A Novel Approach To Classify Cloud Entities: Universal Cloud Classification (UCC)

Sebastian Jeuk
Cisco Systems, San Jose, USA
& UCL, London, UK
Email: ucabsej@ucl.ac.uk

Gonzalo Salgueiro
Cisco Systems
Research Triangle Park, USA
Email: gsalguei@cisco.com

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

***Abstract*—One of the fundamental requirements of Cloud Computing is the capability to provide scalable, transparent and isolated networks. This is achieved by using L2 segmentation via 802.1Q VLANs or overlay approaches such as 802.1ad, VxLAN, "Stateless Transport Tunneling" (STT) or "Network Virtualization using Generic Routing Encapsulation" (NVGRE). All of these technologies struggle to provide the required level of scalability, flexibility, performance and network isolation within a Data Center. Research efforts in the area of classification have fundamentally approached these challenges by introducing identifiers for segmentation or providing overlay solutions to tunnel traffic. However, these research approaches are too specific without tackling the actual Cloud Computing classification challenges. Here, we investigate classification approaches with the goal of introducing a scalable, optional, hierarchical, end-to-end and transparent Layer 3 provider, service and tenant isolation scheme. This proposal addresses major challenges and limitations of current cloud classification schemes by offering these five advantages: (1) hierarchical end-to-end classification, (2) transparency to upper-layer protocols, (3) optional for en-route and endpoint evaluation, (4) flexibility, and (5) improved performance over current overlay technologies. The solution proposal will be implemented and evaluated based on its feasibility, functionality, performance and usability in cloud-related use-cases.**

***Keywords*-Service isolation, Tenant isolation, Provider isolation, Data Center, Classification, Cloud Computing**

I. Introduction

One of the fundamental requirements of Cloud Computing is to provide scalable, transparent and isolated networks. Segmentation is a way to distinguish entities on the network. Most prominently known by virtual LANs, segmentation has originally been defined to isolate entities by their organizational affiliation. However, classification in Cloud Computing is more complex and requires separation not by departments but by Cloud Providers, Services and Tenants. Cloud classification, therefore, can be defined as a way to distinguish cloud provider entities by their service and tenant affiliation. The way isolated networks are affiliated in Cloud environments demands new classification approaches to cope with characteristics specific to Cloud Computing.

A key issue in Cloud Computing is the context-based classification of service and tenant assets across different boundaries. Classification is used to implement policies that define the who, what, where, when and how. If we cannot identify and isolate the who on the network layer, we cannot define and implement any of the others. Current Layer 2/3/4 attributes are no longer sufficient to uniquely and in a uniform way identify the who in a cloud-enabled data center network.

Cloud Computing comes with many different definitions for their applications and users. Here, an application within a Cloud Provider is called a service. However, the Cloud Providers network cannot distinguish services from services running on top of other services. Hence, a service run on top of another service can be considered a tenant of that particular service. A user or consumer of a particular service can also be considered a tenant (in addition, OpenStack uses the word tenant as a way to describe projects). These ambiguities make it extremely difficult to uniquely identify services and their tenants in cloud environments.

Classification in Cloud environments is achieved today using legacy L2 segmentation via 802.1Q VLANs or overlay approaches such as 802.1ad, VXLAN, STT or NVGRE. All of these technologies struggle to provide the required highly scalable, flexible, performant and isolated networks within a Data Center while being transparent to network entities and applications.

The objective of this work is to determine the current shortcomings caused by the lack of adequate cloud-specific classification. Furthermore, the focus is to design, evaluate and implement a solution that provides classification beyond typical network boundaries and enables currently impossible use-cases both inside and outside a cloud environment. Our goal is a scalable, universal isolation of cloud entities that is transparent to current technologies and state-of-the art hardware.

We aim to address current issues under consideration in Cloud open source development and standardization organizations (e.g.: OpenStack, etc.) as well as in broader Internet Standards Development Organizations (SDOs), such as the Internet Engineering Task Force (IETF). These efforts validate the relevance, criticality and timeliness of our research.

II. Related Work, planned contribution and the research significance

Here, we highlight current research and state-of-the-art technologies used to classify traffic and isolate entities in cloud environments. We highlight their shortcomings, perform a gap analysis and demonstrate how this thesis contributes to advancing the research field of classification in Cloud Computing.

