Towards Cloud-Aware Policy Enforcement with Universal Cloud Classification as a Service (UCCaaS) in Software Defined Networks

Sebastian Jeuk
UCL & Cisco Systems
Department of Computer Science
London, UK & San Jose, USA
ucabsej@ucl.ac.uk

Gonzalo Salgueiro
Cisco Systems
Raleigh, USA
gsalguei@cisco.com

Shi Zhou
University College London (UCL)
Department of Computer Science
London, UK
s.zhou@ucl.ac.uk

***Abstract*—Network services are a critical component of today's networks. They apply critical functions (e.g. security, routing or quality of service) to traffic to enhance the network operators and application consumers experience. Today these services are inserted physically on the data-forwarding plane without providing much flexibility to deal with different traffic types or affiliations. Cloud Computing, however, demands policy enforcement on a per-Provider, per- Service and/or per-Tenant basis. In addition, there is an increasing need for dynamic transparent network chaining independent of the underlying transport infrastructure. We first introduce the concept of Universal Cloud Classification as a Service (UCCaaS). Followed by highlighting how it can be leveraged in conjunction with Network Service Headers (NSH) to address above challenges. UCC provides an addressing scheme to isolate traffic streams on a per-provider, per-service and/or per-tenant basis. To enable bi-directional policy enforcement in network functions we extend the UCC proposal by adding source and destination support. NSH is a way to steer network traffic dynamically across a set of network functions. We demonstrate the feasibility and advantages of our UCCaaS + NSH proposal with an example application, where a service chain defines Access Control Lists and traffic rate limiting on a per-Service and per-Tenant basis. Our proposal opens a door for a wide range of cloud-aware network services and functions.**

***Keywords*-Cloud Computing; SDN; Controller; Classification; XaaS; Universal Cloud Classification; Service Function Chain;**

I. Introduction

Cloud Computing is the new buzz word in the way compute, network and storage resources are deployed. It provides access to dynamic, flexible and scalable infrastructures and resources to deploy a variety of different applications. Network resources are thereby the crucial element to define and enforce functions to data in flight. With cloud computing, network functions have to adapt to handle dynamic and ever changing traffic flows. Classifying these cloud-specific traffic flows according to their service/tenant affiliation is key to fully provide cloud-aware policy enforcement.

Today, network functions are physically inserted into the data-forwarding path. Packets are directed through those network functions using statically defined transport mechanisms such as VLANs or policy-based routing techniques. This service insertion model tightly couples services with the existing physical network infrastructure not taking into account the demands and requirements of cloud services.

Network service functions, once introduced into the physical topology of a network, are not easily moved, removed or changed. Adding new network functions is only possible by re-designing the physical topology. Paul Quinn et. al. [14] calls these limitations the ”antithesis of highly elastic environments” perfectly describing the impact of today's service insertion model on cloud environments.

Classifying traffic in today's networks is typically handled on the transport layer using technologies such as VLANs, VxLANs or GRE tunnels amongst others. These approaches face critical limitations when leveraged in cloud deployments such as scale, ambiguity and lack of end-to-end isolation and transparency.

Recent research introduced Universal Cloud Classification as a way to identify and isolate cloud providers, services and tenants on the transport layer. It provides means to enable guaranteed unique cloud-specific classification on an end-to-end basis. The research around UCC proposes the usage of IPv6 extension headers as a possible vehicle of IDs that identify provider, service and tenant affiliation both inside and outside a cloud providers network.

The paper introduces a novel approach to enable cloud-aware policy enforcement by leveraging the UCC scheme and NSH. This solution tackles relevant issues seen around dynamic network function insertion and the application to cloud-specific traffic.

Here, we first introduce the technologies used to enable cloud-aware policy enforcement, UCC and NSH. This is followed by an overview of problem areas the solution tries to tackle to provide a more agile and flexible way of network function insertion. The solution itself is described using an traffic flow example, showing how cloud-enabled policy enforcement can be used to specify Access Control Lists (ACLs) and rate-limiting on a per-Service and per-Tenant basis in a cloud environment. To show the solutions feasibility the paper also outlines potential use-cases beyond the used security example.

II. Background

Before diving into the description of the solutions components we clarify some terms used in this paper.

- Cloud Service: The definition of the term ”service” is ambigious as multiple perspectives exist. In this paper, we use the term ”cloud service” to refer to an offering by an application hosted on a cloud providers compute and storage resources. However, the term ”Service Chain” refers to the linkage of multiple network functions.
- Network Function: A network function is a transit service offered by the network such as network security or IP connectivity.The network function applies and enforces policies to cloud service specific traffic flows.

