

Internet Research Task Force (IRTF)  
Internet Draft  
Category: Experimental

R. Krishnan  
Brocade  
Dilip Krishnaswamy  
IBM Research  
D. R. Lopez  
Telefonica I+D  
Peter Willis  
BT  
Asif Qamar  
Evolv

Expires: December 2014

July 21, 2014

## **An Open NFV Architectural Framework for Virality Based Content Caching**

[draft-krishnan-nfvrg-open-nfv-virality-00](#)

### Abstract

One of the key goals of Network Functions Virtualization (NFV) is achieving energy efficiency through workload consolidation. A good example for maximizing energy savings is the Virtualization of Content Delivery Networks (vCDNs) NFV use case where the video streaming workloads exhibit significant difference between prime-time and non-prime-time usage of the infrastructure. This draft examines the practical challenges in maximizing energy efficiency for vCDN workloads. This draft proposes an open NFV architectural framework for conveying content virality information from Cloud applications such as YouTube, Twitter and mechanisms for leveraging it to maximize the energy efficiency for vCDN workloads.

### Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents

at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April, 2014.

## Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

## 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](#).

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction.....</a>	<a href="#">3</a>
<a href="#">2.</a>	<a href="#">Open NFV Architectural Framework.....</a>	<a href="#">4</a>
<a href="#">3.</a>	<a href="#">System Analysis.....</a>	<a href="#">6</a>
<a href="#">4.</a>	<a href="#">Open API Parameters.....</a>	<a href="#">7</a>
	<a href="#">4.1.</a> Content Virality Information.....	<a href="#">7</a>
	<a href="#">4.2.</a> Other Application Information.....	<a href="#">7</a>
<a href="#">5.</a>	<a href="#">Summary.....</a>	<a href="#">7</a>
<a href="#">6.</a>	<a href="#">Future Work.....</a>	<a href="#">7</a>
<a href="#">7.</a>	<a href="#">IANA Considerations.....</a>	<a href="#">7</a>
<a href="#">8.</a>	<a href="#">Security Considerations.....</a>	<a href="#">7</a>
<a href="#">9.</a>	<a href="#">Acknowledgements.....</a>	<a href="#">7</a>
<a href="#">10.</a>	<a href="#">References.....</a>	<a href="#">7</a>
	<a href="#">10.1.</a> Normative References.....	<a href="#">7</a>
	<a href="#">10.2.</a> Informative References.....	<a href="#">7</a>
	<a href="#">Authors' Addresses.....</a>	<a href="#">9</a>



## 1. Introduction

Network Operators use a variety of proprietary hardware appliances. Hardware appliances, though deliver performance, are complex to manage, not easy to scale up/down in capacity and not cost effective. NFV [1] is a movement by network operators all over the world such as AT&T, BT, CenturyLink, Deutsche Telekom, Telefonica, Verizon, and more to address the aforementioned issues. NFV involves the implementation of network functions in software that can run on a range of industry standard server hardware, and that can be moved to, or instantiated in, various locations in the network as required, without the need for installation of new equipment. NFV has many use cases [2], notable of which is the vCDN. The goal of vCDN would be to address virtualization of all the CDN components, but the biggest and immediate impact would be on the cache nodes given the growth in content especially in mobile networks [3].

The key benefits of vCDN are

- . Overall capacity is shared by all content delivery appliances and other virtualized appliances.
- . Since appliances are pure software, it is easy to replace or modify them in the event of new CDN requirements.
- . Besides caching of operator's own content, this enables operators to offer content caching as a service to CDN providers (e.g. Akamai Aura CDN) and large content providers with private CDNs (e.g. Netflix OpenConnect).

Currently, the allocation of VMs for vCDN follows a static model based on weekday prime-time characteristics, business hours etc. This model results in substantial resource over-provisioning, since a lot of content viewed over websites like YouTube and shared over social media like Twitter follow a virality pattern during anytime of weekday or weekend [4]. Additionally, many industry standard servers consume substantial power in the active idle state, which results in severe energy inefficiency. For example, HP ProLiant DL380p Rack Server has a peak power utilization of 324 Watts and consumes 105 Watts (approximately 30% of peak) in the active idle state - this is depicted in Page 17 of [5].