A. Related Work

Segmentation and network isolation has been a critical topic for networks since the introduction of IEEE 802.1q VLANs. Since then many different practical approaches have been introduced such as IEEE 802.1ad, VLAN, STT or NVGRE. In addition, there is extensive research looking into techniques to separate and isolate data both on the network layer and the application layer. Uniform Resource identifier (URI), Virtual Hosting in HTTP, Server Name Indication (SNI) or the Security Group Tag (SGT) defined as part of Cisco's Identity product suite are all example of attempts to solve certain classification challenges. Khoudali et al. [1] and Benzidane et al. [2] propose a Frame Tag header based on md5 hash values to identify tenants and services. These IDs are added in front of the payload and therefore require a change of the TCP/IP stack, which is not feasible. Content-Centric Networking (CCN), closely aligned to our research focus, identify the content or data residing on the host instead of the host itself. This leads to very large routing tables.

B. Gap Analysis

Many of the required classification characteristics remain unsolved by current research. Existing solutions or research proposals are limited in the way they approach classification in networks. Many approaches are focused on isolating tenants on the application layer but not addressing the need for a more generic and application-independent approach. Others are transparent to upper-layers but are not scalable or too localized to be useful in large cloud environments. Yet others introduce overlay approaches whereby traffic is tunneled across the network and therefore eliminating the possibility to use classification en-route for use-cases such as QoS, Security or Routing. In summary, current classification mechanisms are not suitable for cloud environments. Comprehensive cloud-specific approaches have not yet been investigated, proposed or developed.

C. Planned Contribution

This thesis contributes towards the research field of cloud classification by defining a distinct set of characteristics for a comprehensive isolation approach. Followed by the design of a hierarchical, end-to-end classification approach and it's evaluation is the development of Cloud-critical use-cases. Preliminary evaluation results show that the proposed implementation is feasible, compatible with state-of-the-art hardware and better performant than current overlay approaches such as VxLAN.

We argue that the novelty of UCC is based on the approaches' hierarchical and cloud specific design. The scheme consists of three IDs (Cloud/Service/Tenant), which are more consistent with cloud infrastructures and architectures than traditional IP endpoint-based mechanisms. These IDs are carefully selected and defined as a complete data set to solve the classification challenges and inadequacies currently seen in Cloud Computing.

D. Significance

The growing scale requirements of services and their tenants places the classification and isolation of cloud entities at the heart of the design of cloud infrastructures.

The Universal Cloud Classification (UCC) concept is designed with the criteria identified in Section III-A for a good classification in mind. The proposed approach, therefore, can be considered a long-term solution that tackles some fundamental problems of Cloud Computing. Its novelty is based on the hierarchical ID structure reflecting a Cloud Environment consisting of a Provider, Services and Tenants. The identity scheme is designed to provide guaranteed uniqueness globally.

The current work on IETF Internet-draft [3] highlights the proposals significance for industry related use-cases. The draft defines a use-case to authenticate communication between Virtual Machines (VMs) in an OpenStack-based private Cloud.

E. Scope

This thesis focuses on the conceptual design of a hierarchical classification model reflecting cloud architectures and supporting cloud characteristics. In addition, it looks into the evaluation and definition of critical use-cases both inside and outside cloud providers. Extensive interoperability studies and the comprehensive evaluation of introduced use-cases is considered outside the scope of this thesis.

III. Research Accomplishments

Here, we highlight our research accomplishments to date, define the desired classification characteristics and outline the Universal Cloud Classification (UCC) approach and its evaluation. In [4] we outlined a way to extend the scalability of IEEE 802.1q by introducing a Tenant-ID on layer-2. That allows network segmentation by Tenant and the associated VLAN set. Here, each Tenant can leverage all 4096 available VLANs without clashing with other tenants. This work is considered preliminary and therefore not further discussed here. Papers [5] and [6] define the Universal Cloud Classification proposal, show how the identifiers (IDs) are build and highlight the advantages of the proposed approach. In addition, [6] outlines the evaluation of the UCC proposal using a proof of concept environment based on state-of-the-art hardware. The mechanisms required to understand

the new approach are developed and run on a Linux-based container. The proposal is currently discussed in an IETF Internet-draft [3].

