(Poster) LPEES: Lightweight Policy Enforcement and Evaluation for SDN-based Multi-Domain Communication

Abdulhakim Sabur, Ankur Chowdhary, Kritshekhar Jha, and Ming Zhao
Arizona State University

***Abstract*—While Software-defined networking (SDN) is widely used for intra-domain network communication, the inter-domain communication implementation relies on traditional routing approaches such as BGP-based routers. The role of the BGP router is to perform control and data planes functionality. This approach, however, prevents the administrator from exploiting the SDN benefits as the BGP routing table lookup and data packets forwarding can significantly degrade the data plane performance. This paper emphasizes the benefits of adopting fully SDN-based data plane packet switching by introducing LPEES, a lightweight policy framework for SDN-based inter-domain communication. LPEES limits the functionality of BGP only in the control plane. Also, the goal of LPEES is to manage global policies that traverses multiple domains like in the Wide-Area Networks (WAN). Our analysis shows that utilizing SDN-based systems for global networking prevents exploiting security vulnerabilities due to security policy conflicts that arises from integrating traditional networks with SDN-based networks.**

I. Introduction and Motivation

Boarder Gateway Protocol (BGP) has been the standard protocol for the internet networks. This approach limits the network performance to BGP capabilities and allows cyber adversaries to exploit the old protocol and cause network outage. For example, BGP router is still used for both the control plane and the data plane, making it an extra node along with the data plane communication path for inter-domain packet forwarding, where the communication throughput will be limited by the BGP routing table lookup time and processing delay.

A critical issue with inter-domain routing that has not been considered in the literature is the transit domain trustworthiness. The sender sends the traffic through a transit domain(s) without knowing whether or not the specified policies are actually enforced by the transit domains. For example, the source domain in Figure 1 does not trust the transit domain B and requires all of the network flow to be sent direct from transit domain A to the destination. Yet, the transit domain A violates this policy and send the network traffic to the transit domain B. The sender will not be able to know in this case that their policy is violated. This issue can be avoided if the sender has an approach to compute a trust-based routing decision such that it will know transit domain A cannot be trusted and an alternate route should be selected.

In this paper, we present a Lightweight Policy Enforcement & Evaluation Framework for SDN-based Multi-Domain framework (LPEES), that translates the inter-domain routing into the data planes' flow rules to address the above discussed

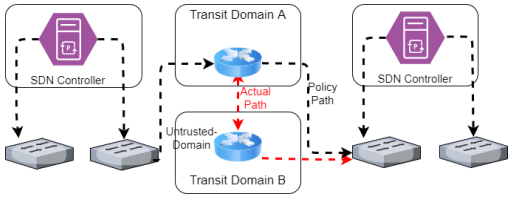

Fig. 1: An example of trust policy violation by the transit domain A.

multi-domain SDN issues. Although BGP and SDN integration problem is already solved [1], [2], [3]. Yet, the BGP router in the existing solutions is executing control and data planes functions, or the inter-domain solution is implemented using a centralized controller. Furthermore, LPEES approach does not reveal the domain's infrastructure information like the clustering or centralized controller implementation approaches. We show in this abstract our prelimnary results for enhancing the multi-domain communication delay, and we present the LPEES design which will be extended to include trust factor between multi-SDN domains.

II. LPEES Architecture and Workflow

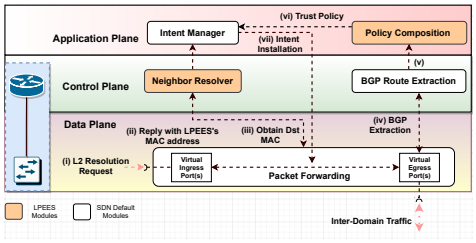

Fig. 2: The design and the internal communication flow in the presented LPEES framework.

In this section, we describe the LPEES framework's modules. The LPEES is a virtual entity that is composed of a BGP router and an SDN edge switch. Throughout the paper, we call the data plane switch/device in LPEES as the edge switch since it is located at the edge of the domain. In this work, *multi-SDN domains* represent each SDN domain with its own controller, and they do not share their infrastructure information with other domains.

A. policy composition

The *policy composition* module is implemented using python alongside the standard SDN controller implementation.

The module interacts with the *BGP route extraction* and *intent manager* modules that are in the SDN controller via REST APIs. The policy composition receives BGP updates from the *route extraction* module, route's attributes and custom policy attributes defined from the network administrator to compose the policy. Next, the composed policies in the form of flow rules and network intents are pushed to the controller. Hence, the policy composition module is an enabler for the inter-domain trusted policy framework.