One of the key goals of Network Functions Virtualization (NFV) is achieving energy efficiency through workload consolidation [6]. This draft proposes an open NFV architectural framework for conveying content virality information from applications such as Twitter,



Facebook, YouTube and mechanisms for leveraging it to maximize the energy efficiency for vCDN workloads without compromising performance. This enables network operators to offer new types of content caching services to its CDN customers for example, "Virality Based Content Caching" and also optimize the resource usage of their own CDNs. This draft also does a performance comparison of the proposed approach with the current static model of allocation of VMs for vCDNs and demonstrates the benefits of the approach.

## **2. Open NFV Architectural Framework**

Figure 1 depicts the Open NFV Architectural Framework, adapted from the definition of the ETSI NFV Architectural Framework [7] and extended in this draft to show support for content virality management. It has the following virtual and physical or hardware (HW) infrastructure components as part of NFV Infrastructure (NFVI)

- 1) Compute - physical servers hosting computing elements in the form of Virtual Machines (VMs)
- 2) Storage - physical/virtual storage
- 3) Networking - physical/virtual routers.

The Virtual Infrastructure Manager (VIM), which could be accomplished by open source software like OpenStack [8] for example, performs lifecycle management of the above infrastructure components and maintains a dynamic resource pool of the same. Virtual Network Functions (VNFs) such as firewall, load balancer, CDN etc., each of which runs on multiple VMs, are managed by Virtualized Network Function Manager (VNFM), which performs lifecycle management of VNFs and maintains dynamic resource pool(s) for different types of VNFs. The VNFM exchanges Virtual/Physical resource information with the VIM.

The other elements of the NFV architectural framework include a service orchestrator, and management and support systems such as an Element Management System (EMS), an Operations Support System (OSS), and a Business Support System (BSS).



## Content Virality Information

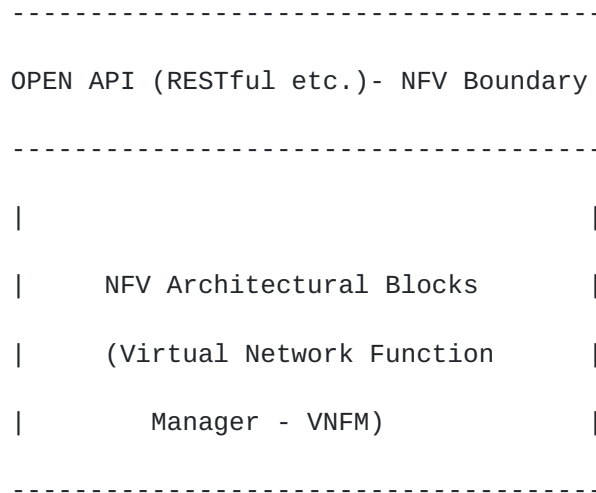


Figure 1: Open NFV Architectural Framework

(Adapted from [7])

The various interfaces in the Open NFV architectural framework are

- Vi-Ha - Interface between the virtualization layer (e.g. hypervisor for hardware compute servers) and hardware resources
- Vn-Nf - Represents the execution environment provided by the NFVI to VNF (e.g. a single VNF could have multiple VMs)
- Nf-Vi - Interface to the VIM - used for VM lifecycle management and other purposes
- Ve-Vnfm - Interface between VNF/EMS and VNF Manager - used for VNF lifecycle management and other purposes
- Se-Ma - Used for getting information about VNF deployment template and other purposes
- Os-Ma - Interface to OSS/BSS - handles network service lifecycle management and other functions.
- Vi-Vnfm - Interface between VIM and VNFM - handles resource allocation requests by the VNF manager and other functions



- Or-Vnfm - Interface between Orchestrator and VNFM - handles collection of state information necessary for network service lifecycle management and other functions

To support content virality information in this open architecture, we suggest the availability of such information through open APIs as depicted in Figure 1. The open API can be based on RESTful framework [9] [10]. Content virality information can be streamed in real-time from applications such as YouTube, and submitted to the VNFM through open APIs. This information can be used by VNFM to populate the dynamic vCDN resource pool with the optimal VNF capacity needed for content caching which can be consolidated to a minimal set of VMs and physical servers. Besides content virality information, we also suggest that the architecture could optionally provide a generic open API framework for handling other application information, such as information regarding firewall services, in real-time if available.

