GENIX: A GENI-based IXP Emulation

Shahzeb Mustafa, Prasun Kanti Dey, and Murat Yuksel
University of Central Florida, Orlando, USA
Shahzeb.Mustafa@knights.ucf.edu, Prasun@knights.ucf.edu, Murat.Yuksel@ucf.edu

***Abstract*—We present *GENIX*; a modular emulation framework that mimics the behavior of IXPs on a public test-bed. *GENIX* is implemented in Global Environment for Network Innovations (GENI) and capable of enumerating various interactions among ISPs, expected in the real world. *GENIX* is the first effort towards alleviating the complexities of large-scale emulation to abstract a lightweight design where IXP dynamics can be usefully applied. We test *GENIX* for its ability to handle large volumes of traffic flow and its performance during network congestion.**

***Index Terms*—Peering, IXP, ISP, GENI**

I. Introduction

The Internet landscape is progressively transitioning towards a flat structure to prune multiple Internet Service Provider (ISP) layers. At the core of this transition are the Internet Exchange Points (IXPs) which play a critical role in mediating traffic exchange among ISPs [1], [2]. IXPs bring multiple ISPs under the same roof and provide the infrastructure and services required for a hassle-free peering connection establishment.

Though peering has existed for decades, its emergence as a key ingredient is relatively new and presents great room for research and advancement. However, the actual state of the IXPs and peering fabric is changing at a very low pace mainly because of two reasons:

a. Privacy concerns: ISPs and IXPs are very cautious when sharing their internal information publicly. Although a few peering datasets are available (e.g., PeeringDB, RIPE, Packet Clearing House), a detailed analysis and aggregation of these datasets does not reveal substantial information. Additionally, validation of the available IXP data is still difficult [3].

b. Lack of confidence: Due to only a partial visibility into the IXP infrastructure, researchers and experimenters are restricted in the extent to which they can test it. Consequently, this interdependence results in a deadlock, and prevents innovative frameworks from industrial deployment.

We present *GENIX*: An emulation that captures the key structural and functional features of IXPs in order to take a closer look at their behavior in the wild. This paper makes the following key contributions:

1) Design and development of a simple yet scalable IXP emulator that can assist the research community and IXP industry,
2) Models of public and private peering within an IXP, and
3) Automation of the complete process for convenient usage and public availability of an experimental version of the developed IXP model.

This work was supported in part by US National Science Foundation award 1814086.

II. GENIX: Design and Development

GENIX employs emulation models for public as well as private peering. We leverage recent advancements in the field of dynamic virtualization technologies and design the entire topology of our model using publicly available tools. Both the models employ eight virtual BGP routers, four virtual switches, and a remote SDN controller. In the case of public peering, a route server (RS) is also part of the topology. We use Ubuntu 16.04.6 LTS (xenial) and 4.4.0-174-generic kernel in all of the virtual machines.

In the private peering model, we abstract the underlying switch topology using a remote SDN controller that can dynamically install traffic flow rules in the switches. Since switches are layer 2 devices, adding a flow rule is analogous to adding a dedicated physical connection between the routers. For the public peering model, we use the virtual route server module available in the Free Range Routing (FRR) routing suite along with a switch topology similar to the private peering model mentioned above. RS collects all the routes advertised by the routers connected to it, aggregates them, and sends them to every one of the connected routers. For traffic exchange between one another, routers use the underlying fat-tree switch topology as shown in Figure 1.

Each node in the IXP model is a virtual machine reserved in GENI [4], a research test-bed used for network experimentation that allows users to generate virtual network topologies. We install FRR version 7.2 [5] to set them up as BGP routers. FRR is a Linux-based routing software suite and includes all standard routing protocols including BGP, RIP, OSPF, and IS-IS. We set up the switch topology using Open vSwitch (OVS) version 2.13.90 [6], an open-source multi-layer software switch suited to work in dynamic virtual environments. We connect all switches to a remote SDN controller running Floodlight version 1.2 [7] to allow simpler route management. We deploy the controller in a separate location and connect the switches remotely so it remains independent of changes in the internal topology for experimentation.

Although GENI provides a User Interface (UI) for designing custom network topologies, we have simplified the IXP emulation process by automating each stage of design and development. Our scripts reserve the required resources and install the necessary tools and dependencies on each node. For each node, BGP configuration files are automatically generated and routing tables are populated. We open-source the scripts to the community for further testing and feedback [1].

[1]https://github.com/shahzebmustafa/GENIX

Authorized licensed use limited to: University of Central Florida. Downloaded on October 09,2020 at 17:24:46 UTC from IEEE Xplore. Restrictions apply.

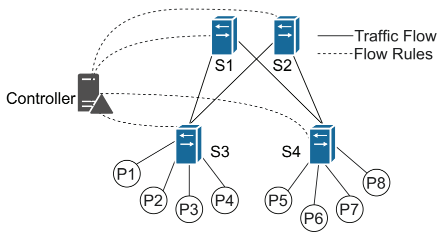

Fig. 1. Fat-tree switch topology connects member ISPs in an IXP.

