SDN-based Cloud Computing Networking

Siamak Azodolmolky, Philipp Wieder, Ramin Yahyapour
Gesellschaft für Wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG
Am Faßberg 11, 37077 Göttingen, Germany
Tel: (49) 551 2011510, Fax: (49) 551 2012150, e-mail: siamak.azodolmolky@gwdg.de
(Invited)

ABSTRACT
Software Defined Networking (SDN) is a concept which provides the network operators and data centres to flexibly manage their networking equipment using software running on external servers. According to the SDN framework, the control and management of the networks, which is usually implemented in software, is decoupled from the data plane. On the other hand cloud computing materializes the vision of utility computing. Tenants can benefit from on-demand provisioning of networking, storage and compute resources according to a pay-per-use business model. In this work we present the networking issues in IaaS and networking and federation challenges that are currently addressed with existing technologies. We also present innovative software-define networking proposals, which are applied to some of the challenges and could be used in future deployments as efficient solutions. Cloud computing networking and the potential contribution of software-defined networking along with some performance evaluation results are presented in this paper.
Keywords: cloud computing networking, infrastructure as a service, software-defined networking, network virtualization, performance evaluation.

1. INTRODUCTION

Cloud computing has emerged as a widely accepted computing paradigm built around core concepts such as elimination of up-front investment, reduction of operational expenses, on-demand computing resources, elastic scaling, and establishing a pay-per-usage business model for information technology and computing services. There are different models of cloud computing that are offered today as services like Software as a Service (SaaS), Platform as a Service (PaaS), Network as a Service (NaaS) and Infrastructure as a Service (IaaS) [1]. In spite of all recent research and developments, cloud-computing technology is still evolving. Several remaining gaps and concerns are being addressed by alliances, industry, and standards bodies. Some of these questions are: What are the potential solutions using the existing technologies for the implementation of virtual networks inside IaaS? What are the challenges behind the virtual networking in clouds? Is there any space for Software Defined Networking (SDN) [2] to address virtual networking challenges? When cloud federation is involved, should the servers in the cloud be on the same L2 network or, should a L3 topology be involved? How would this approach operate across multiple cloud data centers?

SDN is an appealing platform for network virtualization since each tenant's control logic can run on a controller rather than on physical switches. In particular OpenFlow [3] offers a standard interface for caching packet forwarding rules in the flow table of switches, querying traffic statistics, and notifications for topology changes. In this work we present key challenges and issues in IaaS and cloud computing networking, which should be addressed using existing technologies or novel and innovative mechanisms. Virtual networking and extensions of cloud computing facilities along with federation issues are the focus of this work. SDN as a novel and innovative mechanism provides proper solutions for these issues.

2. CHALLENGES AND EXISTING IMPLEMENTATIONS

Existing cloud networking architectures typically follow the "one size fits all" paradigm in meeting the diverse requirements of a cloud. The network topology, forwarding protocols, and security policies are all designed looking at the sum of all requirements preventing the optimal usage and proper management of the network. Cloud tenants should be able to specify bandwidth requirements for applications hosted in the cloud, ensuring similar performance to on-premise deployments. Many tiered applications require some guaranteed bandwidth between server instances to satisfy user transactions within an acceptable time frame and meet predefined SLAs. Enterprises deploy a wide variety of security appliances in their data centres to protect their applications from attacks. These are often employed alongside other appliances that perform load balancing, caching and application acceleration. Traffic isolation and access control to the end-users are among the multiple forwarding policies that should be enforced. These policies directly impact the configuration of each router and switch. Changing requirements, different protocols, different flavours of L2 spanning tree protocols (STP), along with vendor specific protocols, make it extremely challenging to build, operate and inter-connect a cloud network at scale. The network topology of data centres is usually tuned to match a pre-defined traffic requirement. The topology design also depends on how the L2 and/or L3 is utilizing the effective network capacity. Applications should run "out of the box" as much as possible, in particular for IP addresses and for network-dependent

failover mechanisms. Applications may need to be rewritten or reconfigured before deployment in the cloud to address several network related limitations. Network appliances and hypervisors are typically tied to a statically configured physical network, which implicitly creates a location dependency constraint. A typical three layer data centre network includes Top of Rack (ToR) layer connecting the servers in a rack, aggregation layer and core layer, which provides connectivity to/from the Internet edge. This multi-layer architecture imposes significant complexities in defining boundaries of L2 domains, L3 forwarding networks and policies, and layer-specific multi-vendor networking equipment. Connectivity between the data centres to provide the vision of "one cloud" is another challenge. There are situations where an enterprise needs to be able to work with multiple cloud providers due to locality of access, migration, merger of companies working with different cloud providers, etc. Cloud federation has to provide transparent workload orchestration between the clouds on behalf of the enterprise user. Connectivity between clouds includes L2 and/or L3 considerations and tunnelling technologies that need to be agreed upon.

