Cloud Computing Networking based on SDN

Ajay Sonkesriya

Lecturer
Kalaniketan Polytechnic College Jabalpur MP

Abstract - SDN is a technology that enables network operators and data centres to operate their networking hardware in a flexible manner using software that runs on external servers. The SDN framework separates the data plane from network control and administration, which are often handled in software. On the other hand, utility computing is materialized via cloud computing. On-demand provisioning of networking, storage, and computing resources in accordance with a pay-per-use business model is advantageous to tenants. We discuss networking concerns in IaaS as well as networking and federation problems that are currently being solved with current technology in this study. Additionally, we offer some ground-breaking ideas for software-defined networking that are applied to specific problems and may prove to be effective fixes in the future. This paper presents some performance evaluation findings, the potential contribution of software-defined networking, and cloud computing networking.

Keywords: cloud computing networking, infrastructure as a service, software-defined networking, network virtualization, performance evaluation.

I. INTRODUCTION

A new computing paradigm known as "cloud computing" has emerged, and it is based on fundamental ideas like eliminating initial outlay, cutting operational costs, providing on-demand computing resources, allowing for elastic scaling, and establishing a pay-per-use business model for computing and information technology. Different cloud computing paradigms, such as Software as a Service (SaaS), Platform as a Service (PaaS), Network as a Service (NaaS), and Infrastructure as a Service (IaaS), are currently available as services [1]. Cloud computing technology is still developing despite all of the recent research and advancements. Alliances, industry, and standards bodies are addressing a few unresolved issues and gaps. Several of these queries include: What are the potential options for implementing virtual networks within IaaS utilising the technologies that are already available? What difficulties lie behind cloud virtual networking? Is there room for Software Defined Networking (SDN) [2] to overcome the issues with virtual networking? Should the cloud servers involved in cloud federation be on the same L2 network or should an L3 topology be used? How would this method function if used with different cloud data centres?

Since each tenant's control logic can execute on a controller rather than on physical switches, SDN is a desirable platform for network virtualization. In example, OpenFlow [3] provides a common interface for querying traffic data, receiving notifications of topology changes, and caching packet forwarding rules in switches' flow tables. Using existing technologies or new and creative techniques, we should be able to address the major difficulties and problems in IaaS and cloud computing networking that we have identified in this work. This study focuses on virtual networking, cloud computing extensions, and federation-related challenges. SDN offers suitable answers for these problems as a fresh and inventive method.

II. 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 systems. Before being deployed in the cloud, applications may need to be updated or altered to accommodate various network-related restrictions. Typically, network appliances and hypervisors

are bound to a physically static network, which inherently imposes a location dependency constraint. The Top of Rack (ToR) layer, which connects the servers in a rack, the aggregation layer, and the core layer, which provides connectivity to and from the Internet edge make up a typical three-layer data centre network. The definition of L2 domain boundaries, L3 forwarding networks and rules, and layer-specific multi-vendor networking equipment are all made significantly more difficult by this multi-layer design. Another difficulty is achieving the goal of "one cloud" through connectivity between data centres. Due to access limitations, migration, the merger of businesses using several cloud providers, etc., an organisation may need to be able to interact with many cloud providers. For the benefit of the enterprise user, cloud federation must offer transparent workload orchestration between the clouds. L2 and/or L3 considerations and tunnelling technologies that must be agreed upon are part of cloud connectivity.

Enterprise cloud networks' ability to grow, reduce latency, increase throughput, and migrate virtual machines may be constrained by current networking protocols and topologies like STP and Multi-Chassis Link Aggregation (MC- LAG). Although there are several industry standards that improve the 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, existing L3 "fat tree" networks offer a tried-and-true method to address the requirements for a highly virtualized cloud data centre. The main driving force behind the TRILL, SPB, and SDN-based approach is the relatively flat nature of the data centre topology and the need to forward packets across the shortest path between the endpoints (servers), rather than a root bridge or priority mechanism typically used in the STP, to reduce latency. Low-priority traffic can burst and use the unused bandwidth from the queues for higher-priority traffic with greater flexibility according to the IEEE 802.1Qaz standard (also known as enhanced transmission selection) [3]. To overcome the same problems, vendor-proprietary protocols are also developed. Switches made by Juniper Networks utilise the QFabric multipath L2/L3 encapsulation technology, which enables a network's scattered physical devices to share a single control plane and a separate common administration plane. A multipath L2 encapsulation protocol from Brocade called Virtual Cluster Switching (VCS) is based on the Fabric Shortest Path First (FSPF) path selection protocol, TRILL, and a secret technique to find nearby switches. The TRILL-based multipath L2 encapsulation FabricPath from Cisco uses a different MAC learning method and does not include TRILL's next-hop header. With various features for scalability, latency, oversubscription, and management, they all tackle the same problems.