A. Universal Cloud Classification

Universal Cloud Classification (UCC), as defined in [8], [7], [5], [2], [6], is a novel approach in identifying tenants, services and cloud providers on the network layer. The scheme introduces three IDs that are incorporated in an IPv6 extension header. This generic approach does not distinguish between source or destination specific UCC, which we will introduce in this paper.

UCC has been introduced to tackle limitations seen in current classification technologies used in cloud-enabled data centers. The most prominent isolation approaches within network environments include ”Virtual Local Area Networks (VLANs)”, ”Virtual extensible LANs
(VxLANs)” and ”Generic Routing Encapsulation (GRE)” Tunnels. These technologies face critical shortcomings such as scope limitations, extensive administrative overhead and topology lock-in. However, for Cloud Computing these technologies create a far bigger challenge namely the lack of cloud specific classification.

The novelty of the Universal Cloud Classification scheme can be summarized as follows: A hierarchical end-to-end classification scheme consisting of three IDs (Cloud/Service/Tenant) closely reflecting the internal structure of cloud environments. These IDs are carefully selected and defined to solve the classification challenges seen in Cloud Computing. The scheme can be succinctly characterized as a (1) hierarchical, (2) end-to-end, (3) optional, (4) flexible and extensible, (5) universal and (6) guaranteed uniqueness classification scheme.

B. Network Service Headers

Quinn et.al.[13] describes in their IETF draft a multitude of limitations seen in today's service function deployments. They discuss (1) Topological Dependencies, (2) Service Chain Constructions, (3) Application of Service Policy, (4) Per-Service (re) Classification, (5) Common Header Format, (6) limited End-to-End Service Visibility and (7) Transport Dependencies as the most prominent ones amongst others. For a more comprehensive overview of problematic aspects of today's service deployments refer to [15].

The authors of Network Service Header describe ”service chaining” as a construct ”to describe the deployment of composite services that constructed from one or more L4-L7 services using Software Defined Networking (SDN) and Network Function Virtualization (NFV) paradigms.” [14]

Based on these network function placement challenges Quinn et. al. [13] introduce the Network Service Header. The NSH is used to create a dedicated service plane by defining a data plane header. It is comprised of a base header, a service path header, 16 bytes of mandatory fixed context information and optional metadata fields. It is typically added between the original packet or frame and an outer network transport encapsulation (e.g. VxLANs). NSH is therefore transport agnostic and does not depend on a certain protocol.

The service path header contains a service path identifier (SPI) to identify the service path while the service index (SI) defines the location of a packet in a service chain. Figure **??** depicts a typical NSH flow traversing the chained network functions.

III. Problem space

The problem space is defined around the network function utilization in cloud environments and its limitations seen today. This paper does not claim to tackle specific issues in areas such as security but rather tries to solve how policies can be applied to cloud specific traffic. The solution provided tries to solve three distinct problems.

The first problem area depicts how network service chains and their functions classify interesting traffic. In today's networks this is typically done with legacy technologies that aren't fine-grained enough for cloud applications. Technologies used include deep packet inspection (DPI), basic layer 2, 3 and 4 classification and the Network-Based Application Recognition (NBAR) protocol. Each approach has it's own shortcomings due to the lack of cloud-specific information. These classification limitations are discussed in detail here [7] and [5].

Secondly, when using Universal Cloud Classification (UCC) in a cloud environment it is likely that not all network services are enabled or support UCC. This is a challenge for cloud providers relying on UCC to classify their traffic end-to-end. When a network service can't rely on UCC to classify traffic it must fall back to legacy classification approaches. These have limitations that make them less ideal for cloud environments.

Thirdly, when it comes to network service functions the services rely on classification and metadata details to apply policies to traffic. In a cloud environment metadata information often lacks the necessary level of correlation to be useful to tenants or services.

This idea exposes UCC classification details to service functions (SFs) and non-UCC-aware devices as well as leverages the flexibility of the NSH in order to provide useful correlation between UCC and other relevant data sets. This paper focuses on introducing cloud specific identifier metadata to be used in Service Function Chaining (SFC).

IV. Source and Destination Aware UCC

Before we start looking into UCC as a Service and how to leverage NSH to build a software defined environment, we here extend the generic UCC proposal. As previously mentioned UCC was originally defined to identify Cloud Providers, Services and Tenants without distinguishing between origin and destination details.

