwal, R. et al, "BGP MPLS Based Ethernet VPN", draft-ietf-l2vpn-evpn (work in progress)

11. Acknowledgments

In addition to the authors the following people have contributed to this document: Thomas Morin, Rotem Salomonovitch.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Florin Balus
Nuage Networks
805 E. Middlefield Road
Mountain View, CA, USA 94043
Email: florin@nuagenetworks.net

Dimitri Stiliadis
Nuage Networks
805 E. Middlefield Road
Mountain View, CA, USA 94043
Email: dimitri@nuagenetworks.net

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
Email: nabil.bitar@verizon.com

Kenichi Ogaki
KDDI

3-10-10 Iidabashi,
Chiyoda-ku Tokyo, 102-8460 JAPAN
Email: ke-oogaki@kddi.com

Marc Lasserre
Alcatel-Lucent
Email: marc.lasserre@alcatel-lucent.com

Wim Henderickx
Alcatel-Lucent
Email: wim.henderickx@alcatel-lucent.com