#### **PAPER • OPEN ACCESS**

# Manual and automation testing and verification of TEQ [ECI PROPIRETRY]

To cite this article: Ravi Abhichandra and M Jasmine Pemeena Priyadarsini 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 052039

View the article online for updates and enhancements.

## **Related content**

- <u>DEVICES FOR OBSERVATORY</u> <u>AUTOMATION</u> R. K. Honeycutt, B. R. Adams, D. J. Swearingen et al.

 Augmenting SCA project management and automation framework
 M Iyapparaja and Bhanupriya Sharma

- <u>A Write-Once Optical Card Reader/Writer</u> <u>Using Two Optical Sources</u> Shigeru Yoda, Keiichi Tsutsui, Yoshihito Koshiba et al.

## Manual and automation testing and verification of TEQ [ECI PROPIRETRY]

Ravi Abhichandra and M Jasmine Pemeena Priyadarsini

School of Electronics Engineering, VIT University, Vellore, Tamil Nadu 632104, India

E-mail: jasmin@vit.ac.in

**Abstract.** The telecommunication industry has progressed from 1G to 4G and now 5G is gaining prominence. Given the pace of this abrupt transformation, technological obsolescence is becoming a serious issue to deal with. Adding to this fact is that the execution of each technology requires ample investment into network, infrastructure, development etc. As a result, the industry is becoming more dynamic and strategy oriented. It requires professionals who not only can understand technology but also can evaluate the same from a business perspective. The "Information Revolution" and the dramatic advances in telecommunications technology, which has made this possible, currently drive the global economy in large part. As wireless networks become more advanced and far-reaching, we are redefining the notion of connectivity and the possibilities of communications technology. In this paper I test and verify the optical cards and automate this test procedure by using a new in-house technology "TEQ" developed by ECI TELECOM which uses one the optical cards itself to pump traffic of 100gbps.

#### **1. Introduction**

The DLW-NIC has been a scalable solution for verifying and qualifying the performance of today's multi-protocol global communications networks. One of the key features of DLW-NIC is to pump and verify network traffic, and is designed to grow as the network grows, and as technologies change. The NIC products support a variety of test applications for the transport network such as 100G Ethernet, OTN ( up to 112Gbps), SONET/SDH( up to 40Gbps) Next Generation (VCAT, LCAS, GFP), PDH/T-Carrier , Ethernet 10/100/1000 Base T, Gigabit Ethernet, 100 Base FX ,10GigE (LAN, WAN), Fibre Channel 1-10Gbps, ATM. One of the most evident drawbacks of this hardware product is the cost. This product is very expensive. Hence ECI TELECOM has come up with a more cost effective way to verify the performance issues in a communications network with respect to traffic. This developed technology has to be rigorously tested and then automated, before it is carried out in the actual communication network [1-2].

Overcoming the noticeable drawbacks that DLW-NIC holds in the present day communication networks, ECI has come up with a new in-house technology called TEQ *[ECI PROPRIETRY]* which uses one of the optical cards (FIO100) itself to pump traffic of speeds 100gbps. The objective of my project and this paper is to set-up and configure this in-house technology in the R&D Bangalore office, test these optical cards and also automate the test procedure PROPRIETRY.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI. Published under licence by IOP Publishing Ltd 1



Figure 1. The system functional diagram

This paper comprises of seven segments: area II talks about the background of the topic in the previous years. Area III talks about the proposed technique and the work done with a specific end goal to accomplish feasible outcomes. Area IV shows the results at the end. Area V is the conclusion we come to after the testing and verification process [3]. Area VII points out the references.

## 2. Background

The DLW-NIC is a scalable solution for verifying and qualifying the performance of today's multiprotocol global communications networks [4-6]. One of the key features of DLW-NIC is to pump and verify network traffic, and is designed to grow as the network grows, and as technologies change. The NIC products support a variety of test applications for the transport network such as 100G Ethernet , OTN( up to 112Gbps), SONET/SDH( up to 40Gbps) NextGeneration (VCAT, LCAS, GFP), PDH/T-Carrier , Ethernet 10/100/1000 Base T, Gigabit Ethernet, 100 Base FX ,10GigE (LAN, WAN), Fibre Channel 1-10Gbps,ATM.

## 3. Methodology

#### A. SETUP and Installation:

For the hardware setup configuration, we will need the following equipment.

