SDN-based Overlay Networks for QoS-aware Routing

Pablo Belzarena*, Gabriel Gomez Sena*, Isabel Amigo+, Sandrine Vaton+

*Facultad de Ingeniería, Universidad de la República, Uruguay
+Telecom Bretagne, Institut-Mines Telecom, France

Abstract

We propose an SDN-based architecture for distant points interconnection through an overlay network. The main goal of the overlay network is to provide with resilient and high-performance interconnection between its nodes, without the need of changing nor handling Internet routers. Our proposed architecture benefits from the advantages inherited from SDN such as simplicity of management and flow-level Traffic Engineering capabilities. In particular, our approach allows to build the overlay network without any tunnelling technology, which promises to provide a gain in terms of performance, and to ease deployment and management. In addition, we address one of the challenges of SDN, by discussing a possible approach for active monitoring in the proposed SDN-overlay architecture.

1 Introduction

Traffic in the Internet is well known to follow non optimal paths. BGP, the *de facto* standard protocol for interdomain routing, has succeeded in providing scalability, allowing the propagation of interdomain routes of more than 50000 Autonomous Systems (AS) [2]. However, quality of service (QoS) parameters are not taken into account by BGP. As a consequence, Internet flows are prone to quality degradation, while routes with better performance with respect to those followed by Internet traffic exist. At the same time, due to ossification of the Internet, solutions involving changes on IP routing are not likely to be adopted.

Overlay networks have emerged several years ago as a way of controlling Internet flows, traffic, or content, without the need of having access to the Internet Service Provider's (ISP) equipment. In particular, they make it possible to overcome connectivity disruptions due to BGP outages, and lack of quality of service, by dynamically optimizing routes in the overlay. Solutions such as [11], [5], [13] have shown to be efficient in terms of resiliency and quality of service up to a moderate number of nodes. Moreover, nowadays several applications relay on overlay networks. Some of them they do it for quality concerns, such

as content delivery networks, others for privacy issues like the onion routing network [3], and some others to perform a common task such as peer-to-peer networks.

With the advent and proliferation of cloud applications and services, datacentres emerged all over the globe, resulting on hundreds of thousands of applications of different tenants scattered all over the world. Interconnection between these datacentres, and between tenants' virtual servers at different datacentres, require high resiliency and application-dependent QoS. In this paper we look at this use case, of several virtual servers running at distant locations and needing a resilient and high-performance communication between them, and without the collaboration of the ISP.

Our approach for scalable, QoS-aware overlay routing is based on Software-defined networking (SDN) principles. The SDN [8] approach decouples the control plane from the data plane and places the control plane in a centralised location. A standardised communication between control and data plane, along with a programmable forwarding equipment (SDN switch), adds flexibility to the network and avoids vendor lock-in issues. In addition, SDN switches become dedicated, highly-efficient forwarding elements. The control plane becomes centralised, or logically centralised, for scalability purposes. This centralised view of the network allows to perform routing decisions in an optimised way.

Classical solutions for traffic engineering (TE) rely on virtual circuits concepts like ATM or MPLS. SDN technology opens a new opportunity for implementing TE over IP networks. The centralised controller not only allows to apply more sophisticated, per flow, TE rules, but also simplifies the provisioning of virtual paths [4].

Our solution, besides the already mentioned advantages inherited from the SDN approach, allows to build an overlay network without the need of tunnelling, promising gains in terms of performance, deployment and management. The non-tunnelling overlay is achieved thanks to rewriting IP headers at each overlay node (ON), and thanks to a centralised brain which configures the ONs to perform the correct packet headers rewritings. In addition, we also discuss active and passive monitoring solutions, and in particular an active monitoring framework, which is, to the best of our knowledge, an aspect which has not yet received enough attention in the context of SDN.

2 The Proposed Architecture

Overlay solutions for managing Internet flows without the collaboration of the Internet service provider have been proposed before by different previous works. Most of them, like [5, 9] seek for resiliency and QoS. They propose a distributed approach and rely on IP tunneling in order to ensure routing on the overlay. In [14], a centralised approach for connectivity resiliency is proposed, while routes in the overlay also rely on IP in IP tunnelling. Recently, in [7], an approach for cloud resiliency based on an SDN overlay is presented. Their proposed architecture shares many principle with ours, though in their case the main

objective is resiliency and less emphasis is put on QoS-aware routing, and QoS monitoring. In addition, their focus is not put on implementation details, aspect on which we provide insights.

Our proposed architecture, shown in Fig. 1, relies on an ON at each point of presence, and on a centralised controller placed somewhere in the cloud. The ONs are equipped with SDN switches, and with a probe packet generator box (PPG), which we shall explain in Sec. 4. The centralised controller is an SDN controller which interacts through the southbound API with the ONs using OpenFlow protocol, and with a Traffic measurements application (MonApp) and a TE application (TEApp), through a Northbound API.