In typical current systems, the vCDN resource pool is statically populated by policies such as weekday prime-time characteristics, business hours etc., and can be significantly over-provisioned to handle any dynamic requests. Current systems also do not delve into specific targeted use cases or a framework for conveying application information in real-time.

The rest of the contribution of this draft is to develop these aspects further in an open architecture framework as suggested in Figure 1. In effect, the differentiating aspects of the proposed architectural framework in this draft are

- A dynamic resource pool that is used to optimally populate the vCDN VNFs with the right amount of VMs and physical servers to minimize over-provisioning.
- Parameters of interest for real-time streaming of application information such as content virality, which could be utilized for resource optimization in an open-API framework.

### **3. System Analysis**

This work is in progress in ETSI NFV as a proof of concept [11]. More details will be described in the upcoming revisions of this draft.



## **4. Open API Parameters**

### **4.1. Content Virality Information**

TBD

### **4.2. Other Application Information**

TBD

## **5. Summary**

This draft proposes an NFV architectural framework for conveying content virality information from Cloud applications such as YouTube, Twitter, Facebook and mechanisms for leveraging it to maximize the energy efficiency for vCDN workloads without compromising performance.

More - TBD

## **6. Future Work**

TBD

## **7. IANA Considerations**

This draft does not have any IANA considerations.

## **8. Security Considerations**

Security issues may arise due to the usage of open APIs for exchanging content virality information. These security issues apply to all forms of open APIs and not limited to exchange of content virality information. This is an aspect for further detailed study.

## **9. Acknowledgements**

None.

## **10. References**

### **10.1. Normative References**

### **10.2. Informative References**

- [1] "ETSI NFV White Paper,"  
[http://portal.etsi.org/NFV/NFV\\_White\\_Paper.pdf](http://portal.etsi.org/NFV/NFV_White_Paper.pdf)

- [2] "ETSI NFV Use Cases,"  
[http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/001/01.01.01\\_60/gs\\_NFV001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf)
- [3] "Cisco VNI White Paper: Global Mobile Data Traffic Forecast Update,"  
[http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white\\_paper\\_c11-520862.html](http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html), February 2014
- [4] "Facegroup: How Videos go Viral,"  
<http://www.facegroup.com/how-videos-go-viral.html>
- [5] "SPEC Benchmark Results: HP Proliant DL380p Rack Server,"  
<http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/Comparing-Dell-R720-and-HP-Proliant-DL380p-Gen8-Servers.pdf>
- [6] "ETSI NFV Virtualization Requirements,"  
[http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/004/01.01.01\\_60/gs\\_NFV004v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/004/01.01.01_60/gs_NFV004v010101p.pdf)
- [7] "ETSI NFV Architectural Framework,"  
[http://www.etsi.org/deliver/etsi\\_gs/NFV/001\\_099/002/01.01.01\\_60/gs\\_NFV002v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV/001_099/002/01.01.01_60/gs_NFV002v010101p.pdf)
- [8] "OpenStack Open Source Software," <https://www.openstack.org/>
- [9] Fielding, Roy Thomas, "Architectural Styles and the Design of Network-based Software Architectures," Dissertation, University of California, Irvine, 2000
- [10] "Hypertext Transfer Protocol - HTTP/1.1," [RFC 7230](#), [RFC 7231](#), [RFC 7232](#), [RFC 7233](#), [RFC 7234](#), [RFC 7235](#)
- [11] "ETSI NFV PoC - Virality based content caching in NFV Framework,"  
[http://nfvwiki.etsi.org/index.php?title=Virality\\_based\\_content\\_caching\\_in\\_NFV\\_framework](http://nfvwiki.etsi.org/index.php?title=Virality_based_content_caching_in_NFV_framework)

## Authors' Addresses

Ram Krishnan  
Brocade Communications  
ramk@brocade.com

Dilip Krishnaswamy  
IBM Research  
dilikris@in.ibm.com

Diego Lopez  
Telefonica I+D  
Don Ramon de la Cruz, 82  
Madrid, 28006, Spain  
+34 913 129 041  
diego.r.lopez@telefonica.com

Peter J. Willis  
BT  
peter.j.willis@bt.com

Asif Qamar  
Evolv  
asif@asifqamar.com