

A Competent Automatic Test Packet Generation and Fault Localization for Network Model

Bura Vijay Kumar^{1*}, Nagendar Yamsani², Srinivas Aluvala³, Dr. K. Seena Naik⁴

^{1,2,3,4}Department of Computer Science and Engineering, S R Engineering College, Warangal Urban, Telangana, India

*Corresponding author E-mail: vijaykumar.bura@gmail.com

Abstract

The developing of the fault locating in the network brought about tremendous changes for business models, industry, science and technology applications. The implementation of Automatic test Packet Generation (ATPG) in the network tests for faultiness in the network. It test for liveliness in the network at runtime with the aid of test packets by suitable rules namely forwarding, drop and congestion and fix the problem in the network so that one can easily locate the link failure at each and every terminal in the network in less time. The simulations are carried out with the help of jdk tool to evaluate the port failures in all cases of the rule. The results shows the efficient performance in case of fault localization algorithm that outperforms other traditional approaches.

Keywords: Data plane analysis, network troubleshooting, test packet generation

1. Introduction

In the days the model is to produce a least possible set of packets to access every link in the network maximally or minimally to form of interconnection which places the situation for its exactness and optimality this model form of interconnection and the main contributions will correspond to beta test pattern generation ATPG outline that automatically changes a minimal set of packets to test the levels with the underline topology and the conversation between the data plane set and the configuration specification and starting with us and now it's going to display the single link. As the networks produce the erroneous operations due to improper and as well inadequate administration happened at any time of the network. By applying the proper fault location mechanisms the problems can be solved at ease in less time efficiently. As per previous literature several authors worked hard to find suitable fault mechanism to exactly locate in the network but there is no much concentration on the hard and soft error that most of the engineers are facing problem today. To cope this an efficient framework was organized that main objective is to exactly find the location of fault in the network and apply the rules to solve the problem.

A framework which could be used for model testing and debugging Networks makes no sense to put a minimal set of packets to exercise each and every connection in the network as well as some different network rules.

ATPG recognizes blunders by diverse and independently testing all forwarding security firewall models and any parcel handling show within the arrange test packets are produced algorithmically from the gadget setup less based with the least number of information required for total security. It utilized for testing the insipient information of the basic topology and the coinciding between information plane state and set up models. It moreover complements work in inactive checking and mistakes localization. The device can naturally produce parcels to test comes about attesta-

tions such as bundle misfortune within the current framework, the director physically chooses which ping parcel to be sent. Sending programs among each match off boundary ports is not one or the other wide nor versatile. This framework is satisfactory to discover the slightest set of conclusion to conclusion parcels that travel each connect. In any case, does this require a way of abstracting over device-specific setup records [6], [7] creating headers and joins they reach and at last calculating the least set of test packets. It isn't outlined to distinguish disappointments caused by fade away joins and switches, bugs caused by flawed switch equipment or computer program, and execution issues. The common causes of organizing disappointment are equipment disappointments and computer program bugs, in which that issues [11], [12] show both as reachability disappointments and through-put/latency corruption. To overcome this we are proposing a modern framework.

The main contributions and organization of this paper are summarized as follows: In section 2 we describe background details of different Automatic Test Packet Generation schemes. The section 3 proposed work. The section 4 results and discussion. Finally in section 5 we concluded the paper.

2. Background work

Assume that video activity is mapped to a particular line in a router, but the packets are dropped since the token bucket rate is as well arranged. It isn't at all clear how Alice can track down such an execution blame utilizing ping and follow the route. Investigating arrange is troublesome for 3 reasons. To begin with, the sending state is disseminated over numerous switches and firewalls and is characterized by their for-warding tables, channel rules, and other setup parameters. Moment, the sending state is difficult to watch since it regularly requires physically logging into every box within the organize. Third, there are numerous distinctive programs, conventions, and people overhauling the sending state at the same time. When Alice uses ping and follow the course, she is

employing a rough focal point to look at the current for-warding state for clues to track down the disappointment.

In [3], the authors Link delays and faults in a Service Provider or Enterprise IP network. Two-phased approach attempts to minimize both the monitoring infrastructure costs as well as the additional traffic due to probe messages. In [4], the authors its time consuming n complex procedure to user. Implement a tool KLEE, capable of automatically generating tests that achieve high coverage on a diverse set of complex.

In [2], the authors Risk of programming errors that make communication less reliable. Implement a single controller program manages the network, seems to reduce the likelihood of bugs. In [5], the authors Location of performance anomalies (e.g., high jitter or loss events) is critical to ensuring the effective operation of network infrastructures. Framework for detecting and localizing performance anomalies based on using an active probe enabled measurement infrastructure deployed on the periphery of a network.

3. Proposed framework

The proposed scheme basic working introduces the collection of test packets there main job is to test each and every rule can be applied to all aspects of the network. In the given test packet after inspecting, at least one will success to detect the faultiness in the network. Upon detecting the defective nature the ATPG only produces the little amount of test packets that can assist by the packet to be forwarded rule with necessary remedies to fix the problem immediately in the network links [1]. Referring to the Fig.1, let us introduce the systematic block diagram of the ATPG system. The initial step of the approach is to accumulate every forwarding state can be obtained from the network then performs the given test present in the network.