1) Policy Formula: The policy in this paper is composed of two parts, mandatory and optional fields. The mandatory fields are written in the form of a tuple M = $< IP_s, IP_d, P_s, P_d, edgeP >$, where $IP_s \& IP_d$ are the source and destination IP addresses, respectively. $P_s \& P_d$ are the source and destination port numbers, respectively, and $edgeP$ is edge device's egress port number for this policy. Note here that we do not use the ingress port number as our goal is to handle inter-domain network traffic using the egress port. The optional fields are written in the form of a tuple $O =< opt_1, opt_2, ..., opt_n >$, where opt_i is an optional field such as MAC address, priority, VLAN ID, etc. The overall policy rule can be expressed as $P = \{M, O\}$.

2) Policy Compilation: Once the policy formula is established, the intent manager receives it and compiles it in the form of a *point-Intent*, because this intent type allows specifying the exact port on the edge device for network flow. As the number of ASes can be large, a direct mapping of high-level policies into intents is insufficient and may result in many generated intents. We present a group-based policy management approach to group similar forwarding behaviors into equivalence classes to address this issue.

The equivalence class has a set of IP prefixes that share the same forwarding behavior from the source domain to the destination domain. For example, domain *A* has a policy to send all traffic destined for host *h2* with TCP port *4444* through domain *C* and the rest of the traffic for the same host through domain *B*. Yet, if we rely on the IP prefix only to aggregate this requirement, it will not suffice since domain *B* might broadcast multiple IP prefixes ($p1 \& p2$ for instance) that are not contiguous IP address blocks.

B. Neighbor Resolver

The neighbor refers to the neighboring domain (inter-domain neighbor). The role of this module is to set up a virtual MAC address for LPEES. Because LPEES cannot rely on the IP address for the policy compilation, it uses the virtual MAC address to tag the outgoing packets. The motivation behind this design is due to the standard packet forwarding between the traditional routers, where the router specifies the next-hop MAC based on the resolution protocol. In the case of LPEES, the custom policy from the administrator might override this default value by specifying different *output* ports on the edge device, for instance. Hence, LPEES cannot rely on the MAC address received via the resolution protocol. This is why LPEES uses a virtual MAC address for each entity in the SDN domain.

C. Packet Forwarding

Once the policy is compiled into intent, the intent is converted into flow rules that are installed on the edge device. The last step is the *packet forwarding* across the inter-domain networks. The packet forwarding module is a core edge switch module and it perform *match/action* to compare incoming network flow rules to the existing flow table that is constructed based on the *policy composition*. However, because the custom policy is rewriting the default best routing path, the packet forwarding module cannot use the IP address to route the traffic. Also, the neighbor resolver changes the source MAC address. Hence, LPEES will rewrite the source MAC address with its virtual MAC address and the destination MAC according to the corresponding destination's address. The advantage of this method is that we maintain the destination MAC address in the forwarding table according to the final domain's edge device/router.

III. Evaluation and Future Work

Our preliminary evaluation of LPEES measures the communication delay in comparison to the traditional BGP routing approach. The tool *netperf* is used for this purpose. It is noticed from Figure 3 that as the number of SDN domain increases, LPEES framework has lower delay in comparison to the BGP approach by an average of **17.14%**. This delay drop is due to reducing the communication path by restricting the network flow from passing through the BGP router in the data plane. In the future, we plane to extend LPEES evaluation by measuring the effectiveness of the trust module and include security policy conflict management approach that can detect and resolve policy conflicts between the different SDN domains.

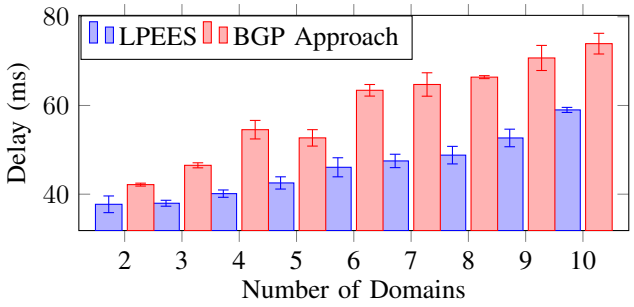

Fig. 3: The communication delay between the traditional BGP routing approach and LPEES in the serial topology. The experiment is an average of 10 emulation runs measured.

References

[1] R. Amin, M. Reisslein, and N. Shah, "Hybrid sdn networks: A survey of existing approaches," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3259–3306, 2018.

[2] X. Huang, S. Cheng, K. Cao, P. Cong, T. Wei, and S. Hu, "A survey of deployment solutions and optimization strategies for hybrid sdn networks," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 2, pp. 1483–1507, 2018.

[3] C. Chen, B. Li, D. Lin, and B. Li, "Software-defined inter-domain routing revisited," in *2016 IEEE International Conference on Communications (ICC)*. IEEE, 2016, pp. 1–6.