3 Non-tunneling Overlay

The TEApp has knowledge of all performance metrics through the MonApp, as we shall explain in Sec. 4. Upon QoS monitoring results, it decides which flows need to be rerouted, and the explicit paths in the overlay those flows should follow. This information is sent to the controller through the Northbound API, who in turn sets the needed forwarding rules on the concerned ONs, providing a new virtual path for the desired flows.

The classic approach to get a flow follow a given path on the overlay network is to tunnel the flow through the desired path, but tunnel provisioning and management is a complex task. Moreover, encapsulation can lead to performance degradation due to IP fragmentation, since the host originating a packet is not aware of the outer headers that are going to be applied by the network, resulting in packets that exceed the MTU size.

Our proposal for routing flows through the overlay without encapsulation, and without the collaboration of the ISPs, uses standard OpenFlow features. The controller uses OpenFlow $OFPAT\_SET\_NW\_DST$ action type to rewrite the destination network address of outgoing packets at each ON in the path of a given flow, as shown in Fig. 1. Using a dedicated range of IPv6 addresses for each ON we manage to distinguish the flows at each ON. The controller sets matching rules at ONs' forwarding tables accordingly, signaling a path from origin to destination.

In the case of an IPv4 scenario, using a dedicated range of addresses for distinguishing flows is not a feasible solution. In this case, flows are identified by origin IPv4 address along with the origin port. Matching rules at every ON, which consider these parameters are configured by the controller.

All in all, the adoption of the SDN-based overlay solution provides with many benefits. TE rules can be applied at a flow level basis, signalling overhead required by classical tunneling solutions to agree on site to site identifiers is avoided, and the solution is independent of ISPs' collaboration. Although our proposed method for non-tunnelling overlay can be seen as a Network Address Translator (NAT)-like solution, it is much more transparent for end users since packets arrive to the final destination with their original addresses.

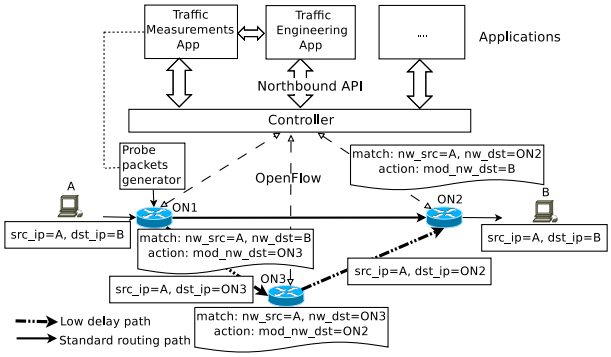

Figure 1: Overlay architecture

4 Monitoring Framework

Traffic measurements and monitoring are critical tasks for TE. Hence, measuring critical QoS parameters (as throughput, delay, and packet loss) has been a widely researched topic for several years. Recently, in the context of SDN networks, this problem has also been subject of research.

Flows' throughput measurements in SDN networks has been addressed by several works and tools. The main problem is that SDN architectures use existing flow-based monitoring tools from traditional IP networks. Thus, a trade-off between the measurement accuracy and the overload of network resources exists, in this case, given by the switches' resources and the overhead of OpenFlow control traffic between the controller and the switch. In the last years, different authors have proposed many solutions to accurately measure throughput without network overloading. Some works estimate a flow's throughput using only two OpenFlow control messages: `PacketIn`, that indicates the first packet of a flow to the controller and `FlowRemoved` indicating the end of a flow. The controller using the first message defines the start of a flow and with the second one obtains the size of the flow and its duration. The main issue with this method is the measurement accuracy, that it is not particularly suitable to observe throughput variations in short time-scales. Other proposals use other messages to have a better accuracy in short time-scales. These messages are: `FlowStatisticRequest` sent from the controller to the switch, and `FlowStatisticReplay` sent in the opposite direction. The controller using these two additional messages, can estimate the throughput between two statistics requests. The issue is that sending these messages all the time for every flow and switch can overload the network. This issue has been addressed in many previous works. Payless [6], for example, proposes a monitoring framework and addresses the network overload problem adapting the time between two consecutive messages (`FlowStatisticRequest`) according to the throughput variations seen in the previous messages.

Packet loss can also be measured using messages provided by OpenFlow. For

example, OpenNetMon [10] uses the existing OpenFlow `FlowStatisticsRequest` message in order to measure packet loss of one flow between two switches in a path. In OpenNetMon, the controller requests both switches the statistics and subtracting the increase in the flow switch packet counter between the source switch and the destination switch, it estimates the packet loss.

Path delay between two switches is more difficult to measure. OpenFlow switches do not timestamp packets, therefore, passive measurement of path delay in OpenFlow is unfeasible. In the last years, two interesting works [10, 12] focus on the problem of active measurements using OpenFlow messages. OpenFlow has a `PacketOut message`, which is sent by the controller to the switch and allows injecting a packet into the network. Both proposals use a `PacketOut` message, which carries a timestamped raw packet to be injected into the switch. The controller also programs on all switches belonging to the path the forwarding rules for that packet, and in the last switch a rule is set to get the packet back to the controller. With this information the controller estimates the path delay that includes also the delays of the `PacketIn` message of the first and last switch to the controller. In order to overcome the inaccuracy of the measure, in [12] the delay from the switches to the controller is estimated and is subtracted to the probe packet delay.

