International Journal of Research
International Journal of Research
Available at https://edupediapublications.org/journals
p-ISSN: 2348-6848
e-ISSN: 2348-795X
Volume 03 Issue 14
October 2016

Packages Automatic Test Generation

[1]GUNDEBOINA UPENDER [2]B NAVEEN KUMAR

[1] Pg Scholar , Department Of Cse. Gandhi Academy Of Technical Education, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206.

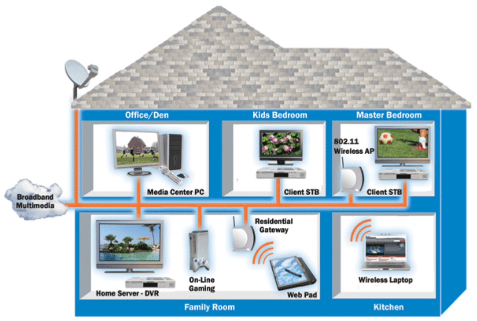
[2]Assistant Professor, Department Of Cse. Gandhi Academy Of Technical Education, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206

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 can detect both functional (e.g., incorrect firewall rule) and performance problems (e.g., congested queue). 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. We find that a small number of test packets suffice to test all rules in these networks: 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 consume less than 1% of link capacity. ATPG code and the datasets are publicly available.

1 INTRODUCTION

What is networking?

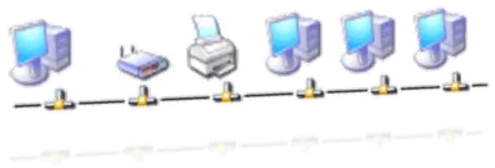
Networking is the word basically relating to computers and their connectivity. It is very often used in the world of computers and their use in different connections. The term networking implies the link between two or more computers and their devices, with the vital purpose of sharing the data stored in the computers, with each other. The networks between the computing devices are very common these days due to the launch of various hardware and computer software which aid in making the activity much more convenient to build and use.

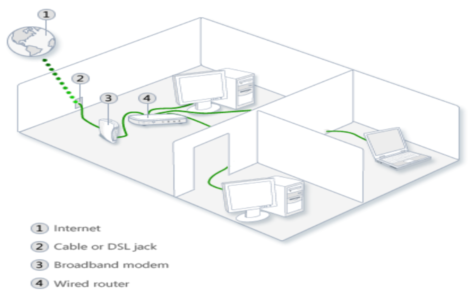


Structure of Networking between the different computers

How networking works?

General Network Techniques - When computers communicate on a network, they send out data packets without knowing if anyone is listening. Computers in a network all have a connection to the network and that is called to be connected to a network bus. What one computer sends out will reach all the other computers on the local network.





Above diagrams show the clear idea about the networking functions

For the different computers to be able to distinguish between each other, every computer has a unique ID called MAC-address (Media Access Control Address). This address is not only unique on your network but unique for all devices that can be hooked up to a network. The MAC-address is tied to the hardware and has nothing to do with IP-addresses. Since all computers on the network receives everything that is sent out from all other computers the MAC-addresses is primarily used by the computers to filter out incoming network traffic that is addressed to the individual computer.

When a computer communicates with another computer on the network, it sends out both the other computers MAC-address and the MAC-address of its own. In that way the receiving computer will not only recognize that this packet is for me but also, who sent this data packet so a return response can be sent to the sender.

2 NETWORK MODEL

ATPG uses the *header space* framework—a geometric model of how packets are processed we described in [16] (and used in [31]). In header space, protocol-specific meanings associated with headers are ignored: A header is viewed as a flat sequence of ones and zeros. A header is a point (and a flow is a region) in the space, where is an upper bound on header length. By using the header space framework, we obtain a unified, vendor-independent, and protocol-agnosticmodelof the network2 that simplifies the packet generation process significantly.

A. Definitions

summarizes the definitions in our model. *Packets:*A packet is defined by a tuple, where

the denotes a packet's position in the network at any time instant; each physicale

3 ATPG SYSTEM

Based on the network model, ATPG generates the minimal number of test packets so that every forwarding rule in the network is exercised and covered by at least one test packet.Whenan error is detected, ATPG uses a fault localization algorithm to determine the failing rules or links.

4 IMPLEMENTATION

We implemented a prototype system to automatically parse router configurations and generate a set of test packets for the network. The code is publicly available.

5 EXISTING SYSTEM

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 . It suffices to find a minimal set of end-to-end packets that traverse each link. However, doing this requires a way of abstracting across device specific configuration files, generating headers and the links they reach, and finally determining a minimum set of test packets (Min-Set-Cover).

To check enforcing consistency between policy and the configuration.

DISADVANTAGE OF EXISTING SYSTEM:

Not designed to identify liveness failures, bugs router hardware or software, or performance problems.

The two most common causes of network failure are hardware failures and software bugs, and that problems manifest themselves both as reach ability failures and throughput/latency degradation.

3 PROPOSED SYSTEM:

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.

It can also be specialized to generate a minimal set of packets that merely test every link for network liveness.

ADVANTAGES OF PROPOSED SYSTEM:

A survey of network operators revealing common failures and root causes.

A test packet generation algorithm.

A fault localization algorithm to isolate faulty devices and rules.

ATPG use cases for functional and performance testing.

Evaluation of a prototype ATPG system using rule sets collected from the Stanford and Internet2 backbones.

4 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 [30]. It suffices to find a minimal set of end-to-end packets that traverse each link.

However, doing this requires a way abstracting across device specific configuration files (e.g., header space), generating headers and the links they reach (e.g., all-pairs reachability), and finally determining a minimum set of test packets (Min-Set-Cover). Even the fundamental problem of automatically generating test packets for efficient liveness testing requires techniques akin to ATPG.

5 REFERENCES

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.

[4] "Beacon," [Online]. Available: http://www.beaconcontroller.net/

[5] 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.

[6] 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.

AUTHOR'S DETAILS:

GUNDEBOINA UPENDER
Pg Scholar , Department Of Cse. Gandhi Academy Of Technical Education, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206.

B NAVEEN KUMAR
Assistant Professor, Department Of Cse. Gandhi Academy Of Technical Education, Ramapuram (Katamommu Gudem), Chilkur(M), Kodad, Telangana 508206