To complete this approach and to be able to define source and destination specific flow rules we define source- and destination-specific UCC. The format of the source and destination specific UCC information are defined on a per flow basis. Instead of adding duplicate information, source- and destination-specific UCC carries only the respective unique IDs while reusing others.

For example, for a flow that originates and completes within a Cloud Providers network, only service and tenant specific IDs are defined for the source and the destination. The provider ID is not duplicated as the same for both. Another example could be the communication between two tenants within the same service and the same provider. Here, only the Tenant-ID is defined for both the source and the destination.

Extending UCC with source and destination specific information allows applying both ingress and egress flow definitions and policies. This enriches the classification options while providing even finer grained policy applications.

Here, in this paper, we will use the source and destination specific UCC information to form a comprehensive software defined networking solution around UCC and NSH. Next, we introduce UCC as a Service as an approach to leverage IDs without relying on the underlying hardware to understand and interpret UCC information.

V. UCC as a Service

UCC was introduced as a classification mechanism on Layer 3 of the OSI Model, transported in IPv6 extension headers. These extension headers are defined to transmit customizable data, therefore forming them is reasonably cheap (in terms of TCP/IP stack processing overhead). However, inspecting these headers in hardware is difficult to achieve without re-designing already existing and proven ASICs. Leveraging the software defined networking (SDN) approach provides a way to use the incorporated IDs while maintaining the 5-tuple flow definitions.

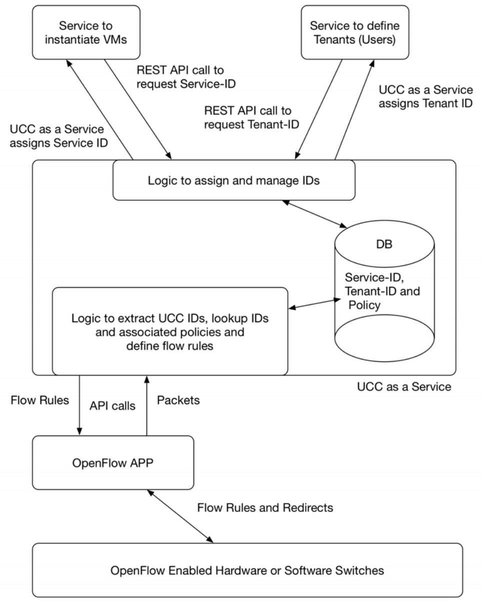
Service to instantiate VMs
Service to define Tenants (Users)
REST API call to request Service-ID
UCC as a Service assigns Tenant ID
REST API call to request Tenant-ID
UCC as a Service assigns Service ID
Logic to assign and manage IDs
DB
Service-ID, Tenant-ID and Policy
Logic to extract UCC IDs, lookup IDs and associated policies and define flow rules
UCC as a Service
Flow Rules
API calls
Packets
OpenFlow APP
Flow Rules and Redirects
OpenFlow Enabled Hardware or Software Switches

Figure 1. Schematics of UCC as a Service

Here, we introduce ”UCC as a Service” as a module in SDN-enabled cloud environments to leverage the UCC IDs. Instead of relying on every device within the environment to inspect and understand the UCC information, UCCaaS moves this logic out to a SDN controller. As an additional app offering south- and north-bound APIs, UCCaaS taps into already existing infrastructure applications used to define forwarding rules. In addition, it leverages the UCC IDs incorporated into the IPv6 extension headers, defines forwarding decisions on internal rules and pushes these out using the standardized 5-tuple flow rules. That way the logic around UCC can be enabled or disabled by simply adding another application to an SDN controller. The flow rules, defined by using the 5-tuple classification, can be defined without direct understanding of UCC ID’s on the hardware and/or software switches.

The UCC as a Service application is split into two operations. First it is used to manage the provider specific service and tenant IDs and maintains the data store of the assigned IDs. The IDs are handed out by the services that are used to instantiate VMs and define tenants for cloud services. The information is then used to define policies for flows within the cloud providers network.

VI. Our proposal on UCCaaS + NSH

As shown in the Background section both UCC and NSH can be used independently of each other. UCC, with its cloud specific identifiers, leverages IPv6 extension headers to transport cloud-ID, service-ID and tenant-ID information to isolate traffic flows. NSH provides the means to define network service chains independent of the underlying infrastructure.

