SDN DEBUGGER

CS 6250 Computer Networks - Project Milestone 1

Domain: Software Defined Networking

Software defined networking (SDN) is a novel and trending approach to build computer networks by decoupling the network system that makes decisions about the control plane from the underlying system that forwards traffic to the data plane. This architecture provides programmable central control of network traffic without requiring physical access to the network's hardware devices.

SDN has emerged as a new paradigm of networking and it has the potential to redefine the future of networking. This project would provide us with a nice learning platform and expose us to a variety of technologies like OpenFlowJ, Mininet and POX/NOX Controllers associated with SDN. After using debuggers extensively for software development and having appreciated its prowess, we are excited to explore the same with software defined networking.

Problem Scope

Networks are generally hard to debug, the existing tools such as ping and trace-route do not provide us with enough information to isolate bugs in the system. Also tools such as tcp dump and NetFlow are passive in nature and attempt to reconstruct the complex and distributed state of the network in an adhoc way. These challenges exist in SDNs as well but SDNs provide us an opportunity to rethink how networks can be debugged.

Proposed Solution

A tool similar to Linux gdb providing options like breakpoint and backtrace for debugging software defined networks would be of great benefit to network administrators, framework and application developers. This tool primarily aims at diagnosing bugs like control logic errors, network race conditions and configuration mistakes. Moreover, it can be used as a statistical analyzer using network metrics like path lengths and switch hops stats, thereby improving load balancing and network management. Since debugging would be performed by live monitoring, this approach would avoid the overhead caused by recreating the complex network state. Further, we also plan to extend this tool by providing additional debugging features like forward trace and watch.

References

[1] Nikhil Handigol, Brandon Heller, Vimalkumar Jeyakumar, David Mazières and Nick McKeown. *Where is the Debugger for my Software-Defined Network?*

[2] Bob Lantz, Brandon Heller and Nick McKeown. *A Network in a Laptop: Rapid Prototyping for Software Defined Networks.*

Team Members

Ishwarya Balaji Gururajan	GT ID: 902916905
Lokesh Balakrishnan	GT ID: 902883749
Shunmugam Murugan	GT ID: 902936921
Prabu Shyam Mayavaram Mahalingam	GT ID: 902612550
Dhananjayan Ramesh	GT ID: 902892838