International Journal of Scientific Engineering and Applied Science (IJSEAS) - Volume-1, Issue-2, May 2015
ISSN: 2395-3470
www.ijseas.com

A SOFT SCALABLE WAY FOR SERVICE FOR TESTING AND MONITORING LARGE NETWORK SYSTEMS

Sabiha[1],S.Rajiya sulthana[2]
[1]CSE M.Tech, BCETW,AP,India
[2]CSE M.Tech, BCETW,AP,India

ABSTRACT

Networks are getting larger and more complex,yet administrators rely on rudimentary tools such as and to debug problems. We propose an automated and systematic approach for testing and debugging networks called "Automatic Test Packet Generation" (ATPG). ATPG reads router configurations and generates a device-independent model. The model is used to generate a minimum set of test packets to (minimally) exercise every link in the network or (maximally) exercise every rule in the network. Test packets are sent periodically, and detected failures trigger a separate mechanism to localize the fault. ATPG complements but goes beyond earlier work in static checking (which cannot detect liveness or performance faults) or fault localization (which only localize faults given liveness results). We describe our prototype ATPG implementation and results on two real-world data sets: Stanford University's backbone network and Internet2. For example, 4000 packets can cover all rules in Stanford backbone network, while 54 are enough to cover all links. Sending 4000 test packets 10 times per second consumes less than 1% of link capacity. ATPG code and the data sets are publicly available.

Index Terms—Data plane analysis, network troubleshooting, test packet generation.

INTRODUCTION

Every day, network engineers wrestle with router misconfigurations, fiber cuts, faulty interfaces, mislabeled cables, software bugs,intermittent links, and a myriad other reasons that cause networks to misbehave or fail completely. Network engineers hunt down bugs using the most rudimentary tools (e.g.ping,traceroute, SNMP, and) and track down root causes using a combination of accrued wisdom and intuition. Debugging networks is only becoming harder as networks are getting *bigger* (modern data centers may contain 10 000 switches, a campus network may serve 50 000 users, a 100-Gb/s long-hauland are getting *more complicated.*

ATPG detects and diagnoses errors by independently and exhaustively *testing* all forwarding entries, firewall rules, and any packet processing rules in the network. In ATPG, test packets are generated algorithmically from the device configuration files and FIBs, with the minimum number of packets required for complete coverage. ATPG can adapt to constraints such as requiring test packets from only a few places in the network or using special routers to generate test packets from every port. ATPG can also be tuned to allocate more test packets to exercise more critical rules.

EXISTING SYSTEM

We are unaware of earlier techniques that automatically generate test packets from configurations. The closest related works we know of are offline tools that check invariants in networks. In the control plane, NICE attempts to exhaustively cover the code paths symbolically in controller applications with the help of simplified switch/host models. In the data plane, Anteater models invariants as boolean satisfiability problems and checks them against configurations with a SAT solver.

End-to-end probes have long been used in network fault diagnosis in work such as shown in previous system. Recently, mining low-quality, unstructured data, such as router configurations and network tickets, has attracted interest. ATPG improves the detection granuality to the rule level by employing router configuration and data plane information.

PROPOSED SYSTEM

In the proposed system, the system is what we call an Automatic Test Packet Generation (ATPG) framework that *automatically* generates a minimal set of packets to test the liveness of the underlying topology *and* the congruence between data plane state and configuration specifications. The tool can also automatically generate packets to test *performance* assertions such as packet latency.

Test packets are fed into the network so that every rule is exercised directly from the data plane. Since ATPG treats links just like normal forwarding rules, its full coverage guarantees testing of every link in the network. However, all-pairs ping does not guarantee testing of all links and has been found to be un scalable for large networks such as Planet Lab.

CONCLUSION

Testing liveness of a network is a fundamental problem for ISPs and large data center operators. Sending probes between every pair of edge ports is neither exhaustive nor scalable. ATPG, however, goes much further than liveness testing with the same framework. Our implementation also augments testing with a simple fault localization scheme also constructed using the header space framework. We hope network ATPG will be equally useful for automated dynamic testing of production networks.

REFERENCES

[1] “ATPG code repository,” [Online]. Available: http://eastzone.github.com/atpg/
[2] “Automatic Test Pattern Generation,” 2013 [Online].Available: http://en.wikipedia.org/wiki/Automatic_test_pattern_generation
[3] P. Barford, N. Duffield, A. Ron, and J. Sommers, “Network performance anomaly detection and localization,” in *Proc. IEEE INFOCOM*,Apr. , pp. 1377–1385.