III. CLOUD NETWORKING BASED ON SDN

In an upcoming network architecture known as SDN, "network control functionality" is directly programmable and separated from "forwarding functionality." The underlying infrastructure can be "abstracted" for applications and network services thanks to the transfer of control that was previously firmly integrated into each piece of networking hardware into a single location that is easily accessible computing devices. Enterprises that utilise OpenFlow-enabled SDN as the connectivity foundation for private and/or hybrid cloud connectivity can take advantage of a number of broad benefits. A comprehensive view (abstract view) of cloud resources and access network availability will be offered by a logically centralised SDN control plane. This will guarantee that cloud federation is routed to data centres with suitable resources, over lines with adequate bandwidth, and at service levels. Key components of an SDN-based cloud federation include 1) OpenFlow-enabled cloud backbone edge nodes that connect to the data centres of businesses and cloud providers, 2) OpenFlow-enabled core nodes that effectively switch traffic between these edge nodes, and 3) OpenFlow-enabled cloud backbone edge nodes. 3) a WAN network virtualization programme, an OpenFlow and/or SDN-based controller to set up the flow forwarding tables in the cloud backbone nodes, and finally 4) Hybrid cloud operation and orchestration software for provider and enterprise data centre federation, inter-cloud workflow, resource management for compute and storage, and inter-data centre network management.

SDN-based federation will enable multi-vendor networks between enterprise and service provider data centres, assisting enterprise customers in selecting best-in-class vendors while avoiding vendor lock-in; selecting the right access technology from a wider variety (for example, DWDM, PON, etc.); accessing dynamic bandwidth for on-demand, timely workload migration and processing between data centres; and relieving the burden of under utilized, expensive high-capacity fixed private leased lines. Services for bandwidth-on-demand with SDN support offer automated and intelligent service provisioning that is guided by client needs and cloud service orchestration logic.

IV. COMPARISON OF IMPLEMENTATIONS OF VIRTUAL NETWORKING

Scalable, on-demand, and orchestrated cloud networking and server virtualization are required. In a perfect world, the physical network would act as the transport, hypervisors would handle the virtual machine service, and virtual networks [5] would be built on top of the transport network. Since VLANs are only capable of supporting 4096 segments, the traditional method of implementing virtual segments

incredibly scalable There are various proposals that suggest using IEEE 802.1ad (Q-in-Q) to overcome the 4K constraint, although Q-in-Q currently lacks orchestration support. Virtual segments are offered by Amazon EC2 by utilising IP over IP and a rich control plane. Other methods include VM-aware networking, Edge Virtual Bridging (IBM's EVB, IEEE 802.1Qbg), vCloud Director Networking Infrastructure (vCDNI), MAC over MAC, or EVB with PBB/SPB, VXLAN (Cisco), Network Virtualization using Generic Routing Encapsulation (NVGRE), MAC over IP (Microsoft), and Nicira Network Virtualization Platform (NVP), MAC over IP with a control plane. All of these proposals can be divided into three architectural categories: a) dumb virtual switches integrated into the hypervisor plus conventional physical switches (such as the traditional VLAN model); b) dumb virtual switches combined with intelligent physical switches (such as VM-aware networking, EVB); and c) intelligent virtual switches combined with typical (L2/L3) physical switches (such as vCDNI, VXLAN, NVGRE, NVP, etc.). Table 1 provides a summary of virtual networking implementation.

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 initial VLAN restriction is a 4K VLAN limit. Second, the physical switches in the network can see every MAC address from every VM. This may cause physical switches' MAC tables to become full, especially if the deployed switches are older models. Unicast frames can be received by typical NICs for a small number of MAC addresses. If there are more virtual machines (VMs) than allowed, the NIC must be set to promiscuous mode, which uses the CPU to handle inundated traffic. Hypervisor CPU cycles and bandwidth will be wasted in this way. Scalability is slightly improved by VM-aware networking. The entire concept is to dynamically modify the VLAN list on the physical switch to the hypervisor link based on the needs of the servers. This can be accomplished via VM-aware TOR switches (Arista, Brocade), VM-Aware network management servers (Juniper, Alcatel-Lucent, NEC), VM-FEX from Cisco, or EVB from IBM, which configures the physical switches dynamically. This method lowers flooding to the servers and CPU usage, and it is also possible to lower flooding in physical switches by employing proprietary protocols (such as Qfabric). However, MAC addresses are still accessible on the physical network, there are still 4K limitations, and the physical network's transport is L2 based, which has flooding issues. Large virtualized data centres could employ this method, but IaaS clouds couldn't. The primary concept behind vCDNI is that a virtual distributed switch that employs a proprietary MAC-in-MAC encapsulation instead of VLAN and is isolated from the rest of the network and managed by vCloud director. As a result, the physical network cannot see the VM MAC addresses. The 4K VLAN constraint is no longer in effect because the vCDNI protocol has a larger header. Multicast flooding does exist in this strategy even though unicast flooding does not in this solution. Furthermore, L2 transport is still employed. Conceptually, VXLAN is comparable to the vCDNI concept; however, instead of running on top of L2 with a proprietary protocol, it does so with UDP and IP. As a result, there are port groups inside the hypervisor that are close to VXLAN framing, generating UDP packets that travel through the IP stack in the hypervisor and out to the physical IP network. To overcome the typical VLAN restriction, VXLAN segments are virtual layer 2 segments over L3 transport infrastructure with a 24-bit segment ID. IP multicast simulates L2 flooding. VXLAN's lack of a control plane is its lone drawback.

