An SDN-based approach for QoS and Reliability in Overlay Networks

Isabel Amigo*, Gabriel Gómez Sena†, Marwa Chami*, Pablo Belzarena†

*IMT Atlantique Bretagne-Pays de la Loire, Brest, France, † Universidad de la República, Uruguay
Email: {isabel.amigo, marwa.chami}@imt-atlantique.fr, {ggomez,belza}@fing.edu.uy

***Abstract*—We propose an Overlay network architecture for reliable and QoS-aware interconnection between its nodes, without handling Internet routers and without tunneling overhead. The architecture is based on the SDN paradigm. We demonstrate the feasibility and challenges of such a system using mininet and pox controller.**

I. Introduction

Overlay networks have emerged several years ago for different purposes, namely content delivery, privacy, peer-to-peer, on-line gaming, data centers interconnection, among others. Indeed, it is well known that quality-of-service (QoS) is not taken into account by the current Internet *de facto* routing protocol, BGP, and that following non-BGP routes can provide better QoS and overcome connectivity failures due to BGP outages. Overlay networks can thus bypass these issues, without the need of controlling intermediate nodes nor changing an ossified protocol stack. On the other hand, previous work either doesn't provide overlay scalable solutions for QoS-aware overlay networks (see e.g. [1]) or rely on decentralized solutions adding overhead (such as relaying on encapsulation techniques).

To fill this gap, and thanks to the new opportunities the SDN paradigm offers, we have proposed, in [2], an SDN-based architecture to allow QoS-aware and reliable routing in overlay networks. This solution leverages three key features of SDN: 1) a virtually centralized view of the architecture, in order to efficiently perform smart monitoring and routing decisions, 2) a fine-grained, flow-based forwarding capability, 3) a decoupled data and control plane interfaced through a standardized protocol (OpenFlow).

In this paper, we focus on the control of the overlay for achieving a scalable end-to-end QoS aware routing. For this sake, we propose to use the functionalities provided by SDN and in particular the southbound interface OpenFlow to modify, at each controlled switch, packet headers as convenient, for having packets to follow the adequate path, in a per flow basis. This paper demonstrates the feasibility of such an approach under a flexible testbed based on mininet[1], OVS switches[2], and the pox controller[3].

[1]An Instant Virtual Network on your Laptop www.mininet.org
[2]Open Virtual Switch, www.openvswitch.org
[3]https://github.com/noxrepo/pox

II. System Architecture

Fig. 1 depicts our proposed architecture, which is composed of three main blocks: a traffic engineering one, a monitoring one, and the controller itself. This virtually centralized infrastructure controls an overlay network which is composed of different sites (Overlay Nodes, ON). ONs are OpenFlow-capable switches and are interconnected among them through the Internet, constituting the overlay network. We briefly describe the main key functionalities of the system, while reader is referred to our previous work [2] for more details.

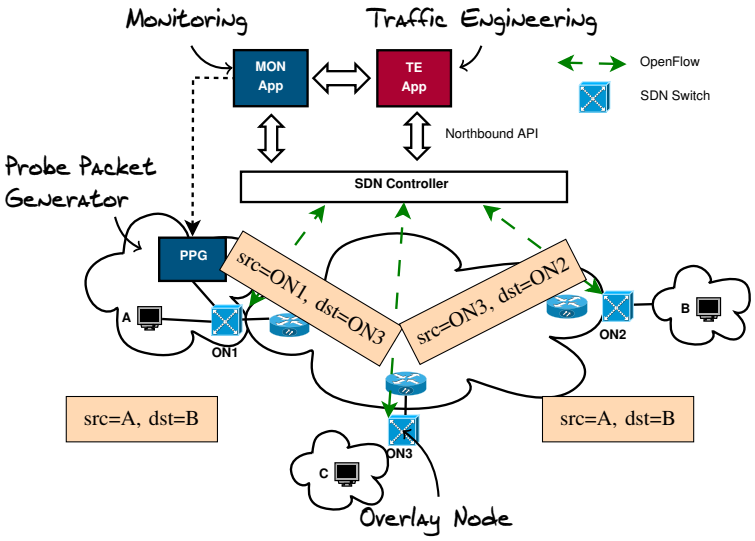

Fig. 1: System description

A. Monitoring and Traffic Engineering

TEApp computes the best paths according to some QoS metrics (e.g delay). MonApp assists TEApp by providing feedback information about network state. Solutions existing in the literature for performance monitoring in SDN, in particular for delay measurement, present a trade-off between accuracy and overloading the network with traffic between controller and switches (see e.g. [3]). They are not convenient to our scenario, where the controller can be far from switches. Our proposed solution is based on equipping each overlay site with a probe packet generator (PPG, for convenience we only show one in Fig. 1) which is commanded by the MonApp. Active monitoring is thus possible thanks to locally-generated probing packets, following instructions from a centralized intelligence.

