Umbrella: a deployable SDN-enabled IXP Switching Fabric

Marc Bruyere[1], Gianni Antichi[3], Eder L. Fernandes[4], Remy Lapeyrade[2],
Ignacio Castro[4], Steve Uhlig[4], Philippe Owezarski[2], Andrew W. Moore[3]

[1]Information Technology Center, University of Tokyo, JP [3]Computer Laboratory, University of Cambridge, UK
[4]Queen Mary University of London, UK [2]CNRS, LAAS, FR

ABSTRACT

Software Defined Internet eXchange Points (SDXs) are a promising solution to the long-standing limitations and problems of interdomain routing. While proposed SDX architectures have improved the scalability of the control plane, these solutions have ignored the underlying fabric upon which they should be deployed. This work makes the case for a new fabric architecture that proposes stronger control and data plane separation.

KEYWORDS

Software Defined Networking, OpenFlow, Internet eXchange Point

1 INTRODUCTION

Internet eXchange Points (IXPs) are a central element of the Internet ecosystem: IXPs can carry huge traffic volumes and interconnect a multitude of networks of different types [1]. The fundamental service provided by IXPs is a Layer 2 neutral facility where heterogeneous networks exchange IP traffic. While IXPs are the ideal vehicle to extend the benefits promised by Software Defined Networking (SDN) to the interdomain level [4], reliability and scalability are essential aspects of an IXP that cannot be compromised by the introduction of SDN. In particular, the impact of control channel disruptions or outages can cause severe disturbances, potentially affecting hundreds of networks and huge traffic volumes [2]. Control plane failures in large scale deployed SDN networks outweigh largely data or management combined together [3].

In this paper we make the case for a stronger control and data plane separation and propose Umbrella, a novel approach to IXP fabric management that can be deployed in any IXP topology to reduce the risks of a fabric excessively dependent on the control plane. Umbrella leverages SDN programmability to tackle part of the control traffic, removing the actual mac learning mechanism in legacy IXPs networks. For this, the broadcasted Address Resolution Protocol (ARP) traffic is directly programmed within the data plane. Secondly, we use a Layer 2 encode path to minimize the resources, management cost and extend scalability. This approach greatly simplifies the management of the fabric: the only role of the controller is `supervising` the network, leveraging its global knowledge.

The main contributions of this paper are:

- We make the case for a stronger control and data plane separation in SDN IXPs.
- We introduce the Umbrella architecture and show how it leverages SDN programmability within the data plane.

2 THE CASE FOR STRONGER CONTROL AND DATA PLANE SEPARATION

Previous works have shown how OpenFlow (OF) [6] could be deployed at the exchange [4] using an IXP fabric with a central controller for all the peering routers at the IXP. In such architecture, the SDN controller would be co-located with the Route Server (RS) to ensure that the SDN and BGP control planes can talk to each other with a minimal delay. Despite advantages such as richer policies, one challenge remains: data plane issues may affect control plane messages, leading to a slow or unresponsive control plane, further aggravating the effect on the data plane. The critical problem resides in the centralized ARP-proxy: delays in the control channel might lead to all the connection oriented mechanisms (i.e., BGP, TCP) failing. For example, if the ARP messages of the peering router A get delayed inside the IXP fabric to reach the SDN controller: all the BGP sessions between the router A and its peers would suffer, forcing (in the worst case scenario) establishing new connections.

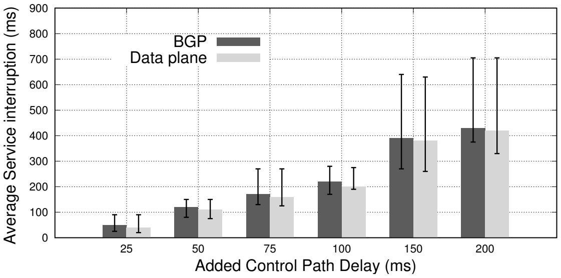

Figure 1: The dependency between control and data plane.

We show in Fig. 1 the disruption caused by ARP delays on the data plane or BGP sessions. We emulated the above SDN scenario of a central controller co-located with the RS on Mininext, a Mininet extension to build complex networks. We instantiated virtual containers acting as peering routers: one for the RS and two more working as client hosts directly connected to one peering router each. We represented the IXP as a single open vSwitch coupled with a Ryu controller acting also as an ARP-proxy, as in [4]. Fig. 1 shows that, even a small delay of a few tens of milliseconds for ARP messages may trigger much larger disruptions on the data plane or BGP. Given the large volumes of traffic IXPs' critical role, such disruptions are not acceptable [7].

3 A NEW SDN FABRIC FOR IXPS

IXPs apply strict rules [5] to limit the side effects of a layer-2 shared broadcast domain, e.g., the MAC address of the router with which the member connects to the peering fabric must be known in advance. Only then the IXP will allocate an ethernet port on the edge switch and configure a MAC filtering Access Control List (ACL) with that MAC address. The location of all the member's routers is thus known to the IXP. For this reason, Umbrella eliminates the need of location discovery mechanisms based on broadcast packets, i.e., ARP request, IPv6 neighbor discovery, making unnecessary the active ARP-proxy daemon proposed in previous SDN-enabled IXP solutions. Umbrella makes on-the-fly translation of broadcast packets into unicast by exploiting the OF ability to rewrite the destination MAC address of a frame matching a rule.

