ETALIQ

intelligent quality

Test Automation:

Do More Testing in Less Time

A Case Study
by Etaliq Inc.

www.etaliq.com

Etaliq Inc.
4B–2548 Sheffield Road
Ottawa, ON K1B 3V7

Phone: (613) 241–1385
Fax: (613) 241–1523
http://www.etaliq.com

Previous Configuration Generation System Abandoned.

ETA Selected as Tool Of Choice

Background

The subject is a large telecommunications solutions supplier with hardware and software solutions for all facets of network engineering. They employ over 10,000 engineers. The particular business unit creates, maintains, and supports large carrier-class core and edge nodes.

Executive Summary

Who

- Carrier class network equipment provider
- Over 10,000 engineers

Challenge

- Configure DUT in preparation for manual testing
- Currently takes each engineer 2–6 weeks

Solution

- ETA Client and Server Install
- Develop a flexible table-based configuration suite

Result

- 6 person months meet 99% of requirements
- Provided 80 additional test cases to verify stability
- Became a licenced Etaliq ETA customer
- ROI well beyond expectations

Test lab setups are increasingly complex. A single manual testing Verification Engineer must set up and configure as many as 7 or 8 physical nodes to be used for testing on a per project basis. Several logical configurations are often required, where testing of new features must be accomplished on all classes of hardware, interface, topology and technology. Several of these engineers are assigned to each project.

Depending on the scope of the testing to be performed, each of the engineers are allotted between 3 and 6 weeks to prepare their test lab setup and create their configurations. Each Test Engineer used his own set of one-time-use Tcl/Expect scripts to generate their configurations.

Several attempts had been made to internally create a tool that would assist in reducing the time to generate the configurations. In the most recent example, they committed six person-years of senior Test Automation Engineers who wrote 75 K lines of code (LOCs), mostly in Tcl/Expect. The resulting system required two Senior Development Engineers working full-time to maintain and support it. The time to generate a configuration of intermediate size and complexity was 24 hours. It supported only 75% of the specified requirements. Due to difficulty of use, and speed of operation, the project was abandoned.

Challenge

In total they spent nearly $1,000,000 and over a two years of development engineering time with 3 engineers. It was over 75 K LOCs, using Tcl/Expect; yet, only supported 75% of the predefined requirements. Worse, it required high maintenance and was very fragile, frequently aborting due to minor script errors or subtle environment changes. Use by other engineers to create their own configurations required significant training and often resulted in problem reports to the tool support staff.

Due to the single-resource risk, lack of flexibility, reliability and ease of use, the project was abandoned.

> *"After another attempt at an in-house configuration generation tool using their own Tcl/Expect automation system, costing nearly $1,000,000, they abandoned the project. Manual Test Engineers were still required to create one-time-use scripts to generate configurations at the beginning of each project."*

Requirement

Reduce the human effort required at the beginning of every test cycle to create multiple, complex, scaled lab setup configurations.

They needed a new system. One that was easy enough to use that their individual Manual Test Engineers could define and request their own unique configurations. One that had a minimal learning curve.

At a minimum: The new test tool must fully configure the entire testbed, including the traffic generator and traffic analyzer. As such, it must configure the network's layer 2 controller, and its interfaces and sub-interfaces. It also has to be able to configure the device under test (DUT) to use a wide variety of edge routing protocols and must support all core routing protocols that the device does. It must configure the routing filters to balance traffic or flood traffic, as necessary. The clear objective is to have a tool that can create a configuration in less than 6 hours. It needs to support CE, PE, and P class router configurations, and both IP and Multicast traffic, either in v4 or v6. More detail is provided in at the end in Network Requirements and Traffic Analyzer/Generator Requirements.

As a stretch goal: The new test tool must include the ability to add test cases for these complex configurations. Basic tests for disable/enable, crash/recover, and configure/un-configure, are performed for each project. Creating automation for these would assist the Manual Test Engineer in frequently operating automated tests as regression during the test cycle. An ability to specify that these tests could repeatedly run, for a period of time, to create stress cases, while retaining logged results, would again assist during the test cycle.

> *"The tool must fully configure the entire testbed in a scaled configuration on all platforms and applicable hardware, supporting all classes of interfaces, topologies, and technologies, emulating a core and edge network with* CE, PE, *and P class routers, as well as configure the traffic generator analyzer for traffic and routes, etc."*

Solution

They challenged Etaliq to create such a tool, specifying it as a pre-requisite to becoming a customer. Etaliq with its ETA tool accepted the challenge.