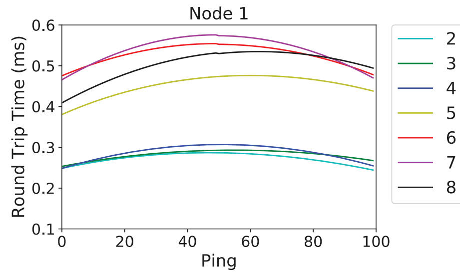

Fig. 2. Latency for ping from Node 1 to every other node for 100 pings.

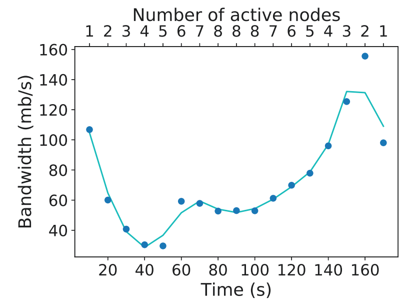

Fig. 3. Node bandwidth with an active node added every 10 seconds.

III. Evaluation and Analysis

We perform thorough tests to verify and evaluate the functionality of the emulator and to demonstrate that the proposed model correctly captures the key characteristics of an IXP. Since the underlying topology for traffic exchange is same for both the models, the average test results are also the same. Therefore, we refer to the results as IXP-specific rather than private or public specific. In order to test the maximum performance of the underlying topology of the emulator, we add traffic flows between all of the nodes P1 through P8.

To analyze the effect of hop count on latency, we use Pings to measure latency between all nodes. As shown in Figure 2, Node 1 has a lower latency with 2, 3 and 4 as they are all connected to the same switch and therefore only one hop away from one another. Since Nodes 5 through 8 are connected to a different switch, they are more hops away from Node 1 which explains the higher latency. Although the numeric difference in latencies is negligible, the ratio between them corresponds to the distance between the two nodes.

Since we are using machines that are not optimized for traffic exchange, the total bandwidth capacity of our model is significantly lower than an actual IXP that can handle terabytes of data per second [8]. However, the traffic patterns will still depict those of an IXP and help analyze the capacity of this emulation. We use *iPerf3* [9] for testing bandwidth capacity for TCP traffic between the peering routers. An *iPerf* session is established between all possible pairs of nodes. We consider a node to be *active* if it is sending traffic to all other nodes. We make all nodes active at the same time and keep them active for 100 seconds. We observe that as time progresses, average network throughput remains fairly constant and shows a steady bandwidth.

To better analyze the effect of network congestion on average throughput, we conduct a similar experiment and activate one node every 10 seconds. Figure 3 shows a plot of the network-wide bandwidth average against time. With every added active node, the average bandwidth reduces. We see a steady bandwidth for 30 seconds during which all nodes remain active. From time 110 s, we start deactivating one node every 10 seconds which the average bandwidth gradually. The inconsistent jumps at time 60 s and time 160 s are due to a relatively higher bandwidth capacity for nodes 6 and 7. At time 60 s, activating Node 6 increases the overall network bandwidth. Similarly, at time 160 s, Node 7 becomes inactive and which causes the overall network bandwidth to drop.

To add resilience against routine failures, we use a fat-tree switch topology which provides an extra layer of switches for redundancy. If either one of the switches S1 and S2 fails, traffic converges to other switch to prevent network outage. We configure the switches to be in standalone mode which reduces load on the controller, and also allows continued traffic flow in case of controller failure. This design gives a buffer time to the IXP administrator to replace or reboot the controller without causing a major disruption of service. We also dynamically generate BGP configuration files to ensure transition in case of router failure. It should be noted that we implemented dynamic BGP configuration only for testing purposes but in reality an IXP can not control the BGP sessions among routers.

IV. Conclusion

We believe having an IXP-specific design platform will enable extensive model testing which will accelerate its development and deployment process. Additionally, it can streamline the Internet exchange evaluation process. We present the design and development specifications for an initial implementation of *GENIX* and we believe it offers great opportunity for future research and development in the area of IXPs.

References

[1] A. Dhamdhere and C. Dovrolis, "The internet is flat: modeling the transition from a transit hierarchy to a peering mesh," in *Proceedings of the 6th International COnference*, 2010, pp. 1–12.

[2] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 75–86, 2010.

[3] R. Klöti, B. Ager, V. Kotronis, G. Nomikos, and X. Dimitropoulos, "A comparative look into public ixp datasets," *ACM SIGCOMM Computer Communication Review*, vol. 46, no. 1, pp. 21–29, 2016.

[4] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "Geni: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5 – 23, 2014, special issue on Future Internet Testbeds â Part I.

[5] "FRRouting." [Online]. Available: https://frrouting.org

[6] "Open vSwitch." [Online]. Available: http://www.openvswitch.org

[7] "Floodlight." [Online]. Available: https://floodlight.atlassian.net/wiki/

[8] J. C. Cardona Restrepo and R. Stanojevic, "Ixp traffic: a macroscopic view," in *Proceedings of the 7th Latin American Networking Conference*, 2012, pp. 1–8.

[9] "iperf: Speed test tool," Online. [Online]. Available: https://iperf.fr/

Authorized licensed use limited to: University of Central Florida. Downloaded on October 09,2020 at 17:24:46 UTC from IEEE Xplore. Restrictions apply.