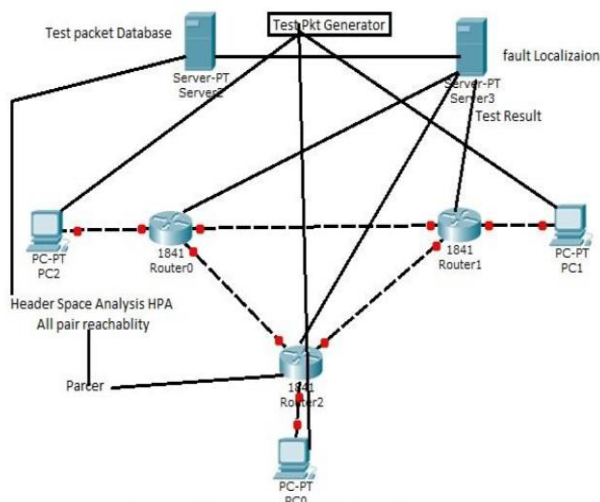


Fig.1: Proposed Automatic packet generation and fault localization

Step 1- In this phase certain configuration files and get the required network topology so that uses sufficient header space to measure reachability among the terminals.

Step 2- The outcome subsequently test packet choice process to calculate a minimal set of test packets that can test all the proper rules.

Step 3 - The packets thus generated are shared regularly with the help of the test terminals

Step 4 – when it detects the fault in the network, the fault localization algorithm is applied to fix the problem.

We reconsidered the particular controller assembling the policy (A) into explicitly the required configuration files (B), thereby the working forwarding of each packet (C). To test the liveness of the network acts as premeditated, all the cases should keep on consistent during all the cases i.e. moreover, the topology, illus-

trated in the figure.1, ought to also meet a set of liveness properties.

Minimally, have need of that adequate links and nodes are correctly behaving as assumed, the control plane stipulates that a terminal can access the suitable server without any risk, the anticipated outcome can fail if links fail.

Test Packet Generation: Let us consider a collection of terminals can share their packets and as well receive in return test packets so that to make use of them for testing all the rules in the switch module. Also as mentioned earlier that main intention is to test every switch of the network for fault situation, if it detects fault in the network immediately as it limits the nature of every link in the queue. After generating it should satisfy the two key issues the primary one is to measure the terminals that are existed and at each terminal it is supposed to use the headers.

Generate All-Pairs Reachability Table: ATPG starts by computing the complete set of packet headers that can be sent from each test terminal to every other test terminal. It is apparent that the all the packets headers can be tested before sending from one terminal to the other terminal compulsory. To make it possible ATPG attach the set of rules for every path with the assistance of reachability approach as discussed earlier. Upon reaching to every terminal in the network it is attached with the suitable function for initial switch in the network so that it will pass to remaining test terminals with the header format.

ATPG Tool: The main functionality of this tool is produce the minimum quantity of particular test packets , thereby each forwarding rule is definitely tested by at least by one of the packet and upon finding the fault the suitable algorithm is applied by applying appropriate rules to the network.

Fault Localization: In this section ATPG shares a group of test packets at regular intervals, when it detects the fault it locates the fault information. Now it is to say the rule fails in case of its abnormal function as compares to normal operation. So therefore the ATPG sets monitoring of all the rules for its failure with the pre-defined function that decides the pass or fail in case of working rules. In this sense the forward rule fails in the situation of packet undelivered to the intended port, at that time the drop rule exactly works good due to dropped packets

Procedure:

- 1) Add new connection to add new nodes into the network.
- 2) Enter the IP address of the node, node name and port number of that node. After successfully configuring the node.
- 3) Run the nodes, to run any node in the network go to nodes folder and execute the application.
- 4) Enter the port number of a node that has to start.
- 5) Running a node with port number 5000, then followed with the port number 5001.
- 6) As the A & B nodes are started the ping will be successful.

4. Results and discussion

The simulations carried with the aid of java jdk7 for comparing the performance of algorithms namely traditional s our proposed based on the parameters in the network across all the neighbor nodes. Java engineering gives a convenient, strong, tall performing environment for advancement. Java gives transportability by compiling the bytecodes for the Java Virtual Machine, which is at that point deciphered on each stage by the run-time environment. Java could be an energetic framework, able to load code when required from a machine within the same room or over the planet.