Etaliq installed a laptop demo Linux server into their lab facility. Etaliq provided them with a custom node class to match their DUTs. In two days, the test system was operational; They could begin the development process. It took only five months for a Senior Etaliq Automation Engineer to meet 99% of requirements with only 6 K LOCs. An existing Senior Networking Engineer spent an additional month verifying the generated configurations, to ensure maximum quality.

That expenditure also netted them 80 test cases to verify the stability of the development images to make sure that they would be able to run the generated configurations.

The resulting configuration generation system is very robust and sees frequent use during the development cycle by many Manual Development Test Engineers. The system is very fast, reducing their test setup time by a factor of six. It is also very reliable, and needs only a single Student Test Engineer to maintain it.

Best of all, the configurations are table-driven, making them easy to modify, even for non-automation personnel.

> *"Table driven setups for complex multi-node carrier class network configurations including traffic generator setup, and it is easy."*

Result

The time allotted to each engineer at the beginning of the test cycle is now only one week. Each engineer is creating and saving multiple configurations for their testbeds. They are using configurations with more complexity and scalability than ever before. Some configurations are more than 75,000 interfaces over the entire testbed with mixes of all different interface classes, topologies and technologies.

Internally, it is agreed that each and every time the system is used to generate a configuration, the time savings in an average of 2.5 weeks of engineer time. That puts the ROI at about $6,000 per engineer, per project.

After only a few months of effort, with only one developer, Etaliq provides a tool that makes the creation of complex generated configurations easy, where several internal attempts, with multiple person years of effort, had proven fruitless. *Impressive!*

> *"After only a few months of effort, with only one developer, Etaliq provides a tool that makes the creation of complex generated configurations easy. And it includes 80 basic operational tests for use in stress and scale testing."*

> *"Impressive! Amazing!!! Astounding!!!!! Remarkable!!!!!!!"*

Customer Feedback

At the end of the trial, this company became the second Etaliq ETA licensed client. They provided the following immediate feedback:

Technical Lead:

> *"It is as very valuable tool. We intend on using it throughout various testing projects. It's astounding how it was created in such a short period of time, compared to our own in-house tools, and with very little code. Remarkable!"*

Test Engineer:

> *"It's very simple to learn and use. It reduced my test setup time from 12 days to 2 days, including all hardware installs. It even sets up the traffic analyzer! It's a very big time saver. I guess that's why it's called* ETA *(Easy Test Automation)."*

Appendices

Network Requirements

1. Layer 2 controller and interface/sub-interface configurations required:
   (a) Channelized support for DS0, T1, T3, Sonet (STS)
   (b) Bundled interface support for POS and Ethernet
   (c) Multi-link interface support for Frame-Relay and PPP
   (d) Sub-interface support for Frame-Relay and VLAN
   (e) Main interface support for Serial (HDLC, PPP), FastEthernet, GigEth, TenGigE, POS, Frame-Relay
2. Routing protocol support
   (a) Edge routing support for BGP, OSPF, IS-IS, RIP, and Static

- (b) Create filters such that each unique traffic path gets traffic and flooding routes
- (c) Core support
  - i. Tunnels
  - ii. Any routing protocol
  - iii. MPLS or IP Core

3. Topology support:
   - (a) CE, PE, P, class router configurations
   - (b) VRF: VPNv4, VPNv6, Dual Stack,
with Multi-Physical links possible per VRF
   - (c) L2VPN: Pseudo-wire, Bridge Domains
   - (d) Native: All
   - (e) MPLS or IP Core
4. Configured Traffic support for
   - (a) IPv4, IPv6
   - (b) Mcast v4
   - (c) Mcast v6
   - (d) Igmp/Pim

Traffic Analyzer/Generator Requirements

1. Streams
   - (a) CE to CE
   - (b) Added for all end-to-end paths
   - (c) Configured with correct Total bandwidth for each Path
   - (d) Any combination of IPv4, IPv6, Mcast v4, Mcast v6
2. Routing
   - (a) BGP routes generated for all above noted Traffic Paths
   - (b) Additional BGP Flooding Routes generated for Stress
3. Multicast
   - (a) Sources and Groups for all Mcast Streams
   - (b) PIM or IGMP
   - (c) Joins, Leaves, . . .