- i. FIO100M optical card which behaves like a TEQ which has a single port, having capacity of 100Gbps
- ii. Optical card- TR100M
- iii. CFP (SR10 or LR4)
- iv. CODEM
- v. Connection cables

#### Installation of optical card- TM100

The optical card is fitted into slot-u4 of the 9608 chassis present in R-2 RACK-09 in the server lab. As the optical card supports 2 types of transceiver, ie client side and line side, OTR100-SR10DR is fitted for the line side and OTR100P2-SR10 is fitted for the client side.

#### **Connections**

For testing the OTU4 traffic, a 100Gbps fiber optic connection is made between the single port of TEQ (FIO100M) and port 0 of TM100. To avoid a LINK-FAIL on the GUI due to open connection, we can either put a physical loopback on port 1 with the help of a 100Gbps loopback optical fiber, or configure a software loopback. During my testing process, I have configured a software local loopback on port 1.

For testing the ETY100G traffic, a similar 100Gbps fiber optic connection is made between the single port of TEQ (FIO100M) and port 1 of TM100. To avoid a LINK-FAIL on the GUI due to open connection, we can either put a physical loopback on port 0 with the help of 100Gbps loopback optical fiber, or configure a software loopback. During my testing process, I have configured a software local loopback on port 0.

#### Configuration of TR100M.

Once the above installation and connections are properly made, we then have to connect to the chassis and configure the optical card to our desired specifications. A remote connection can be made to the chassis by telnet to the management IP of the chassis-172.18.138.94.Entering into the configuration mode will provide the flexibility of entering all our specifications for the testing and verification process [5].

The optical card-TM100 is configured to TR100M for transponder application. After configuration of slot-type comes the configuration of the two ports present in TM100M.

The ports are configured as below:

>	PORT	0	
	0	SLOT:	U4
	0	PORT-TYPE:	OTU4
	0	TRANSCEIVER TYPE:	OTR100-SR10DR
<b>&gt;</b>	PORT	1	
	0	SLOT:	U4
	0	PORT-TYPE:	ETY100G
	0	TRANSCEIVER TYPE:	OTR100P2-SR10

## How to open the GUI.

In order to use the elasteq GUI you need the following:

- User account in the linux system
- Vnc account on linuxcad62

To open the GUI do the following:

- Open vnc-viewer
- Press on the elasteq icon.



Figure 2. Logging into the VNC viewer



Figure 3. After logging into the VNC viewer

**IOP** Publishing

#### 4. Manual testing

#### Starting Elasteq

In order to start elasteq after power, we should initially start the GUI. Then insert the IP number in the TEQ IP number. Next step is to choose our desired TEQ mode (OTU4 or ETY100G) for which we will be doing the testing and verification process. Then press the REBOOT button and wait for the process to be finish. The link-state lamp will turn green and the indicators should show "LINK-OK", "READY" and read "waiting for command…" when it is finished.

#### **Open Elasteq**

For connecting the GUI to an already open elasteq, we should first open and start the GUI. Then insert the IP number in the TEQ IP number. Next press on the "CONNECT" button in the TEQ OPERATION panel and wait for the process to finish. The GUI will update the TEQ mode. The PORT-STATUS lamp will turn green, and the indicators should show "LINK-OK", "READY" and read "waiting for command…" when it is finished.

#### Start Test

Once all the specifications are given, we start the test by clicking on the play button on top of the GUI. This automatically starts the test timer and goes into the statistics tab. We then start pumping traffic from TEQ by pressing the traffic signal button also on top of the GUI. After pressing this button, there will be a delay of few seconds before we start to see the frames sent counter incrementing.

#### **Testing Phases**

🗙 💿 Packet Generator and Analyzer version 12 - elstq15	
🛩 🖬 🕨 🗄 💭 🗮 📫 🗙	00:00:31 💭 🔴
Hardware Flows Configuration Faults Errors Statistics Capture Log Test	
TEQ Configuration	
TEQ IP Number 172.18.139.122	
TEQ Mode: C 100GbE • OTU4 C 10GbE C OTU2 C OTU2EP C OTU2E	
OTU FEC: C None • G709	
TEQ Operation:	
Connect Restart Optics Reboot	
TEO Locar	
Close Laser	
TEQ Automation	
Host elasteqInx	
Port 7125	
Maintenance (Dev Only)	
Statistics source: Mac • Analyzer	
TEQ Self test:    None Serdes Elasteq (Dev only) Through	
Waiting for command Analyzer Stat No Self-Test Err-Fai	Link-Fail Ready

Figure 4. Traffic testing phase 1

## **IOP** Publishing

IOP Conf. Series: Materials Science and Engineering **263** (2017) 052039 doi:10.1088/1757-899X/263/5/052039