We propose a label-oriented forwarding mechanism to reduce the number of rules inside the core of the IXP fabric. Umbrella edge switches explicitly write the per-hop port destinations into the destination MAC field of the packet. The first byte of the MAC address represents the output port to be used by the core switch.

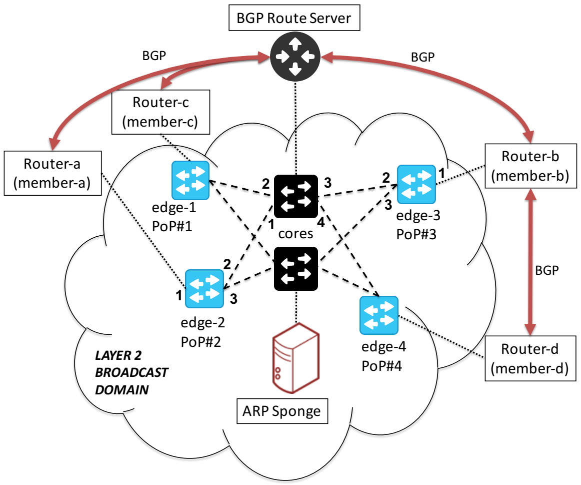

Figure 2: Typical topology of a medium to large IXP.

We now explain how Umbrella using the IXP topology in Fig. 2. The path to connect `router-a` to `router-b` via `core-a` goes through port numbers 2, 3 and 3. When the `router-a` send an ARP request (i.e., broadcast message), the switch `edge-2` receives the frame, rewrites the destination MAC address with the corresponding ports path, `03:01:00:00:00:00`, and forwards it to the core switch. Once the frame reaches the core, it is redirected to output port 3, and then to switch `edge-3` (i.e., the forwarding in the core is based on the most significant byte). Finally, `edge-3`, before forwarding the frame through the output port indicated in the second byte of the MAC address, rewrites that field with the real MAC address of `router-b`.

When the source and destination are directly connected to the same edge switch, no encoding is needed, and the the edge switch directly replaces the broadcast destination address with the target MAC destination address. In an IPv6 scenario, the OF match pattern indicated in the edge switch needs to be on the IPv6 ND target field of the incoming ICMPv6 Neighbor Solicitation packet. The matching table on the edge switch should maintain an association between IPv6 addresses and their location, as in the IPv4 case.

3.1 A label switching approach

Umbrella's forwarding mechanism allows reusing legacy switches in the core, limiting the burden (and costs) of upgrading the hardware. A core switch only needs to forward packets based on simple access filtering rules, whereas the edge switches need OF-like capabilities to rewrite the layer-2 destination field. While this approach is directly applicable to single-core IXP fabrics, it cannot handle multiple-hops fabrics. With a single hop, the core switch would expect the output port to be encoded in the most significant byte of the destination MAC address. In the multi-hop case, since a packet can traverse multiple core switches, a new encoding scheme is needed to distinguish the output ports at different core switches. This is a fairly common case in hypercube-like topologies, such as the ones adopted by LINX or MSK-IX.

Umbrella leverages source routing to allow a correct packet forwarding in such topologies. An ordered list of `output ports` is encoded by the fabric input edge in the destination MAC address as a stack of labels. Each core node then processes the frame according to the value on the top of the stack and pops it before forwarding the frame. With this configuration each switch only needs to look at the most significant byte of the address, regardless of where is it located in the path toward the destination. Popping out from the MAC destination address, the last used label requires header rewriting capabilities, making this solution feasible only for OF-enabled core switches. In particular, every core switch must have 2 action tables: forwarding and copy-field.

4 CONCLUSION

Umbrella is a solution that enhances IXP reliability, manageability and scalability. By handling the control traffic directly within the data plane, our approach reduces failures or disruptions and illustrates the advantage of a sharper separation between control and data plane for IXP management. We see Umbrella as a first step towards SDN architectures less dependent on the control plane, supporting the controller in its role of an intelligent supervisor, rather than as an active and dangerously critical decision point.

REFERENCES

[1] Bernhard Ager, Nikolaos Chatzis, Anja Feldmann, Nadi Sarrar, Steve Uhlig, and Walter Willinger. 2012. Anatomy of a Large European IXP. In *SIGCOMM*. ACM.

[2] Vasileios Giotsas, Christoph Dietzel, Georgios Smaragdakis, Anja Feldmann, Arthur Berger, and Emile Aben. 2017. Detecting Peering Infrastructure Outages in the Wild. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. ACM, 446–459.

[3] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. 2016. Evolve or Die: High-Availability Design Principles Drawn from Googles Network Infrastructure. In *SIGCOMM*. ACM.

[4] A. Gupta, L. Vanbever, M. Hahbaz, S.P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett. 2014. SDX: A Software Defined Internet Exchange. In *SIGCOMM*. ACM.

[5] M. Hughes, M. Pels, and H. Michl. 2015. Internet Exchange Point Wishlist. https://www.euro-ix.net/ixps/ixp-wishlist/. (2015). [Online; accessed Feb. 2018].

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling Innovation in Campus Networks. *CCR* 38, 2 (2008).

[7] Hung D. Vu and Jason But. 2015. How RTT Between the Control and Data Plane on a SDN Network Impacts on the Perceived Performance. In *ITNAC*. IEEE.