**Automatic Test Packet Generation**

**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 suffices 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 consumes less than 1% of link capacity. ATPG code and the datasets are publicly available.

**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.

**DISADVANTAGES 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 reachability failures and throughput/latency degradation.

**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.

**SYSTEM ARCHITECTURE:**

****

**SYSTEM REQUIREMENTS:**

**HARDWARE REQUIREMENTS:**

* System : Pentium IV 2.4 GHz.
* Hard Disk : 40 GB.
* Floppy Drive : 1.44 Mb.
* Monitor : 15 VGA Colour.
* Mouse : Logitech.
* Ram : 512 Mb.

**SOFTWARE REQUIREMENTS:**

* Operating system : Windows XP/7.
* Coding Language : JAVA/J2EE
* IDE : Netbeans 7.4
* Database : MYSQL

**REFERENCE:**

Hongyi Zeng, Peyman Kazemian,George Varghese,and Nick McKeown,“**Automatic Test Packet Generation**”,VOL. 22, NO. 2, APRIL 2014.