A. Desired Classification Characteristics

Before designing a new classification approach to address the shortcomings of current segmentation approaches for Cloud environments, it is critical to define desired criteria and properties for comprehensive classification. Based on the nature of Cloud Computing it is obvious that an isolation technique needs to be scalable, flexible, optional and transparent to services. In addition, we argue that end-to-end isolation of providers, services and tenants is key to not only solve segmentation challenges of cloud entities within but also outside cloud environments. From an implementation standpoint a new classification approach should be compatible with current hardware to guarantee easy adoption. Looking at performance criteria for a good classification scheme we postulate that limited additional overhead for superior classification and isolation capabilities is a worthwhile tradeoff provided performance is equal or better than current segmentation approaches, such as VxLAN. To summarize, a new classification approach needs to be designed with Cloud Computing characteristics in mind to be successful.

B. The UCC Concept

The UCC proposal introduces three identifiers: (1) Cloud-ID (4 bytes), (2) Service-ID (6 bytes), and the (3) Tenant-ID (6 bytes). The structure of all three IDs is based on the Document Object Identifier (DOI) [7] scheme.

The Cloud-ID is used to identify a Cloud Provider globally. We propose a registry service similar to DNS to manage IDs, their assignments and purge. The ID is split into two sections; (1) defining the registry location while (2) provides a unique Cloud Provider number in that region.

The Service-ID identifies a service within a Cloud Provider and encompasses three sub-IDs defining the Data Center, Service and an optional field. This ID only has significance within a Cloud Providers network and is assigned per cloud environment. The Data Center value can be used by the Provider to specify where the Service is predominately running. This will be useful to classify services according to their geographic location (for example to adhere to laws). The optional field can be used as needed by the Cloud Provider and does not have any pre-defined structure. A potential use-case for this field is the incorporation of path requirements to enable Internet Routing of Cloud-Data.

The Tenant-ID is defined as a 6-Byte number with cloud provider significance. It is used to identify a consumer of a Service offered by a Cloud Provider.

The overall IPv6 extension header is 22 bytes in size encompassing the three IDs plus 2 additional bytes per ID of flags to define en-route behavior.

C. The UCC Implementation

For the implementation of the UCC proposal we suggest the use of the existing optional header definitions of IPv6. Here, we propose to use the hop-by-hop extension header for en-route inspection. Alternatively, the destination option header can be used if UCC is only relevant for the endpoints of a conversation. IPv6 extension headers are standardized and are therefore part of the specs for IPv6. This is advantageous to incorporate classifiers without modifying the TCP/IP stack.

```
00000    60 73 5c df f3 3c 00 00 06 1e 43 9f 86 dd 60 00
00016    00 00 00 30 3c 40 20 01 ab ab 00 00 00 00 00 00
00032    00 00 00 00 00 01 20 06 ab ab 00 00 00 00 00 00
00048    00 00 00 00 00 02 3b 02 00 04 12 12 12 12 00 06
00064    ab ab ab ab ab ab 00 06 cd cd cd cd cd cd cc 0e
00080    3d 1b 2c 04 53 96 49 78 69 60 00 00 00 00 de fc
00096    04 54 8d 0c 74 73
```

Figure 1. **Example of an IPv6 packet with UCC identifiers:** The Cloud-ID is 12:12:12:12, the Service-ID is ab:ab:ab:ab:ab:ab and the Tenant-ID is cd:cd:cd:cd:cd:cd. In addition, each ID is preceded by a 1 byte optional and length field. Overall, this equals to a 22 byte long extension header.

The proposed IPv6 extension header is one possible implementation. We posit that this approach is most versatile and enables many use-cases both inside and outside a cloud environment. With the increase in MTU potential challenges arise in discovering the correct PATH MTU.

D. The UCC Evaluation

The evaluation of our UCC proposal is separated into three stages. Here, we discuss the first two stages with the third part being discussed in the Future Work section. In stage one we focus on evaluating the chosen implementation of UCC using an IPv6 hop-by-hop extension header. The second stage shows feasibility and performance results while also highlighting the billing use-case.

We designed a typical data center environment using state-of-the-art physical hardware to gather the most accurate results possible. The code to both capture and analyze the UCC header has been written in C and runs on an embedded linux environment on-top of the network entities. The IPv6 hop-by-hop extension header is defined based on the outline of the UCC schematics, including flags to define en-route behavior.

IV. Discussion

In the following paragraphs we discuss some of the challenges we see with the introduction of an identification scheme on Layer 3. Here, we only briefly highlight them, for further details refer to [5] [6].

