Enabling End Host Network Functions – Public Review

Z. Morley Mao
University of Michigan
zmao@umich.edu

Increasingly the functionality implemented within networks, e.g., load balancing and congestion control could benefit from the knowledge of the application-level messages. Traditional network management approaches rely on deep packet inspection or other heuristics to infer the application message a packet belongs in order to implement such network functions. Admittedly, this is both inaccurate and high overhead. End hosts appear to be an obvious alternative enforcement point to realize the network functions that make use of application data units or messages because they have fine-grained visibility into application semantics. This paper explores how to design an architecture at end hosts to achieve network functions aware of application semantics.

This paper proposes Eden, a new architecture for realizing network functions such as flow scheduling and datacenter QoS at end hosts with minimal network support. This architecture provides an architecturally cleaner realization of many network functions, which in turn can enable finer-grained and more accurate decision making. The authors have implemented a prototype that relies on modifying sockets to pass down application level class information to enclaves, which apply action functions based on a packet's class. In addition, the authors have also conducted preliminary experiments showing how Eden can implement some popular network functions such as load balancing with little overhead.

The reviewers find this work to be very timely, given the recent innovations in SDN and the growing amount of functionality support within the network. There is a clean separation among the different components, i.e., the logically centralized controller, Eden-compliant applications (stages), and end-host enclaves for executing the network function using match-action tables. This realizes application-aware data plane programmability, reminiscent of active networking, but restricted at end hosts only. The reviewers appreciate the insight made by the authors that there is a need for executing network functionality that must run at line rate, can be expressed as match/action rules, and can benefit from application knowledge as well as platform flexibility. No existing frameworks can explicitly address this challenge.

Besides the knowledge of application semantics, there are several other benefits for placing network functionality at end hosts, as pointed out by the authors. It's easier to achieve consistent behavior by enforcing a function at the traffic source compared to a distributed enforcement (which may be network path dependent). Only a subset of network traffic needs to be operated on, leading to a natural partitioning of state and computation overhead. Finally, it's simpler to infer certain network properties such as flow-level RTT.

At the same time, Eden leaves several avenues open for future work in this space. Despite the given definition of network functions in the paper, we have not seen the full potential for using the proposed architecture in terms of enabling new network functions. It also remains to be seen whether a variety of functionality implemented by middleboxes can all be realized by the Eden architecture. So far, the evaluations shown in the paper, have not demonstrated significant benefits to using Eden both in terms of new functionality enabled as well as improvement to the existing ones. Hence, it's unclear to argue for the tangible benefits of Eden, leaving room to explore new network functions that can fully utilize the advantage of end host based enforcement point.