With the introduction of UCC as a Service a workflow

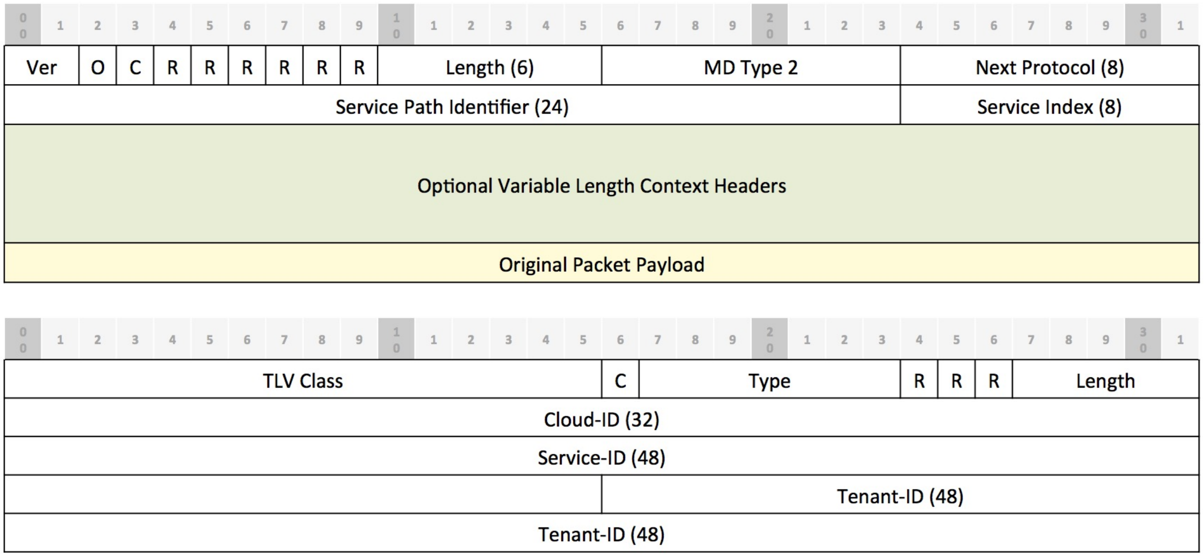

Figure 2. (Top) Generic NSH Type 2 Header with variable length context header; (Bottom) Example NSH Type 2 Header transporting UCC specific information

can be defined that leverages the UCC IDs and makes use of Network Service Headers to define function chains. In this section the authors outline how UCC can be embedded into the NSH metadata field to provide cloud entity visibility to network functions. Path selection and policy enforcement is then depict by introducing an example with a firewall and a traffic rate limiter. To outline how the proposal can be used in different applications this section also summarizes a set of potential use-cases.

The service classifier is the first component of the Network Service Header (NSH) architecture and determines which traffic requires services. It forms the logical start of the service path. With UCC as a Service, the traffic classifier is based around the functions offered by UCCaaS. It can be considered the first network function as part of a function chain. However, instead of classifying traffic going into the UCCaaS application all traffic is forwarded and then based on policies, flow rules are defined to forward traffic through network functions.

In addition to handling the classification of interesting traffic for network service chains UCCaaS also provides a mechanism to incorporate the UCC IDs into the network service header on-demand (if requested by a certain network functions). If that is the case, the UCC IDs are defined in either NSH type 1 or type 2 headers so that network functions in the chain can make use of these IDs to apply fine-grained policies.

The way NSH is handling service path flows is not changed when introducing UCC. The authors therefore won't further outline the underlying transport and how network elements handle NSH headers. For further details on NSH transport and network element NSH specific details refer to [3] [17].

As soon as the packet is received by the network function, the NSH is inspected and the metadata field is used to gather required cloud-specific isolation details. The details are then used to define and apply policies on a per-service and per-tenant basis. The application of policies is network function specific and is therefore independent of UCC + NSH.

It is important to note that UCC + NSH does not influence the network function applied to cloud specific traffic itself. It solely provides the means to get a fine-grained separation so that policies can be define to service and tenant needs.

With the introduction of UCC as a Service a workflow can be defined to enable UCC aware NSH in an SDN-enabled Cloud environment.

VII. An example application on cloud-aware ACL policy

The figure below illustrates how cloud-aware policy enforcement is realized using UCC and NSH.

