ONIX: A Distributed Control Platform for Large-scale Production Networks

Ashutosh Mittal (E-mail: amittal@kth.se)

I. Paper summary

THIS paper introduces *ONIX* which is a platform for implementing network control planes in a Software Defined Network in environments as diverse as the WAN, the public cloud, and the enterprise data center. There are four components in an Onix controlled network, the first is physical infrastructure (switch, router, load balancers, etc.) which support basic connectivity and an interface allowing Onix to read and write state controlling the elements behavior. Second, connectivity infrastructure constituting channel to control messages flow. It should be bi-directional communication (in-band or out-of-band) optionally supporting convergence on link failure. Third, ONIX itself, which is distributed control platform to giving the control logic programmatic access to the network. Last, control logic, which is determining the desired network behavior and implemented on top of ONIXs API.

Onix's API models the network as a graph of objects referred to as the Network Information Base (NIB). Applications built on top of Onix defining the logic can read and update objects, and get notified if the state of the objects changes. NIB is replicated in multiple servers for backup and consistency purposes.

For *scalability* purposes it provides partitioning as well as aggregation in NIB. Further, it also provides applications with control over the consistency and durability of the network state using memory based DHT and replicated transactional databases. The paper also illustrates various ways of dealing with *reliability* issues like Network element and link failures, ONIX failure and Connectivity infrastructure failures.

The authors implemented Onix in roughly 150,000 lines of C++ code with ability to integrate a number of third party libraries. The paper illustrates evaluation of Onix with micro-benchmarks designed to test its performance as a general platform, and with end-to-end performance measurements of an in-development Onix application in a test environment.

The stellar paper provides an overall framework in which application programmers can easily write a management application controlling an SDN.

II. Significant Contributions

- Onix solves five most important challenges in building a production-quality control platform: Generality, Scalability, Reliability, Simplicity and control plane performance. Networks like Ethane, Distributed virtual switch, Multi-tenant virtualized data centers, and Scale-out carrier-grade IP router built on top of ONIX illustrates each of these strengths that it posses.
- Onix provides flexible distribution primitives (such as DHT storage and group membership) allowing application designers to implement control applications without re-inventing distribution mechanisms.
- User API built on Onix enjoy great degree of flexibility to make performance/scalability trade-offs as dictated by the application requirements.

III. Unresolved Issues

- Onix relies on API built on top to detect and provide conflict resolution of the network state. I believe at least conflict detection should be part of default Onix platform.