ISSN: 2348 - 2117

International Journal of Engineering Technology and Computer Research (IJETCR)
Available Online at www.ijetcr.org
Volume 3; Issue 6; November-December-2015; Page No. 34-36

Review on Automatic Test Packet Generation

Ashwini Bhore[1], Gayatri Bhandari[2]

[1]JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi Pune 412 207

ashwinibhore1@gmail.com

[2]JSPM's Bhivarabai Sawant Institute of Technology and Research, Wagholi Pune 412 207

Abstract

Now a day's networks are increasing in size and it is becoming a difficult job for network administrators to debug the network, since they rely on traditional tools such as ping and trace route for this job. This paper puts an automated and systematic approach to test and debug a network which is called as Automatic Test Packet Generation (ATPG). ATPG produces a model which is independent on devices after reading configuration from routers. The model is used to generate a minimum set of test packets to exercise every link in the network or exercise every rule in the network. ATPG is capable of analysing both functional and performance problems. The working of few offline tools which automatically generate test packets are also given, but ATPG goes beyond the earlier work in static checking.

Index Terms: performance analysis, test packet generation.

Introduction

It is difficult task to debug a network. The network administrators face problems like Fibre cut, mislabelled cables, router misconfiguration, software bug, Faulty interfaces etc. Network administrators try to overcome these problems using tools such as ping and trace route. Debugging networks is getting more and more difficult as not only size of networks but also their level of complexity is increasing day by day.

Let us consider few examples of different types of problems network administrators face in day to day life.

Consider router with a line card having a fault, so it silently drops test packets, as a result, many users straggling for connections complain to network administrator. Now if that administrator is administrating 100 routers he has to go to each router to see if configuration is not altered, and if the answer is no, he uses his knowledge of topology to search faulty device using techniques such as ping and trace route .

Consider one more example where video traffic is put in a particular queue, and token bucket ratio is low and the reason why packets are dropped. Such performance faults are impossible for network administrators to investigate. All responses to that survey is given .From the survey it is clear that administrators have fight with complex symptoms and causes. Many problems associated with networks occur frequently and it takes much time to overcome of them, so the cost of debugging a network becomes insignificant.

This paper put forward an automated and systematic approach for testing and debugging a network called Automatic Test Packet Generation (ATPG). ATPG produces a model which is not dependent on devices after reading the configuration from routers. Another advantage of ATPG system is that it covers each link and rule in the network with minimum number of test packets. Uniformly the test packets are send, and if any fault is detected, it is triggered by separate mechanism namely fault localization. ATPG can solve both of above problems, hence it can cover both functional and performance faults.

The figure 1 is uncomplicated view of network state. In the lower half of figure there is forwarding table. The function of forwarding table is to forward packets. Packet is consists of forwarding information base (FIB), access control lists etc. It is control plane which writes forwarding state.

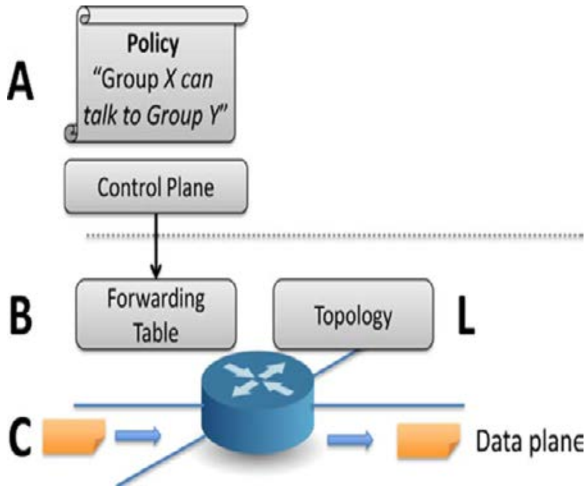

Figure 1: Network state

Figure 1 can be decomposed mainly in three parts as A, B and C.
We can consider the policy (A), which is compiled by controller into configuration files which are device specific (B), and then shows the forwarding behaviour of every packet (C). To ensure that whether network behaves as per requirement, all the three steps at all times should remain persistent, that is same as A=B=C. At a same time, the topology, as shown in the figure, should also be able to satisfy a set of liveness properties shown by L.

It is not too long ago when scientists come up with tools showing compactness between policies and configuration files A=B, but these tools can't deal with performance problems which require checking of liveness property L or B=C. ATPG can do this job efficiently.

1. RELATED WORK:

In this section some of the earlier techniques used for automatically generating test packets are given.
A) Offline Tools Supporting Automatic Test Packet Generation
B) Header Space Analysis
C) Overview of the system

A) Offline Tools Supporting Automatic Test Packet Generation

One of the offline tools which have been used for generating test packets automatically in control plane is NICE. NICE stands for "no bugs in controller execution". NICE brings the bugs in controller program to user's notice more efficiently with the help of model checking and symbolic execution in open flow system. Working of NICE is shown in Figure 2. NICE programmer has to supply controller program along with topology of network which consist of specification of switches and hosts. The NICE according to fixed plan looks into the possible system behavior and checks it with correctness properties supplied by the programmer. Finally NICE gives the traces of property violation or properties which are not up to the mark with their indications as output.