The previously described approach has two drawbacks. First, the inaccuracy introduced by the delay between the controller and the switches. In a datacenter network like in [12], the delay between switches and controller can be more controllable than in a SDN Overlay Network over Internet. In this last case, the delay between switches and controller could have strong variations and could be comparable to the probe packets' delay through the selected path. Second, active measurements can send many packets and with, for example, inter-departures times with an specific distribution so as to estimate the QoS parameters seen by applications. Sending the probe packets from the controller to the first switch can increase with unnecessary traffic the path between controller and switches, and change the probe packets' inter-departures times as sent from the first switch.

For the previously exposed reasons, we propose a different approach for active measurements. Our proposal can be seen in Fig. 1. We include in the application layer a Traffic measurement application (MonApp), and in the data plane a probe packet generator box (PPG) to be programmed by the MonApp. When the TEApp needs to measure QoS at any path, it sends a request to the MonApp to perform a specific type of measurement. MonApp defines the characteristics of the probe packets to be sent. In addition, it configures the PPG and starts it. The PPG sends probe packets to the first switch of the path. The framework assumes that at each ON a PPG exists. The MonApp requests to the controller, through the Northbound API, the routes to be configured at each ON. In turn, the controller sets the forwarding rules for the probe packets in each switch belonging to the tested path, through the OpenFlow protocol. The last switch of the path can send the probe packets backward to estimate the RTT or, if PPGs are synchronized, the last switch can send the probe packets to its PPG for a one way delay measurement. By this approach, control paths are

not overloaded with probe packets, and more accurately active measurements can be performed.

A possible approach for the implementation could be extending the functionalities of RIPE Atlas [1] data collection system. RIPE atlas is a worldwide Internet connectivity measurement network which has probe boxes scattered allover the world. These probe boxes can be programmed to perform different customizable Internet measurements through a RESTful API.

5 Conclusions and Future work

In this work we propose an architecture for traffic engineering that takes advantage of the SDN benefits to provide an overlay routing architecture. Our solution, allows to build an overlay network without the need of tunnelling, presenting advantages in terms of performance, deployment and management. In addition, we discuss active and passive monitoring solutions, and we present an active monitoring framework, which is, to the best of our knowledge, an aspect which has not yet received enough attention in the context of SDN. In our future work we will test the architecture with Mininet and analyze scalability and security issues.

Acknowledgements

This work was partially founded by CSIC groups grants, Uruguay and Stic Amsud PROVE project.

References

[1] RIPE Atlas, the RIPE NCC's global, open, distributed Internet measurement platform. https://atlas.ripe.net/.

[2] The CIDR REPORT for 15 Mar 16. http://www.cidr-report.org/.

[3] Tor projet, anonimity online. https://www.torproject.org/.

[4] I. F. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou. A roadmap for traffic engineering in sdn-openflow networks. *Comput. Netw.*, pages 1–30, 2014.

[5] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient Overlay Networks. In *18th ACM SOSP*, 2001.

[6] S. R. Chowdhury, M. F. Bari, R. Ahmed, and R. Boutaba. Payless: A low cost network monitoring framework for software defined networks. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014.

[7] A. Fressancourt and M. Gagnaire. A sdn-based network architecture for cloud resiliency. In *Consumer Communications and Networking Conference (CCNC), 2015 12th Annual IEEE*, pages 479–484, Jan 2015.

[8] OnF White papers. Software-Defined Networking: The New Norm for Networks. [Online] https://www.opennetworking.org, April 2012.

[9] S. Savage, T. E. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. M. Voelker, and J. Zahorjan. Detour: informed Internet routing and transport. *IEEE Micro*, (1), 1999.

[10] N. LM Van Adrichem, C. Doerr, and F. A. Kuipers. Opennetmon: Network monitoring in openflow software-defined networks. In *Network Operations and Management Symposium (NOMS), 2014 IEEE*. IEEE, 2014.

[11] B. De Vleeschauwer, F. De Turck, B. Dhoedt, P. Demeester, M. Wijnants, and W. Lamotte. End-to-end QoE Optimization Through Overlay Network Deployment. In *Information Networking, 2008. ICOIN 2008.*, Jan 2008.

[12] C. Yu, C. Lumezanu, A. Sharma, Q. Xu, G. Jiang, and H. V. Madhyastha. Software-defined latency monitoring in data center networks. In *Passive and Active Measurement*. Springer, 2015.

[13] H. Zhang, L. Tang, and J. Li. Impact of Overlay Routing on End-to-End Delay. In *Computer Communications and Networks, 2006. ICCCN 2006.*, 2006.

[14] X. Zhang and C. Phillips. Network operator independent resilient overlay for mission critical applications (ROMCA). In *Communications and Networking in China, 2009. ChinaCOM 2009.*, 2009.