

Software Defined Flow-Mapping for Scaling Virtualized Network Functions

Sharon Barkai
ConteXtream
1927 FallenLeaf Ln.
Los Altos, CA
sbarkai@gmail.com

Randy Katz
UC Berkeley
101 Sproul Hall
Berkeley, CA
randy@cs.berkeley.edu

Dino Farinacci
Cisco Systems
170 West Tasman Dr.
San Jose, CA
farinacci@gmail.com

David Meyer
Brocade Communication
130 Holger Way
San Jose, CA
dmm@1-4-5.net

ABSTRACT

In this paper, we describe the use of software defined flow-mapping for scale-assembly of virtual functions features-capacity.

Categories and Subject Descriptors

B.4.1 [Data Communication Devices]: Receivers, Transmitters
F.1.2 [Modes of Computation]: Parallelism and Concurrency.

Keywords

Software Defined Networks, Network Functions Virtualization, Location Identity Separation, Overlay, Underlay, Flows, Mapping

1. INTRODUCTION

Today's network operators are increasingly weighted down by complexities as they work to accommodate the relentless demand for new services and more bandwidth. A recent Network Function Virtualization (NFV) industry white paper laments a growing and increasingly diverse population of proprietary appliances that make service additions and upgrades more and more difficult. The type of appliances referred to by the NFV paper are typically turn-key in-line systems that maintain real-time state of subscriber mobility, voice and media calls, security, contextual content management, etc. Operators want to streamline their infrastructures through NFV and this predicated a consolidation of network functions onto industry-standard servers, switches, and storage hardware located in data and distribution-centers. The benefits of such consolidation and COTS migration include significant cost cutting, operational efficiency, as well as accelerated revenue generating innovation.

2. Leveraging Virtualization: Servers, Networks

The benefits of replacing proprietary-systems with virtualization are clear. Multi-core-threaded computing architectures transformed industry-standard hardware into highly concurrent machines, enabling COTS hardware to take on functions traditionally delivered using proprietary systems. However if complex carrier functions are to be virtualized this inevitably means (a) significant component based unbundling of both capacity and functionality locked today in monolithic systems, and (b) a method to dynamically assemble discrete functional components to elastic end-to-end services. Such "scatter-gather" rearrangement of carrier functionality needs to work on

commodity hardware and changing demand patterns. Dynamic component based models are already heavily used today by Internet providers, however mostly using software designed from the ground up along these lines. Mapping of users to functions to compute resources is based on common map-reduce infrastructure embedded in every module, library, and utility function call.

Although carriers do not typically write their code they can use state-of-the art server and network virtualization to achieve similar goals of flexibility, elasticity, and resiliency. Virtual machines (VMs) are used for porting existing functional code to server executable images, capturing the standard carrier (3GPP etc.) interfaces as well as proprietary operating systems. The

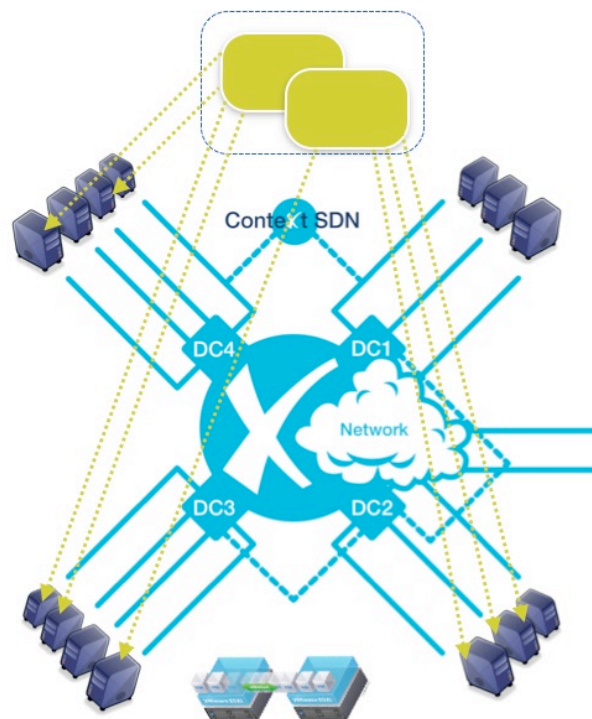


Figure 1: Functionality & Capacity - Decoupled & Unbundled

Hypervisors overlay priority control rings around standard CPUs, enabling multiple VM images to run concurrently on servers, reducing their span to the compute resources within the server each VM can use, enabling sharing with minimal cross interference. Virtualization overlay applies to the network as well.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
HotSDN'13, August 16, 2013, Hong Kong, China.
Copyright © 2013 ACM 978-1-4503-2178-5/13/08...\$15.00.

Having unbundled functions to small virtual units (scatter) we turn to their dynamic assembly, global coordination, and scaling. For this task (gather) we turn to network virtualization and Software Defined Networks (SDN). In order to perform this role SDN needs to map the right traffic flows, in the right sequence, to the right virtual functions. In order for that to happen, we must first enhance the SDN control model to overcome built-in non-linear scale limitations (similar ones to out-of-band circuit switching). We add to the open-flow interface, decoupling control and forwarding.. an overlay interface, decoupling control and forwarding topologies. The Overlay needs for global-lookup are realized using a mapping database, built-in and distributed by the Underlying IP network. The resulting solution is a virtual network that functions as a flow-mapping-fabric connecting virtualized network functions; balancing load among instances of classes, preserving affinity of subscriber traffic to associated functional instances, protecting states, and assembling end-to-end services.

