Review of Header Space Analysis: Static Checking For Networks

Yusu Weng

KazemianPeyman, VargheseGeorge, & McKeownNick. (2012). Header Space Analysis: Static Checking For Networks. NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation.

1 Introduction

In NSDI'12 Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation, Kazemian, Varghese and Mckeown proposed a method called Header Space Analysis to check network configuration and identify typical kinds of failures such as Reachability Failures, Forwarding Loops and Traffic Isolation and Leakage problems

2 Summary

Header Space Analysis is based on a geometric model as following:

Header Space, H

We regard the header as a binary sequence of packet's header in $\{0,1\}^L$ and ignore its protocol-specific meaning.

Network Space, N

Network Space N is represented as $\{0,1\}^L$ X {1, ..., P}. Here {1, …, P} is the list of all ports in the network.

Network Transfer Function, Ψ() and Topology Transfer Function, Γ():

We use function T to represent a node's transfer function:

$T(h, p) : (h, p) \rightarrow \{(h1, p1),(h2, p2), ...\}$

So for all nodes in the network, we can use function Ψ to represent it:

$\Psi(h, p) =$

$T1(h, p)$ if $p \in$ switch1

... ...

$T_n(h, p)$ if $p \in switch_n$

another function Γ can represent link's transfer function:

$$\Gamma(h, p) = \begin{cases} \{(h, p^*)\} & \text{if } p \text{ connected to } p^* \\ \{\} & \text{if } p \text{ is not connected.} \end{cases}$$

Based on this geometric model, the article then define some basic operations like intersection, union and complementation to support the analysis. At last, the author also wrote a tool and implemented it on a network and found some potential failures.

3 Contributions

This article was published in 2012 when SDN was still at an early stage. The method proposed by this article can represent a practical use of SDN framework, diagnosing potential problems and failures in a network. Unlike other solid design or implementation, this article aims to solve a more deep-going problem: given all network configurations (e.g. forwarding rules), it can detect all potential failures such as loop in the network.

On the technical level, this problem is not difficult to solve since if we have known configurations of all nodes, we can just implement a traversal based on all possible conditions. However, what I think this article cool is that it promotes it to a mathematical level by considering packet forwarding as spatial alternation. If we have x bits in the header, we could assume it as an x-dimensions space and packet forwarding is that mapping a subspace to another.

The geometric insight in this article can be reused in many other fields， a similar example I known is Symbolic Execution in dynamic program analysis. It assumes that the input is a symbolic value and each control statement is a transfer function so that the program will map the input value to a symbolic expression as an output. Unlike HSA which is a white-box, here we do not have any knowledge about the program, and we need to build a Symbolic Execution Tree to reveal the structure. I believe that a similar method can be used to discover network structure in the future.

4 Limitations

As mentioned in the limitation section, HSA can only be used in the static analysis, and it cannot normally work when network configurations are dynamic, or network topology is changing.

Another potential problem is complexity. For example, the running time of reachability analysis is $O(dR^2)$ which is still an exponential growth even though some tricks are used like lazy subtraction to reduce running time.

At last, HSA is designed based on the point-to-point mapping and it didn't consider multiple mapping, which means that a header space h can be mapped to h1, h2 …, hn after forwarding. In this case, some basic operations like range inversion will be invalid and the following analysis cannot go on. E.g. if (h, p) can map both (h1, p1) and (h2, p2) in a switch, then the switch will forward it to 2 ports. Another case is that both (h1, p1) and (h2, p2) are mapped to (h3, p3) in a switch. To deal with such problems, the geometric model should be improved by adding multiple mapping in transfer function and modifying set operations.