With point-to-point GRE tunnels as an alternate encapsulation format, Nicira NVP is quite similar to VXLAN. However, Open vSwitch receives the MAC-to-IP mapping through a centralised OpenFlow controller. As opposed to VXLAN, this controller does not require any flooding (using IP multicast). This method specifically makes use of MAC over IP with a control plane. The virtual switches that are used in this method are OpenFlow enabled, which implies that an external OpenFlow controller can control the virtual switches (e.g., NOX [6]). Unfortunately, OpenFlow cannot provision the point-to-point GRE tunnels used by these Open vSwitches. Since OpenFlow lacks a Tunnel provisioning message, these tunnels must be provisioned via alternative methods. Between servers that have VMs from the same tenant, full mesh GRE tunnels are built using the Open vSwitch Database Management Protocol (OVSDB) [7]. A GRE tunnel will be created whenever two hosts each have one VM that is part of the same tenant. The MAC to IP mapping is downloaded as flow forwarding rules over OpenFlow to the Open vSwitches instead of employing dynamic MAC learning and multicast. Because there is no physical network state to maintain, this method scales better than VXLAN. ARP proxy can also be used to prevent L2 flooding. For autonomous GRE tunnel provisioning, an OpenFlow and OVSDB controller must operate in tandem.

For a comparative analysis, Open vSwitch's software tunnelling performance is assessed in terms of throughput and CPU overhead for tunnelling (i.e., CPU utilisation). 'Netperf' was used to create traffic in order to simulate a

high-bandwidth TCP flow. The VM and physical NICs' Maximum Transmission Units (MTU) are 1500 bytes, and the packet payload size is 32k. The findings contrast software tunnelling with no tunnelling (OVS bridge) example. Additionally, the outcomes demonstrate aggregate bidirectional throughput, which translates to 20 Gbps as a 10G NIC delivering and receiving data at line rate. All testing were carried out on Intel Xeon 2.40GHz servers connected by a 10Gbps Ethernet switch using Ubuntu 12.10 and KVM. For this experiment, common 10Gbps Ethernet network Interface cards (NICs) were employed. The percentage of a single core that was utilised by each of the observed processes is shown in CPU utilisation numbers. The performance of a single flow between two VMs running on different hypervisors is displayed in the data below (Table 2). To compare the performances with a reference case, we add the Linux bridge. The hypervisor's dedicated CPU for packet switching is the sole CPU included in the CPU utilisation; the guest operating system's overhead is not included.

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 findings suggest that the overhead of tunnelling software is minimal. In order to tunnel in software, the tunnel bits must be copied onto the packet header, there must be an additional search (at least on the receive side), and the extra bits must be transmitted with a transmission delay. The overhead is insignificant in comparison to all the other tasks that must be completed during the domain switching between the guest operating system and the hypervisor. Tunnels therefore barely increase network overhead in virtualized environments. The top levels, where the software controllers that guarantee network consistency are located, are the best places to innovate.

V. CONCLUSIONS

Guaranteed application performance when moving applications from on-premises to the cloud facility, flexible appliance deployment (such as intrusion detection systems or firewalls), and associated complexity with policy enforcement and topology dependence are some of the challenges in the current cloud networks. SDN offers a fresh, dynamic network architecture that upgrades basic service-delivery platforms from outdated network backbones. SDN-based design isolates the underlying infrastructure from the applications that use it by decoupling the network control and data planes. As a result, the networking infrastructure is scalable and configurable. Adopting SDN can boost the company data center's network manageability, scalability, and dynamism. A unique cloud federation approach can be thought of as SDN-enabled core and edge nodes with an appropriate SDN controller and network application. Virtual networks are provided by technologies like VLAN, VM-aware networking, vCDNI, VXLAN, and Nicira NVP in cloud infrastructures. The effective method for implementing virtual networks is provided by Nicira NVP, which makes use of external OpenFlow control plane and MAC in IP encapsulation.

REFERENCES

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

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

[3] N. McKeown, T. Anderson, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner are three of the authors Campus network innovation is made possible via OpenFlow. Computer Communication Review, 38(2):69-74, ACM SIGCOMM, 2008.

[4] C. J. Sher Decusatis, A. Carranza, and 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, and Zhani, M.F., "Data Center Network Virtualization: A Survey," IEEE Communications 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 are six of the authors. Nox: working on a network operating system. Computer Communications Review, 38(3):105-110, ACM SIGCOMM, 2008.

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