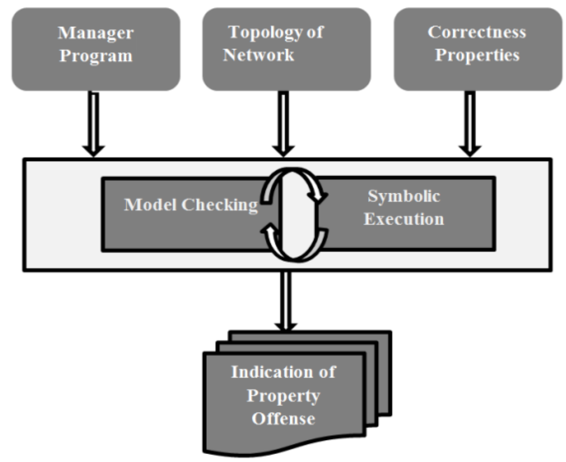

Figure 2: No bugs In Controller Execution (NICE)

B) Header Space Analysis

The automatic test packet generation uses the framework of Header space analysis, in which it uses a geometric model, which allows the ATPG system to statistically check the network specifications and configurations to deal with important classes of failures such as forwarding loops, reachability failures, traffic isolation and linkage problem.

Another advantage of header space analysis is capability to do slicing. Slicing assures isolation between system hosts, users or traffic. Each slice has the separate control plane, and it is up to its owner to decide how packets are routed and processed in that slice.

C) Overview of the system

a) Origination of Test Packets
b) Error Fixing
Figure 3 represents the work flow of automatic test packet generation (ATPG) system.
1) The ATPG system begins by gathering the forwarding state from network, which is represented as first step in the figure..

2) The second step follows the first, in which header space analysis is used by ATPG system to figure out scope of each terminal.
3) The outcome of second step is taken as input by test packet generation algorithm to gauge smallest number of test packets sufficient to test all rules
4) These test packets are sent regularly by test terminals as a penultimate step.
5) Lastly, if an error is disclosed ATPG appeals to fault localization algorithm to curtail root of error.

a) Origination of Test Packets

1) ATPG system begins by estimating the entire set of test packet headers that can be forwarded from each test terminal to every other test
2) Afterwards, ATPG selects greater than or equal to one test packet from identical class of test packets to use every rule which is within reachable distance. This method is capable of finding only those faults for which all packets screened by same rule suffer the same fault.
3) Lastly in the process of generating test packets ATPG goes to compression.

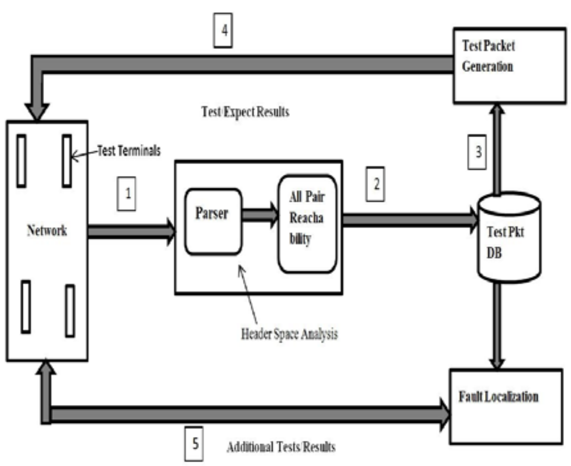

Figure 3: Working of Automatic Test Packet Generation

b) **Error Fixing**

ATPG sends the set of test packets at regular intervals. If in case test packets fail to reach their desired target, ATPG is capable of identifying errors that induced the problem. If watched performance of a rule is different from its normal behaviour then a rule is neglected, in other words it fails.

5.

ATPG monitors that where rules fail by applying a result function R on rule r in a packet pk. A result function takes value 1 if packet pk follows rule r, if not it takes value 0. A forwarding of a rule fails if a test packet is not provided to its planned output port on the other extreme. Forwarding of a rule is successful if either a test packet is provided to its planned output port, or in case it is a drop rule, it is addressed rightly if it is dropped. A link collapse can be characterized by failure of forwarding rule in topology function.

2. Importance

With the help of ATPG system exactness can be improved by testing functional and performance problems. The set of test packets generated in ATPG system can cover each reachable rules in the network, taking into account all port and headers restrictions. To find the faulty test terminals with its rule as well as configurations, fault localization algorithm is used in ATPG.

3. CONCLUSION

In this paper, we have analyzed that ATPG is able to test reachability strategy by testing all rules.
Using performance scales such as delay and loss of test packets ATPG can calculate performance soundness of a network. It uses simple fault localization method constructed with the help of header space analysis to localize faults. The Regular model of ATPG system helps to cover maximum links or rules in a network with minimum number of test packets.

4. REFERENCES

1. Hongyi Zeng, Peyman Kazemian, George Varghese and Nick McKeown," Automatic Test Packet Generation," IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 22, NO. 2, APRIL 2014.
2. "Automatic Test Pattern Generation,"2013 [Online]. Available: http//en.wikipedia.org
3. C. Cadar, D. Dunbar, and D. Engler, "Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs,"
4. M. Kuzniar, P. Peresini, M. Canini, D. Venzano, and D. Kostic, "ASOFT way for OpenFlow switch interoperability testing," in Proc.ACM CoNEXT, 2012.