B. Controlling the Overlay Network

Upon arrival of a new packet to the controller, it decides on which path the packet –and subsequent packets of the same flow– must follow. For such decision, it queries the TEApp module. Upon response, the controller programs switches along that path. In particular, it will choose a flow identifier, and program each switch in the path to match such packets, and to perform a set of actions. Flow identification varies in IPv4 and IPv6 environments (transport ports are used in the former while dedicated IP addresses are used in the latter). Actions are, for instance, modification of destination and source network addresses adequately and flow identifier. Finally, forwarding of packets by switches is performed on a per flow basis, and thanks to the programmed OpenFlow matching rules and actions. Fig 1 illustrates a simple example where a flow from A to B is routed through C.

Source addresses are changed to avoid having packets filtered through reverse path forwarding by intermediate routers or middleboxes. The architecture supposes an in-band control, meaning data and control (and monitoring) traffic use the same physical network. The controller thus programs switches adequately in order to handle these different kinds of traffic.

Our solution reminds an overlay-scale NAT/PAT (Network Address/Port Address Translation), while being more transparent to final users, since it restores all values before reaching the destination. Other state-of-the-art solutions rely on virtual private networks (VPN). Compared to VPN, our approach simplifies management, thanks to the usage of standard OpenFlow actions, and prevents from encapsulation overhead, and thus potential performance degradations due to fragmentation.

III. Implementation and Demonstration

The demonstration focuses on the control and forwarding functions (Subsec. II-B). The objective is to show a working testbed where fine-grained, overlay, QoS-based forwarding is performed leveraging the SDN paradigm with a non-tunneling approach. In previous work [4], we have proposed a joint monitoring and routing solution to suit the needs of TEApp and MonApp. While such a solution is left out of the scope of this demonstration, for the sake of completeness, we consider basic TEApp and MonApp implementations.

Fig. 2 shows the main components of our demonstrator:

a) Overlay network: emulated in mininet. A flexible python script is provided to establish the overlay network. Components of the network include: 1) overlay nodes at the boundary of each overlay site, they are virtual OpenFlow-capable switches (OVS in this case), 2) end hosts belonging to the different overlay sites, 3) IP routers, which represent the Internet, and thus not configurable by the Overlay control and 4) backbone and access links.

b) Controller: based on pox controller. This python-based controller, though simple, is suitable for our case where the objective is to test the feasibility of the solution.

c) TEApp: based on a shortest path module, receiving as input monitoring information and computing the shortest path between two overlay nodes based on the Dijkstra algorithm.

d) MonApp: triggers PPG to measure paths' delays. PPG performs measurement thanks to ping linux command.

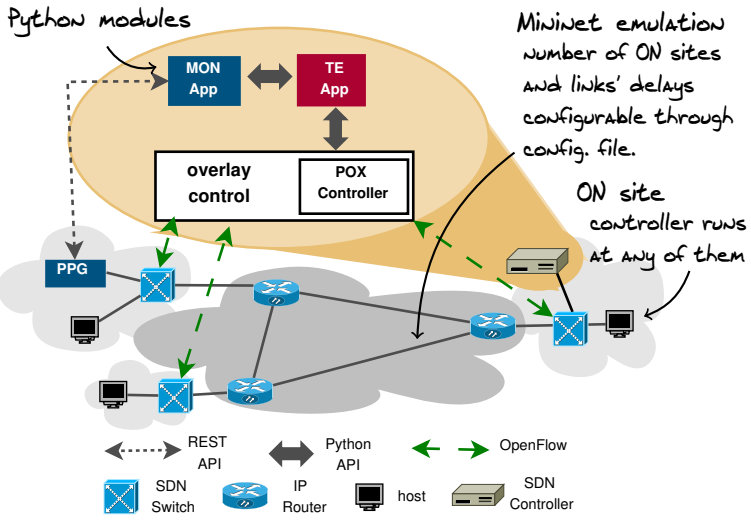

Fig. 2: Demonstration architecture

All software components are publicly available[4] under the GNUv2 license.

To evaluate our implementation, we generate TCP and UDP traffic between the different end hosts, and verify that it follows the path selected by the TEApp. Topology and backbone delays can be easily modified to test several scenarios.

IV. Conclusion

We have demonstrated the viability of implementing a non-tunneling overlay leveraging SDN architecture and OpenFlow messages. This scheme has at least two benefits: it allows to perform an intelligent management and control of the network, thanks to a centralized view; and to handle flow forwarding at a very fine grain, allowing sophisticated QoS aware routing.

In our ongoing work, we are pushing further our testbed along four axes: real distant sites and integration of real measurements, migration to opendaylight controller, and integration of our smart joint routing and monitoring technique.

Akcnowledgements

Brittany region SAD program, France (REREDO project), ANII, Uruguay (FMV_1_2017_1_135526 project) and CSIC groups (Uruguay).

References

[1] D. G. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris, "Resilient Overlay Networks," in *18th ACM SOSP*, (Banff, Canada), October 2001.

[2] P. Belzarena, G. Gomez, I. Amigo, and S. Vaton, "SDN-based Overlay Networks for QoS-aware Routing," in *LANCOMM'16: SIGCOMM workshop*, 2016.

[3] 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.

[4] M. Mouchet, S. Vaton, O. Brun, P. Belzarena, I. Amigo, and B. Prabhu, "Optimisation conjointe de la métrologie active et du routage: une approche markovienne," in *CoRes 2018: Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, 2018.

[4]https://redmine.telecom-bretagne.eu/git/overlay_sdn