1) The SDN controller is intercepting packets. The UCC as a Service app inspects packets and defines flow rules on internally defined policies (on a per Service and per Tenant basis). These flow rules are then pushed back out to the hardware/software switches using a protocol such as OpenFlow.
2) The traffic is then encapsulated with the NSH header, defining the SPI/SI combination.
3) Based on the details encapsulated into the NSH header in Stage 1 the appropriate transport encapsulation is imposed. The packets are then delivered to the first

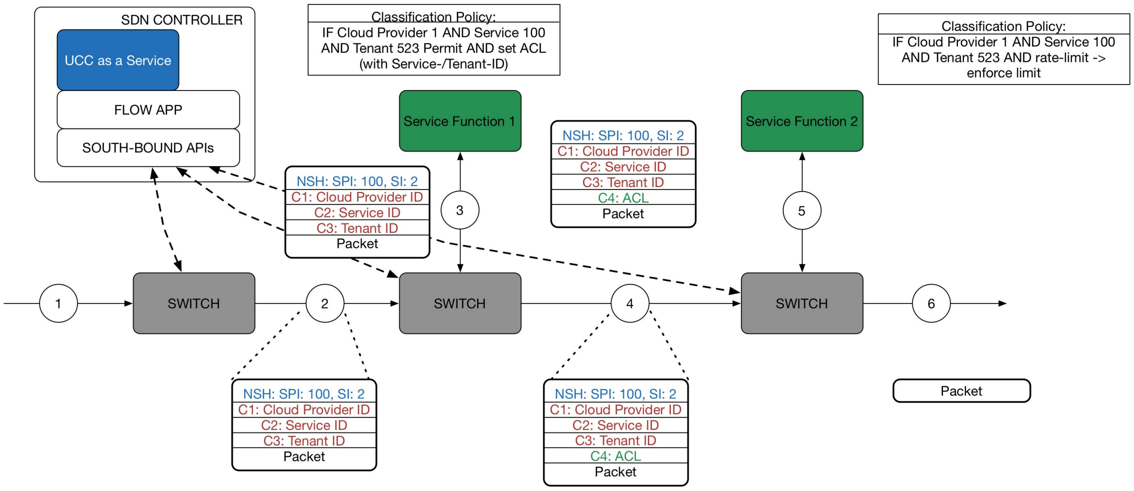

Figure 3. Network Service Chaining workflow with UCC as a Service

service function of the service chain (here called Service Function 1).

4) The packet traverses through Switch 1 and is forwarded to service function 1. The Service Function 1 will apply locally defined policies requiring per-Cloud-Provider, per-Service and per-Tenant enforcement. In our example Service Function 1 classifies the packets, defines the policies and enforces them. Here, we define a ACL that is permitting traffic originating from internal (Cloud Provider 1) Service 100 and Tenant 523.
5) After Service Function 1 operations are complete and NSH + packet are returned to the Switch. The switch uses the SPI/SI information to determine the next Service Function (here Service Function 2). It sends it off to Switch 2.
6) Service Function 2, after receiving the packet from Switch 2, applies its own classification and policy enforcement. It applies a rate-limit for traffic originating from internal (Cloud Provider ID 1) service 100 with tenant 523. It then enforces the defined rate limit.
7) After exiting the last service function of the chain the packet will be forwarded to its original destination.

As shown in this example cloud-aware policy enforcement using UCCaaS + NSH is a novel approach to first classify interesting traffic and then define and enforce policies accordingly while maintaining the required dynamics necessary for cloud environments. In the next section we highlight some example applications of cloud-aware policy enforcement.

The concept around Software Defined Networks is critical in this architecture and workflow to allow dynamic and on-demand installation and removal of policies in the OpenFlow enabled switches. These flows and policies are used to redirect interesting flows to the service functions as defined by the network function chains.

These function chains can be defined on a per Tenant and/or per Service basis.

VIII. Other Potential Applications

Below is a small subset of possible use-cases intended to convey the power and value of the proposed novel approach. The examples listed are not unique to UCC but rather show how UCC-specific classification enhances their application. UCC enables their correlation to Cloud Provider, Service and Tenant IDs. Without UCC the data can't be correlated to a specific user of a Service hosted by a Cloud Provider.

- UCC Identifiers + IMEI Number - The IMEI (International Mobile Station Equipment Identity) Number is used to uniquely identify GSM, UMTS and LTE mobile phones. A Mobile Service Provider running a private Cloud can have multiple IMEIs per tenant (user). This user can leverage multiple of the services offered by the Cloud. To define IMEI and tenant-specific policies within the network function chain, metadata identifying both the tenant and the IMEI Number is required. Including the UCC Identifiers and the IMEI number (or numbers) in the metadata field of the NSH enabled network service provisioning on a per-service/per-tenant and per-IMEI number basis.
- UCC Identifiers + session identifier (SIP, H323) - Session Initiation Protocol (SIP) or H.323 are protocols used to establish voice and/or video sessions between two endpoints. These protocols use identifiers to uniquely identify multimedia sessions end-to-end.