Existing networking protocols and architectures such as STP and Multi-Chassis Link Aggregation (MC-LAG) can limit the scale, latency, throughput, and VM migration of enterprise cloud networks. While existing L3 "fat tree" networks provide a proven approach to address the requirements for a highly virtualized cloud data centre, there are several industry standards that enhance features of a flattened layer 2 network, using Transparent Interconnection of Lots of Links (TRILL), Shortest Path Bridging (SPB) or systems based on SDN concepts and OpenFlow. The key motivation behind TRILL and SPB and SDN-based approach is the relatively flat nature of the data-centre topology and the requirement to forward packets across the shortest path between the endpoints (servers) to reduce latency, rather than a root bridge or priority mechanism normally used in the STP. The IEEE 802.1Qaz, (i.e., Enhanced Transmission Selection) allows low-priority traffic to burst and use the unused bandwidth from the higher-priority traffic queues with higher flexibility [3]. Vendor proprietary protocols are also developed to address the same issues. Juniper Networks produces switches, using a proprietary multipath L2/L3 encapsulation protocol called QFabric, which allows multiple distributed physical devices in the network to share a common control plane and a separate common management plane. Virtual Cluster Switching (VCS) is a multipath L2 encapsulation protocols by Brocade, based on TRILL and Fabric Shortest Path First (FSPF) path selection protocol and a proprietary method to discover neighbouring switches. Cisco's FabricPath, is a multipath L2 encapsulation based on TRILL, which does not include TRILL's next-hop header, and has a different MAC learning technique. They all address the same issues with different features for scalability, latency, oversubscription, and management.

3. SDN-BASED CLOUD NETWORKING

SDN is an emerging network architecture where "network control functionality" is decoupled from "forwarding functionality" and is directly programmable. This migration of control, formerly tightly integrated in individual networking equipment, into accessible computing devices (logically centralized) enables the underlying infrastructure to be "abstracted" for applications and network services. There are general advantages to be realized by enterprises that adopt OpenFlow-enabled SDN as the connectivity foundation for private and/or hybrid cloud connectivity. A logically centralized SDN control plane will provide a comprehensive view (abstract view) of cloud resources and access network availability. This will ensure cloud-federation are directed to adequately resourced data centres, on links providing sufficient bandwidth and service levels. A high level description of key building blocks for an SDN-based cloud federation includes: 1) an OpenFlow enabled cloud backbone edge nodes, which connect to the enterprise and cloud provider data centre, 2) an OpenFlow enabled core nodes which efficiently switch traffic between these edge nodes, 3) an OpenFlow and/or SDN-based controller to configure the flow forwarding tables in the cloud backbone nodes and providing a WAN network virtualization application and finally 4) a Hybrid cloud operation and orchestration software to manage the enterprise and provider data centre federation, inter-cloud workflow, and resource management of compute/storage and inter-data centre network management

SDN-based federation will facilitate multi-vendor networks between enterprise and service provider data centres, helping enterprise customers to choose best-in-class vendors, while avoiding vendor lock-in; pick a proper access technology from a wider variety (e.g. DWDM, PON, etc.); access dynamic bandwidth for ad-hoc, timely inter-data centre workload migration and processing; and eliminate the burden of underutilized, costly high-capacity fixed private leased lines. SDN-enabled bandwidth-on-demand services provide automated and intelligent service provisioning, driven by cloud service orchestration logic and customer requirements.

4. COMPARISON OF VIRTUAL NETWORKING IMPLEMENTATIONS

Cloud networking and server virtualization have to be scalable, on-demand and orchestrated. In an ideal situation the physical network will provide the transport, and the hypervisors will provide the VM service and the virtual networks [5] will be constructed on top of the transport network. The traditional approach is to implement the virtual segments using VLANs, which are limited to 4096 segments (VLANs) and therefore not

