OpenNF: Enabling Innovation in Network Function Control

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

I. Paper summary

THIS paper introduces *OpenNF* which is a control plane architecture that provides efficient, coordinated control of both internal Network Function (NF) state and network forwarding state to allow quick, safe, and fine-grained real-location of flows across NF instances.

With the advent of SDN and NFV, hand-offs between different instances of NF are becoming more and more important. Unless this hand-off is efficient, reliable and cheap, software based control and dynamic resource allocation would not be possible. OpenNF address race conditions and accommodate a variety of application objectives and NF types. Its joint control is generally efficient even when applications have certain stringent requirements; allows applications to make suitable choices in meeting their objectives; and NFs need modest changes and incur minimal overhead when supporting OpenNF primitives. OpenNF controller encapsulates the complexities of distributed state control and, when requested, guarantees loss freedom, order-preservation, and consistency for state and state operations. It uses two novel schemes to overcome underlying race conditions: (1) an event abstraction that the controller uses to closely observe updates to state, or to prevent updates but know what update was intended, and (2) a two phase forwarding state update scheme.

OpenNF adds another dimension to the budding SDN-NFV niche and with some more research and extensive testing, this can become the standard for handling transfers in software controlled resource allocations.t

II. Significant Contributions

- OpenNF is very quick and efficient compared to then existing NF control planes. Thus, its no wonder that this paper won the The Applied Networking Research Prize (ANRP) 2016. It can cut NF scale-in time by tens of minutes compared to other control frameworks and state can be moved, copied, and shared efficiently even when certain guarantees. Evaluation shows that a loss-free move involving state for 500 flows takes only 215ms and imposes only 50ms of additional latency on packets received during the operation.
- It addresses the issue of race conditions comprehensively. To account for them, it introduces an event abstraction to externally observe and prevent local state changes inside NFs, and also a clever two-phase scheme for updating network forwarding state. This ensures that state updates are not lost or reordered during state moves and shared state remains consistent.
- It bounds overhead using a flexible northbound API that control applications use to precisely specify which state to move, copy, or share, and which guarantees to enforce additions to NFs. Also, its southbound API increase code size by at most 9.8%, and packet processing time at NFs increases by less than 6% during state export or import.

III. Unresolved Issues

- OpenNF is not very scalable. The average time per operation increases linearly with both the number of simultaneous operations and the number of flows affected. Also, it cannot automatically export which state should be moved and which should be not.
- This paper introduces yet another controller. Already developers were walking a tight rope with SDN and NFV working together. Now, for handling hand-offs another controller needs to be implemented. This would just result in a whole range of central controllers, one for each application. It would be much better if this can be offered as an integration or extension to existing state-of-the-art controllers.
- It doesn't work at the granularity of chain of middleboxes. If suppose there is some issue in one the boxes in a chain and a reclassification needs to be done starting from a prior box, but that box might already be migrated. Thus, there should be an option in which controller can work above the level of individual devices and coordinate between a series of them.