These identifiers can be carried in the NSH. A Cloud Provider hosting a voice/video application (Cloud Service) can have multiple tenants. Each tenant (user) can establish multiple SIP/H.323 session at a given time. Including the UCC Identifier plus the correlated session identifiers allows applying policies per SIP/H.323 stream owned by a certain Cloud tenant per network function in the network service chain.

- UCC Identifiers + geo-location - Geolocation is used to pinpoint the exact geographical location of an IP connected object such as a mobile phone. Geolocation information is useful in networks to apply policies based on geo-location. For example, certain traffic flows originating in Germany require the application of network services that reflect German laws. Other flows are more specific and local to the US or the UK. These examples show how geo-locations can be used with NSH. Correlating the geolocation details to Cloud-Service and Cloud-Tenant identifiers allows location-specific policies to be applied within the network service chain on a per-service and/or per-tenant basis.
- UCC Identifiers + SLA - Service Level Agreement details are a critical component of service offerings in cloud environments. Cloud Providers are keen to have the means to fulfill the offered SLAs on all levels of a Cloud Environment. With the insertion and correlation of Cloud Identifiers and SLA information into the Network Service Headers service functions can define policies according to SLA agreements on a per-service/per-tenant basis.
- UCC Identifiers + NAT - Network Address Translation (NAT) is a L3 network function that translates internal IP addresses to globally routable addresses. Applying NAT rules on a per-service/per-tenant basis is a key advantage for Cloud Providers and can be enabled by incorporating NAT details per Cloud-ID, Service-ID and Tenant-ID.
- UCC Identifiers + Routing - Defining forwarding decision on a per-Service/per-Tenant basis is a novel approach of defining routing in a Cloud Environment. Incorporating forwarding information and the correlated Cloud Identities in the Network Service Header enables service functions to forward traffic based on Cloud entity requirements.
- UCC Identifiers + Interface Identifier - The Interface Identifier (IID) is part of the IPv6 stack and defined in RFC 4291. It is used to uniquely identify interfaces on a link and typically incorporated into an IPv6 unicast address. Correlating the IID with the Cloud Identifiers allow network services in a network function chain to apply policies per-service/per-tenant flow coming from a particular VM. This is a novel approach in defining network policies as it allows to slice VM traffic and affiliate each slice to a service and tenant tuple.
- UCC Identifiers + Storage Information - Correlating the Cloud Identifiers with Storage information such as WWNs or WWPNs allows creating network function chains with tenant/service specific storage policies
- UCC Identifiers + VPN - VPNs typically require certain details to setup tunnel endpoints. Conveying VPN details in correlation with Cloud identifiers allows network functions to setup VPN tunnel with Service/Tenant specific needs/characteristics.
- UCC Identifiers + Security details - Correlating security (such as keys, encryption strength, etc.) and Cloud Identifier details allows specifying network security functions specific to a Tenant, Service or Cloud Provider
- UCC Identifiers + QoS - NSH can be used to identify interesting flows and signal flow characteristics as a means of consistent treatment of traffic and maintaining user experience across domains. Correlating these flows with UCC identifiers allows Cloud Providers the ability to offer differentiated treatment on a per-tenant and/or per-service basis.

A useful outcome of this proposal is that it offers an alternate mechanism (i.e., NSH) of exposing/transporting UCC classification information. Consequently, this makes network functions aware of these IDs without requiring UCC awareness (i.e., the underlying transport technology used to transport UCC information)

IX. Related Work

With cloud computing becoming more and more prominent research is underway to investigate the usability, feasibility and placement of virtualized network functions (NFV) in the cloud [20]. The majority of research in this area however focuses on NFV placement, chaining and scaling while not investigating the question of how to apply policies on a per-service, per-tenant basis. Yu et. al.[20] outlines some of the challenges seen with NFV, including network function deployment, policy enforcement, performance guarantee and performance management. They propose a SDN based architecture as a basis for further extensive research on NFV. Based on SDN this proposal requires a centralized control plane causing increased control traffic overhead.

Callegati et.al. [4] introduces layer 2 and layer 3 based alternatives to other virtual function chaining approaches. Their proposal is based on either a typical Layer 2 domain where all services are interconnected on a single Layer 2 domain (typically transparent to users). The Layer 3 approach separates network functions into different broadcast domains. Traffic steering is therefore based on ”legacy” network concepts. Traffic classification is based on Deep Packet Inspection introducing latency and large traffic processing overheads. The authors of this paper believe that the proposal presented by Callegati et. al. [4] is not feasible to deal the ever increasing and highly dynamic workloads of cloud environments.