The UCC schematics propose a 4byte/6byte/6byte identification tuple. One question arises on how scalable those IDs are within growing Cloud Environments. We argue that with the current growth rate the ID sizes are sufficient for the next 10+ years. While scalability is a concern, the UCC size needs to be balanced between scalability requirements and bandwidth/Path MTU impact. With this in mind we conclude that the defined sizes provide enough capacity for future

growth without consuming unnecessary bandwidth. With its modular approach UCC provides the possibility to extend IDs to cope with increased scale requirements of the future.

Our preliminary results show that UCC is compatible with current data center hardware. The impact on performance is only minimal compared to a typical IPv6 TCP packet. Here, we see a 2% degradation in performance with a 22byte header increase.

Goodput is a calculated value that defines the real application throughput without considering Layer2-7 header information. Our evaluation shows that UCC only decreases the Goodput by 2% compared to a typical IPv6 TCP header. However, in comparison to state-of-the-art classification techniques, such as VXLAN, UCC is improved by 2%. We therefore argue that UCC does not impact Cloud Provider network operations.

Security, privacy and confidentiality are sensitive topic in Cloud Computing. However, we argue that UCC does not increase any additional security or privacy considerations compared to existing technologies. Well defined state-of-the-art security-specific protocols can be readily used to address these concerns.

UCC enables highly-desired use-cases both inside and outside of cloud environments. Those include (but are not limited to) security, Quality of Service, Cloud Data Routing, per-Service/-Tenant billing,

To conclude, advantages of UCC can be highlighted as (1) globally unique, end-to-end Cloud classification,(2) solves the ambiguity in Cloud data centers, (3) optional, can be ignored by intermediate devices, and (4) enables use-cases inside and outside cloud environments to satisfy business requirements.

V. Future Work

To this point our research focused on the introduction of a novel hierarchical classification approach enabling fine-grained isolation of provider, service and tenant entities. Furthermore, we introduced a use-case empowered by the UCC approach. Billing is a critical capability for Cloud Providers and has previously been limited to coarse metering of the usage of compute and storage resources. With UCC, providers can now accurately meter the network usage per tenant. This use-case demonstrates the feasibility of our proposal within a Cloud Provider's network.

Completion of this research involves finalizing UCC use-case(s) outside of a cloud infrastructure. We plan to introduce an approach for Internet routing of Cloud Data leveraging UCC and potentially Segment Routing. Services offered through Cloud Providers often have different path requirements, such as latency, end-to-end delay, MTU or bandwidth. Today, it is a challenge to define a service-specific path through BGP Autonomous Systems. Internet Service Providers have no means to distinguish traffic from Cloud Providers according to their Cloud Service affiliation. UCC provides the missing link between services and their flows that can then be leveraged by upper-layer routing protocols.

There are several problems we have to address to complete the aforementioned research. First, using the IPv6 based UCC approach we have to investigate any potential pitfalls using IPv6 across the Internet. With legacy routers still being used in some parts of Service Provider networks, potential incompatibility issues may arise. Second, we have to investigate the feasibility of Segment Routing and how UCC can introduce service-awareness into BGP Autonomous Systems.

VI. Conclusion

Here, we propose a novel way to segment entities in Cloud environments. Based on the lessons learned from current L2 segmentation approaches and overlay techniques we defined a hierarchical classification concept.

UCC is fundamentally different from other currently used or proposed segmentation ideas. Instead of applying classification to a subset of entities in an environment or specific to a certain application, UCC is defined to be transparent and independent to upper layer applications. Additionally, UCC is defined in a hierarchical manner that reflects the Cloud Computing definition of Providers, Services and Tenants. This flexibility allows for seamless future extensions such as introduction of support for service-stacking or inter-cloud service-/tenant-isolation.

References

[1] S. Khoudali, K. Benzidane, and A. Sekkaki, "Inter-vm packet inspection in cloud computing," in *Communications, Computers and Applications (MIC-CCA), 2012 Mosharaka International Conference on*, Oct 2012, pp. 84–89.

[2] K. Benzidane, S. Khoudali, and A. Sekkaki, "Autonomous agent-based inspection for inter-vm traffic in a cloud environment," in *Internet Technology And Secured Transactions, 2012 International Conference for*, Dec 2012, pp. 656–661.

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

[4] 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*, May 2013, pp. 642–647.

[5] 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*, May 2014, pp. 792–801.

[6] 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*, Dec 2014, pp. 469–474.

[7] N. Paskin, "Syntax for the digital object identifier," ANSI/NISO Z39.842000, 1999, national Information Standards Organization Staff and National Standards Information Organization.