really scalable. There are some proposals, which suggest to utilize IEEE 802.1ad (Q-in-Q) to address 4K limitation, but there is no orchestration support for Q-in-Q currently. Amazon EC2 is utilizing IP over IP with a rich control plane to provide virtual segments. VM-aware networking, Edge Virtual Bridging (IBM's EVB, IEEE 802.1Qbg), vCloud Director Networking Infrastructure (vCDNI) (VMware, MAC over MAC) or EVB with PBB/SPB, VXLAN (Cisco), Network Virtualization using Generic Routing Encapsulation (NVGRE) (Microsoft) MAC over IP, and Nicira Network Virtualization Platform (NVP) (MAC over IP with a control plane) are other approaches. All of these proposals can be categorized into three architectural groups: a) dumb virtual switch in the hypervisor plus normal physical switch (e.g., traditional VLAN model), b) dumb virtual switch with intelligent physical switch (e.g., VM-aware networking, EVB), and c) Intelligent virtual switch plus a typical (L2/L3) physical switch (e.g.,vCDNI, VXLAN, NVGRE, NVP,…). The summary of virtual networking implementation is presented in Table 1.

Table 1: Comparison of virtual networking implementation

Technology	Bridging	All hosts flooding	vNet flooding	VLAN 4K limit	VM MAC visible	State kept in network
VLANs	Yes	Yes	Yes	Yes	Yes	Yes
VM-aware networking	Yes	No	Yes	Yes	Yes	Yes
vCDNI	Yes	Yes	Yes	No	No	MAC of hypervisors
VXLAN	No	Only to some hosts	Yes	No	No	Multicast groups
Nicira NVP	No	No	Some	No	No	No

The first constraint of VLANs is 4K limitation of VLANs. Secondly, all the MAC addresses from all the VMs are visible in the physical switches of the network. This can fill up the MAC table of physical switches, especially if the deployed switches are legacy ones. Typical NICs are able to receive unicast frames for a few MAC addresses. If the number of VMs are more than these limit, then the NIC has to be put in promiscuous mode, which engages the CPU to handle flooded packets. This will waste CPU cycles of hypervisor and bandwidth. The VM-aware networking scales a bit better. The whole idea is that the VLAN list on the physical switch to the hypervisor link is dynamically adjusted based on the server need. This can be done with VM-aware TOR switches (Arista, Brocade), or VM-Aware network management server (Juniper, Alcatel-Lucent, NEC), which configures the physical switches dynamically, or VM-FEX from Cisco, or EVB from IBM. This approach reduces flooding to the servers and CPU utilization and using proprietary protocols (e.g., Qfabric) it is possible to decrease the flooding in physical switches. However, MAC addresses are still visible in the physical network, the 4K limitations remain intact and the transport in physical network is L2 based with associated flooding problems. This approach could be used for large virtualized data centres but not for IaaS clouds. The main idea behind vCDNI is that there is a virtual distributed switch which is isolated from the rest of the network and controlled by vCloud director and instead of VLAN, uses a proprietary MAC-in-MAC encapsulation. Therefore the VM MAC addresses are not visible in the physical network. Since there is a longer header in vCDNI protocol, the 4K limitation of VLANs is not intact anymore. Although unicast flooding is not exist in this solution, but multicast flooding indeed exist in this approach. Furthermore it still uses L2 transport. Conceptually, VXLAN is similar to the vCDNI approach, however instead of having a proprietary protocol on top of L2; it runs on top of UDP and IP. Therefore, inside the hypervisor the port groups are available, which are tight to VXLAN framing, which generates UDP packets, going down through IP stack in the hypervisor and reaches the physical IP network. VXLAN segments are virtual layer 2 segments over L3 transport infrastructure with a 24-bit segment ID to alleviate the traditional VLAN limitation. L2 flooding is emulated using IP multicast. The only issue of VXLAN is that it doesn't have a control plane.

Nicira NVP is very similar to VXLAN with a different encapsulation format, which is point-to-point GRE tunnels; however the MAC-to-IP mapping is downloaded to Open vSwitch using a centralized OpenFlow controller. This controller removes the need for any flooding as it was required in VXLAN (using IP multicast). To be precise, this solution utilizes the MAC over IP with a control plane. The virtual switches, which are used in this approach, are OpenFlow enabled, which means that the virtual switches can be controlled by an external OpenFlow controller (e.g., NOX [6]). These Open vSwitches use point-to-point GRE tunnels that unfortunately cannot be provisioned by OpenFlow. These tunnels have to be provisioned using other mechanisms, because OpenFlow has no Tunnel provisioning message. The Open vSwitch Database Management Protocol (OVSDB) [7] is used to construct a full mesh GRE tunnels between the hosts that have VMs from the same tenant. Whenever two hosts have one VM each that belong to the same tenant a GRE tunnel will be established between them. Instead of using dynamic MAC learning and multicast the MAC to IP mapping are downloaded as flow forwarding rules through OpenFlow to the Open vSwitches. This approach scales better than VXLAN, because there is no state to maintain in the physical network. Besides, ARP proxy can be used to stop L2 flooding. This requires an OpenFlow and OVSDB controller to work in parallel to automatically provision GRE tunnels.