Bagaa et.al. [1] propose leveraging virtualized instances of Packet Data Network Gateways (PDN-GW) and it's placement in a carrier cloud. They thereby investigate the number of required Gateways and the best gateway selection based on geographical locations, application type and traffic load balancing. Oechsner et. al [12] investigate the relevance of VM placement in the context of NFV deployments. They discussed the requirement of a cloud management system to handle and manage instance placements. The proposed algorithm defines the best placement for instances and communicates the result to the management system provisioning the VMs accordingly. The management system, as the central entity, takes care of placing VMs on compute resources with available resources otherwise requesting a placement suggestion from the algorithm. Here, Oechsner et. al. [12] rely on the management infrastructure of a cloud environment to assure instances gets placed correctly without investigating the need for service chaining. The algorithm does not take into account the often crucial need to chain multiple network functions, therefore risking sub-optimal VNF placement.

Shameli-Sendi et.al. [16] propose an algorithm to address the placement optimization for ordered sequence of virtual security appliances based on the Traveling Purchaser Problem.

Mehraghdam et. al [10] propose a model to formalize chaining of network functions based on a context-free language. Their model takes into account multiple sides and the requirements of tenants and operators of a network. They accurately state that chaining requirements and network function placement objectives are often not easily combined. This is an important consideration when discussing chaining while network functions are distributed across large virtualized cloud environments. Based on the findings of Mehradghdam et.al. [10] and Luizelli et. al. [9] we suggest further research around NSH, its chaining capabilities and how NFV placement can be considered.

Current research [1], [12], [16] but also [11], [18] mainly focus on the question of function placement rather than it's chaining in cloud environment. Also, very limited research is available that investigate the problem of applying and enforcing policies on a per-provider, per-service and per-tenant basis critical in Cloud Computing.

Based on the research shown the authors believe that the proposal in this paper is a novel and key advancement around network function chaining and policy enforcement specific to cloud environments.

X. Prototype Discussion

Here, we proposed a novel way to leverage UCC and its implementation as a Service in a Software Defined Network to enable NSH with cloud specific classification and policy enforcement. To further evaluate this conceptual design we are currently working on a prototype using the SDN Controller Floodlight. The UCC solution itself has already been implemented and evaluated in a prototype like environment [5]. To evaluate this particular use-case we propose a software defined approach eliminating any dependency on hardware or network operating systems (NOS). Implementing UCC and its application with UCCaaS in a SDN based environment allows intercepting packets, inspecting them for the added identifiers and define OpenFlow based forwarding decisions.

Floodlight is a modular Open SDN platform that supports hardware and software in multivendor environments. We will leverage some of the already existing modules defined for Floodlight as a foundation for a newly defined UCC as a Service application. This app makes use of the modules necessary to get packets from the data plane and define flows in OpenFlow enabled software switches such as the OpenVSwitch (OVS). We will be using OpenFlow in version 1.3 to configure rules to forward traffic according to the service function chains defined. This solution will be run in a virtualized environment using either a public cloud compute offering or a dedicated hardware server. The network environment will be based on mininet, a tool enabling orchestration of OVS based topologies and compatible with the floodlight SDN solution.

We plan to investigate several aspects of the proposed UCC as a Service and NSH solution. Firstly we investigate the feasibility of the overall solution and its practicality using OpenDayLight and a SDN approach. Next, we plan to run several empirical evaluations to assure the impact of UCC and its implementation as a Service is minimal and a proper solution for the defined problem space.

XI. Conclusion

In this paper the authors introduced a novel approach towards cloud-aware policy enforcements in data centers using Universal Cloud Classification and Network Service Headers. Both technologies on their own are currently work in progress and are described in either research papers, IETF drafts, RFCs, patents or Early-Field-Trial (EFT) products [19].

Universal Cloud Classification (UCC) is an approach to eliminate the limitations seen in current classification technologies used in cloud environments (such as VLANs, VxLANs, GRE, etc.). By introducing three distinct, cloud-specific identifiers (Cloud-ID, Service-ID and Tenant-ID), UCC enables fine-grained, end-to-end and guaranteed isolation for cloud provider traffic. Here, we extended the proposal previously defined to support source and destination specific information to enable bi-directional policy enforcement in network functions.

UCC and its implementation as a Service is a novel way to circumvent adoption and hardware limitation of UCC while also enabling an on demand workflow.

