Header Space Analysis: Static Checking For Networks

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

I. Paper summary

THIS paper deals with development of a general and protocol agnostic framework called Header Space Analysis (HSA). It allows static checking of network specifications and configurations to identify an important class of failures such as Reachability Failures, Forwarding Loops and Traffic Isolation and Leakage problems. In HSA, protocol header fields are not first class entities; instead it looks at the entire packet header as a concatenation of bits without any associated meaning. Each packet is a point in the $\{0,1\}^L$ space where L is the maximum length of a packet header, and networking boxes transform packets from one point in the space to another point or set of points (multicast).

HSA expresses every region of the header space in terms of wildcard expressions and then can use algebra for working with them: Intersection, Complementation and Difference, Checking subset and equality. The implementation has been tested on Stanford network and given impressive results.

II. Significant Contributions

- In a time where SDN is being widely implemented, using a number of protocols at the same time has become rather easier and prevalent. Thus, HSA is good framework as it is a general framework that has protocol independent implementation.
- With networks going as big and complex, using manual inspection or primitive tool for verification is beyond comprehension. HSA comes across as a strong armament for network engineers to verify the system's Reachability, Looping, Isolation etc.
- The paper also introduces a handy *open source* python based library called Hassel. The library has well written header space and transfer function classes which can implement reachability, isolation and looping checks in less than 50 lines of code!

III. Unresolved Issues

- The graphical representation does not specify how to deal with network elements (like NAT boxes) which have statefull rules. It only deals with stateless elements ie. static transformations.
- Another obvious issue is the time taken for computation of the algorithm. It is not meant for real time analysis, only static. Also, it doesn't specify the point of error even if detects one. Suppose if routing algorithm is broken because routing tables are inconsistent, it does not tell us why.
- It cannot deal well with churn in the network, except to periodically run it based on snapshots. Thus it can only detect problems that persist longer than the sampling period.