Paper Evaluation, A NICE way to Test OpenFlow Application

Jiani Jiang <jianij@kth.se>

1. Paper summary

The paper focuses on a more efficient, systematic techniques for testing unmodified controller programs due to the problems of the risks of bugs and software faults that will cause unreliable communications and hurt the performance of SDN. The authors use their NICE tool to do model checking and makes it possible to explore the possible states a network as a whole can go through, while symbolic execution examines the code paths to identify equivalence classes of packets (relevant input packets) for testing. More specific, NICE combines model checking (to explore system execution paths), symbolic execution (to reduce the space of event orderings). Their NICE applications are written in Python and they carried out evaluations on small examples which shows NICE is five times faster than approaches that apply state-of-the-art tools, and applied NICE to three real OpenFlow applications and uncover 11bugs.

2. Top 3 contributions

1. It proposes a method to test OpenFlow in a SDN network and helps the network be more dependable.

2. Make it possible to identify bugs like forwarding loops or black holes in the emerging OpenFlow controller applications. It it programmable and the programmer can configure the desired search strategy.

3. It provides the functionality to explore state-space more efficient and can help to do some specify correctness which can relief network operators from large quantity works.

3. Problems

1. Due to the number of transitions in any given state increases with an additional switch, it is really hard for NICE to scale with increasing the number of network switches.

2. The evaluation is not sufficient. It did not be tested in a large typical SDN network and the performance of it is unknown.