The performance of software tunnelling in terms of throughput and CPU overhead for tunnelling (i.e., CPU utilization) within Open vSwitch is evaluated for a comparative study. Traffic was generated using 'netperf' to emulate a high-bandwidth TCP flow. The Maximum Transmission Unit (MTU) for the VM and the physical NICs are 1500 bytes and the packet payload size is 32k. The results compare no tunnelling (OVS bridge) case and OVS-STT software tunnelling. Furthermore the results show aggregate bidirectional throughput, meaning that 20 Gbps is a 10G NIC sending and receiving at line rate. All tests where done using Ubuntu 12.10 and KVM on an Intel Xeon 2.40GHz servers interconnected with a 10Gbps Ethernet switch. Standard 10Gbps Ethernet network Interface cards (NICs) were used for this experiment. CPU utilization figures reflect the percentage of a single core, which was used for each of the monitored processes. The following results (Table 2) show the performance of a single flow between two VMs on different hypervisors. We include the Linux bridge to compare the performances with a baseline case. The CPU utilization only includes the CPU, which was dedicated to packet switching in the hypervisor not the overhead in the guest operating system.

Table 2: Performance evaluation results

Approach	Throughput (Gbps)	CPU Utilization (RX side)	CPU Utilization (TX side)
Linux bridge	9.28	86%	76%
OVS bridge	9.36	83%	71%
OVS-STT	9.49	69%	70%

These results indicate that the overhead of software for tunnelling is negligible. Tunnelling in software requires copying the tunnel bits onto the header, an extra lookup (at least on receive side), and the transmission delay of those extra bits when placing the packet on the wire. When compared to all of the other work that needs to be done during the domain crossing between the guest operating system and the hypervisor, the overhead is negligible. Therefore, tunnels add very little overhead to virtualized networks. The place to innovate is at the upper layers with the software controllers that ensure network consistency.

5. CONCLUSIONS

Some of the challenges in the existing Cloud Networks are: guaranteed performance of applications when applications are moved from on-premises to the cloud facility, flexible deployment of appliances (e.g., intrusion detection systems, or firewalls), and associated complexities to the policy enforcement and topology dependence. SDN provides a new, dynamic network architecture that transforms traditional network backbones into rich service-delivery platforms. By decoupling the network control and data planes, SDN-based architecture abstracts the underlying infrastructure from the applications that utilize it. This makes the networking infrastructure programmable and manageable at scale. SDN adoption can improve network manageability, scalability and dynamism in enterprise data centre. SDN-enabled core and edge nodes with a proper SDN controller and network application can be considered as a novel cloud federation mechanism. VLAN, VM-aware networking, vCDNI, VXLAN and Nicira NVP are technologies to provide virtual networks in cloud infrastructures. Nicira NVP, which utilizes MAC in IP encapsulation and external OpenFlow control plane, provides the efficient solution for virtual network implementation.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," September 2011: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf, accessed 30 November 2012.

[2] T. Koponen , M. Casado , N. Gude , J. Stribling , L. Poutievski , M. Zhu , R. Ramanathan , Y. Iwata , H. Inoue , T. Hama , S. Shenker, "Onix: A Distributed Control Platform for Large-scale Production Networks", in Proc. OSDI, 2010.

[3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review, 38(2):69-74, 2008.

[4] C. J. Sher Decusatis, A. Carranza, C. M. Decusatis, "Communication within clouds: open standards and proprietary protocols for data centre networking," Communications Magazine, IEEE , vol.50, no.9, pp.26-33, September 2012.

[5] Bari, M.F.; Boutaba, R.; Esteves, R.; Granville, L.Z.; Podlesny, M.; Rabbani, M.G.; Qi Zhang; Zhani, M.F., "Data Center Network Virtualization: A Survey," IEEE Comm. Surveys & Tutorials, vol.15, no.2, pp.909-928, 2013.

[6] N. Gude, T. Koponen, J. Pettit, B. Pfaff, M. Casado, N. McKeown, and S. Shenker. Nox: towards a operating system for networks. ACM SIGCOMM Computer Comm. Review, 38(3):105-110, 2008.

[7] B. Pfaff, B. Davie, "The Open vSwitch Database Management Protocol," Internet-draft, draft-pfaff-ovsdb-proto-00, Nicira Inc., 20 August 2012.