Network Service Headers (NSH) are outlined by multiple IETF drafts and RFCs to provide means to dynamically and on-demand create service function chains. The header incorporates a service path classified traffic is forwarded through. A service classifier is leveraged to perform the initial classification of traffic interesting to the service chain. The NSH also maintains a variable length metadata field allowing the incorporation of relevant information.

In combining UCC and NSH the proposal enables policy enforcement in cloud environments for different applications. The authors highlight a path selection and policy enforcement example to show how Access Control Lists can be defined on a per-Service and per-Tenant basis.

This proposal can be considered as early-stage work. To further the idea and to evaluate its feasibility in environments the authors propose the development of a prototype. The prototype would allow gathering evaluative measurements to compare the idea to the current network function implementation. In addition, it can be leveraged to gauge the impact on real cloud provider environments.

To summarize, embedding UCC information into the NSH metadata field provides the foundation for a multitude of applications both inside and outside of cloud provider networks. The fine-grained classification approach offered by UCC enables policy enforcement relevant to cloud entities while providing a flexible and dynamic service chaining environment with NSH.

References

[1] M. Bagaa, T. Taleb, and A. Ksentini. Service-aware network function placement for efficient traffic handling in carrier cloud. In *Wireless Communications and Networking Conference (WCNC), 2014 IEEE*, pages 2402–2407, April 2014.

[2] F. Baker, C. Marino, I. Wells, R. Agarwalla, S. Jeuk, and G. Salgueiro. A model for ipv6 operation in openstack. draft-baker-openstack-ipv6-model-02, February 2015. Internet-Draft.

[3] G. Brown. Service chaining in carrier networks @ONLINE, Feb. 2015.

[4] F. Callegati, W. Cerroni, C. Contoli, and G. Santandrea. Dynamic chaining of virtual network functions in cloud-based edge networks. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–5, April 2015.

[5] S. Jeuk, G. Salgueiro, and S. Zhou. Universal cloud classification (ucc) and its evaluation in a data center environment. In *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, pages 469–474, Dec 2014.

[6] S. Jeuk, G. Salgueiro, and S. Zhou. A novel approach to classify cloud entities: Universal cloud classification (ucc). In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*, May 2015.

[7] S. Jeuk, J. Szefer, and S. Zhou. Towards cloud, service and tenant classification for cloud computing. In *Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on*, pages 792–801, May 2014.

[8] S. Jeuk, S. Zhou, and M. Rio. Tenant-id: Tagging tenant assets in cloud environments. In *Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on*, pages 642–647, May 2013.

[9] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary. Piecing together the nfv provisioning puzzle: Efficient placement and chaining of virtual network functions. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 98–106, May 2015.

[10] S. Mehraghdam, M. Keller, and H. Karl. Specifying and placing chains of virtual network functions. In *Cloud Networking (CloudNet), 2014 IEEE 3rd International Conference on*, pages 7–13, Oct 2014.

[11] H. Moens, B. Hanssens, B. Dhoedt, and F. De Turck. Hierarchical network-aware placement of service oriented applications in clouds. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*, pages 1–8, May 2014.

[12] S. Oechsner and A. Ripke. Flexible support of vnf placement functions in openstack. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–6, April 2015.

[13] P. Quinn, J. Guiarch, S. Kumar, and M. Smith. Network Service Header. IETF Standards Track, Feb. 2015.

[14] P. Quinn and J. Guichard. Service Function Chaining – Creating a Service Plane Using Network Service Header (NSH). online, 2014.

[15] P. Quinn and T. Nadeau. Problem Statement for Service Function Chaining. RFC 7498(Draft Standard), 2015.

[16] A. Shameli-Sendi, Y. Jarraya, M. Fekih-Ahmed, M. Pourzandi, C. Talhi, and M. Cheriet. Optimal placement of sequentially ordered virtual security appliances in the cloud. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 818–821, May 2015.

[17] V. Suazo and S. Dasgupta. Network service chaining solution (2015 san diego). Cisco Live San Diego 2015, 2015.

[18] K. Suksomboon, M. Fukushima, M. Hayashi, R. Chawuthai, and H. Takeda. Lawnfo: A decision framework for optimal location-aware network function outsourcing. In *Network Softwarization (NetSoft), 2015 1st IEEE Conference on*, pages 1–9, April 2015.

[19] C. Systems. Enabling service chaining on cisco nexus 1000v series, 2013.

[20] R. Yu, G. Xue, V. Kilari, and X. Zhang. Network function virtualization in the multi-tenant cloud. *Network, IEEE*, 29(3):42–47, May 2015.