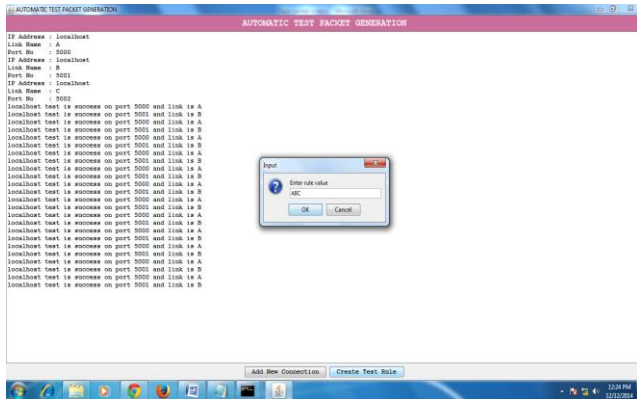


Fig.2: Testing for the nodes A, B and C

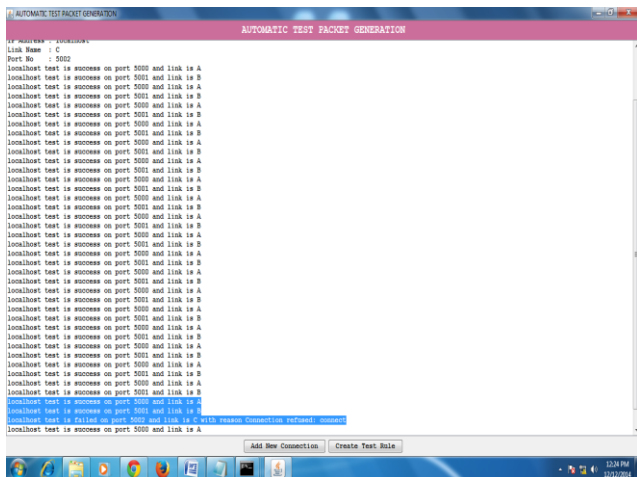


Fig.3: Final Testing for the nodes A, B and C

5. Conclusion

It became a tuff job for several engineers to measure the network liveliness because of real time packet generation for all internet service providers at all data centres. Due to high risk and complexness in measuring such modules for accurate results for higher data rates can be achieved in the network. So by introducing the automatic test pattern generation (ATPG) solves major problems of network due to its ability to test the liveliness of the network rules even it test the reachability of the policy. The work done in this paper recommends so many remedies that are useful for the suitable augments testing with a low cost and easy to implement fault localization approach that explicitly constructed with the aid of header space structure.

Acknowledgement

Authors would like to express sincere gratitude to management and principal of SR Engineering College, for their support and encouragement to carry out the research work

References

- [1] Zeng, Kazemian, Varghese, and Nick "Automatic Test Packet Generation", VOL. 22, NO. 2, APRIL 2014.
- [2] N. Duffield, "Network tomography of binary network performance characteristics." IEEE Trans. Inf. Theory, vol. 52, no. 12, pp. 5373–5388, Dec. 2006.
- [3] Y. Bejerano and R. Rastogi, "Robust monitoring of link delays and faults in IP networks," IEEE/ACM Trans. Netw., vol. 14, no. 5, pp. 1092–1103, Oct. 2006.
- [4] C. Cadar, D. Dunbar, and D. Engler, "Klee: Unassisted and automatic generation of high-coverage tests for complex systems programs," in Proc. OSDI, Berkeley, CA, USA, 2008, pp. 209–224.
- [5] P. Barford, N. Duffield, A. Ron, and J. Sommers, "Network performance anomaly detection and localization," in Proc. IEEE INFOCOM, April pp. 1377– 1385.
- [6] Deepika Vodnala, Dr. S. Phani Kumar, Srinivas Aluvala, "An Efficient On-Demand Link Failure Local Recovery Multicast Routing Protocol" in Advances in Intelligent Systems and Computing, Volume 467, Issue 1, Page No(s) 603 - 616, OCT. 2016, [ISSN (Print):978-981-10-1644-8], DOI: 10.1007/978-981-10-1
- [7] Srinivas Aluvala, "Clustering Based on Similarity Measures Using Concept Data Analysis" in international Journal of Advanced Research in Computer Science and Software Engineering(IJARCSSE), Volume 6, Issue 1, Page No(s) 656 - 660, JAN. 2016, [ISSN (Print):2277- 128x]
- [8] N. G. Duffield and M. Grossglauser, "Trajectory sampling for direct traffic observation," IEEE/ACM Trans. Netw., vol. 9, no. 3, pp. 280–292, Jun. 2001.
- [9] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in Proc. ACM SIGCOMM, 2011, pp. 350–361.
- [10] "Hassel, the Header Space Library," [Online]. Available: <https://bitbucket.org/peymank/hassel-public/>
- [11] Gowdiperu Sucharitha, Srinivas Aluvala, Deepika Vodnala, Y.Nagender, "A Survey on Mobile Ad-hoc Social Networks for Efficient Data Query in Intermittently Connected System" in International Journal on Innovative Research in Computer and Communication Engineering , Volume 3, Issue 11, Page No(s) 11831 - 1183, NOV. 2015, [ISSN (Print):2320 - 9801], DOI: 10.15680/IJRCE.201
- [12] D.Mahesh, Y.Nagender, "DIFFICULTIES OF MANET FOR MOBILE SOCIAL NETWORKS" in International Journal on Computer Science and Engi, Volume 10, Issue 1, Page No(s) 1 - 6, JAN. 2018, [ISSN (Print):2229-5631, ISSN (Online): 0975-3397].