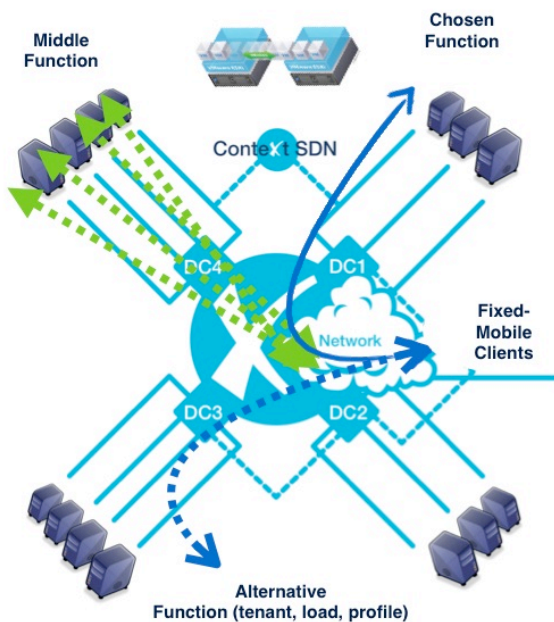
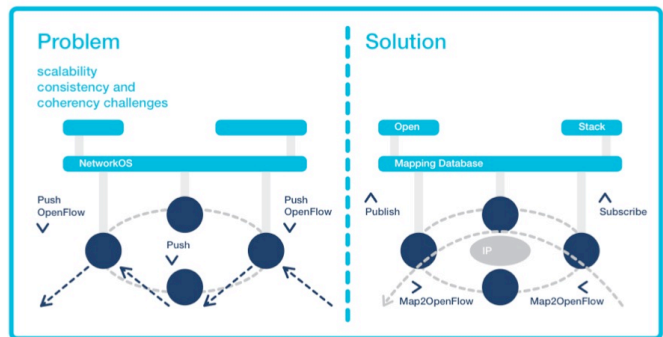


Figure 2: Dynamic Service Assembly by Mapping Flows

3.Overlays, Mapping, and Flows

To recap, in order to virtualize carrier functions we break monolithic systems to components running on virtual machines and we assemble these dynamically using SDN flow-mapping. We scale SDN control to be able to keep track of the potentially hundreds of millions of subscriber flows and their affinity-association to virtualized function instances. To do that we extend SDN control model to decouple not just control and forwarding, but also the control and forwarding topologies. This is done using overlays and results in the below SDN-NFV design characteristics:

- I. flow processing is done strictly at the edges of an overlay network which has only measured (Eco) awareness of topological underlay connectivity.
- II. flow decisions are done at the overlay network strictly by using mapping pull-publish-subscribe, a service enabled by the underlay connectivity.



- From:
 - Concentrated, hop-by-hop, push
- To:
 - Distributed, overlay, pull-pub-sub

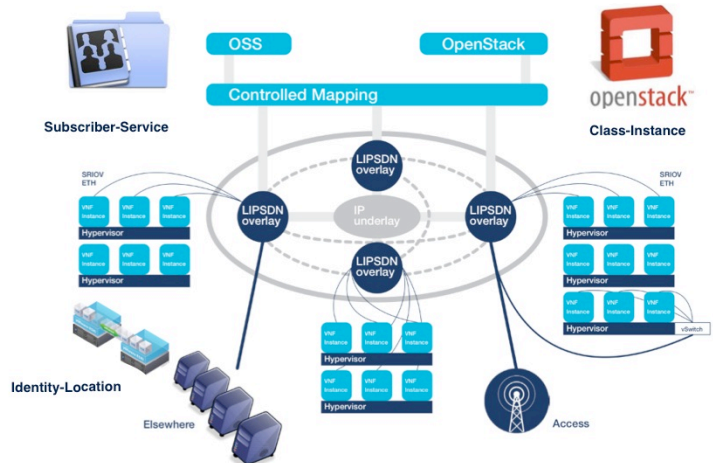


Figure 3: Enhanced SDN: NFV-LISP Overlay, IP+DB Underlay

The diagrams illustrates how RFC6830 LISP based SDN overlays can implement such interconnect for virtualized functions by surrounding data-center spines or national backbones. Such solutions have been deployed in production tier1 carriers over the past 18 months, dynamically connecting mobility, content and signaling functions to 10s of millions of users.

4.REFERENCES

- [1] Benitez, J., Bugenhagen, M., Khan, W., Chiosi, M., Clarke, D., Cui, C., Damker, H., Delisle, D., Demaria, E., Deng, H., Fargano, M., Feger, J., Fukui, M., Ivano, G., Kolias, C., Lopez, D., Loudier, Q., Manzalini, A., Matsuzaki, T., Michel, U., Minerva, R., Ogaki, K., Reid, A., Ruhl, F., Salguero, F. J. R., Sen, P., Shimano, K., and Willis, P. (2012). "Network Functions Virtualization." Presented at SDN and OpenFlow World Congress, 16 pp
- [2] Farinacci, D., Fuller, V., Meyer, D., and Lewis, D. (2013). "The Locator/ID Separation Protocol (LISP)." Internet Engineering Task Force RFC 6830, 74 pp.
- [3] Barkai, S. (2006). "EP 2047479: Distributed Edge Network," European Patent Register
- [4] Kim, C., Caesar, M., and Rexford, J. (2008). "Floodless in Seattle: A Scalable Ethernet Architecture for Large Enterprises." Princeton University, 14 pp.