X 🕤	Packet Generator	and Analyzer	version 12 - els	tq15				×
🛩 🖬	▶ 8 🖵 =	K Ci I				00:00	:38	00
Hardw	are Flows Configu	ration Faults	Errors Statistics	Capture Log Test				
	TEQ Configurat	tion						
	TEQ IP Number		172.18.139.122	[				
	TEQ Mode:	100GbE • C	TU4 0 10GbE	O OTU2 O OTU2EP	O OTU2E			
	OTU FEC:	None • G	109					
	TEO Constitue							
	TEQ Operation:							
	Connect	Restart	optics R	eboot				
	TEQ Laser:							
	Close Laser			R.				
	TEQ Automatio	n		,				
	Host	elaste	qlnx					
	Port	7125						
	– Maintenance (D	ev Only)						
	mannenance (D	ev enny)						
	Statistics source	: 🔿 Mac	<ul> <li>Analyzer</li> </ul>					
	TEQ Self test:	<ul> <li>None</li> </ul>	Serdes	<ul> <li>Elasteq (Dev only)</li> </ul>	<ul> <li>Through</li> </ul>			
	Adaptation for po	ort: all	Start					
Waiting	g for command			Analyzer Stat No Self	-Test No-Err	Link-Ok	Rea	dy

**Figure 5.** Traffic testing phase 2

		00:00:02 🖸 🤅
lardware Eleves Cor	figuration Faults From Statistics Conturn Log Tort	•
ardware   Flows   Col	inguration Paules Errors Statistics Capture Eog Test	
	Name	Value
<u>~</u>	Elapsed time	00:00:02
<b>O</b>	Frames sent	0
	Frames received	0
	Good frames received	0
	Bad frames received	0
	Frames with CRC err received	0
	Frames with Ext err received	0
	Undersized frames received	0
	Oversized frames received	0
	Frames with sq num err received	0
	Bytes sent	0
	Bytes received	0
	Good bytes received	0
	Bad bytes received	0
	Bytes with CRC err received	0
	Bytes with Ext err received	0
	Average frame length received	0
	Average delay	0
	Minimal delay	0
	Maximal delay	0
	Snd Throughput	0
	Rcv Throughput	0
	Disruption time	0
ank hanken akud	Analyzor Stat No Solf 7	

Figure 6. Traffic testing phase 3

ardware Flowe Con	Figuration Faulte From Statistics Conture	og Test	-
aluwale   hows   con	inguration Faults Errors statistics capture i	Log   Test	
	Name	Value	
<u> </u>	Elapsed time	00:05:07	
<u> </u>	Frames sent	310,912,191	
_	Frames received	384,197,749	
R	Good frames received	384,197,749	
	Bad frames received	0	
	Frames with CRC err received	0	
	Frames with Ext err received	0	
	Undersized frames received	0	
	Oversized frames received	0	
	Frames with sq num err received	0	
	Bytes sent	2,984,757,033,600	
	Bytes received	3,716,130,312,132	
	Good bytes received	3,716,130,312,132	
	Bad bytes received	0	
	Bytes with CRC err received	0	
	Bytes with Ext err received	0	
	Average frame length received	9,672	
	Average delay	33.74 usec	
	Minimal delay	33.39 usec	
	Maximal delay	33.83 usec	
	Snd Throughput	78.81 Gbps	
	Rcv Throughput	98.12 Gbps	
	Disruption time	0	

Figure 7. Traffic testing phase 4

## **Error and Fault Testing**

The verification process can be done by injecting known errors and faults into the traffic flow and comparing it with the results shown in the GUI. Different traffic type, support different types of errors and faults. For eg, OTU4 traffic is capable of injecting different types of faults such as OUT\_LOF, OUT\_TIM, ODU\_LCK and errors such as OUT\_BEI, OUT-BIP. Similarly, while testing ETY100G, we can inject different types of faults such as GBE\_LF, GBE\_RF.

Once the test is started, and the traffic is flowing, we can inject any type of fault/error by going to the FAULTS or ERRORS tab on top of the GUI and clicking on the desired entity. After a seconds of delay, we can see the indicators reporting the error and faults.

## **Testing Phases**

	Name	Malua	
~	Name	value	
	Elapsed time	00:06:59	
	Frames sent	487,397,940	
<b>20</b>	Frames received	487,397,940	
n	Good frames received	487,597,940	
	Bau names received	0	
	Frames with Cit en received	0	
	Lindersized frames received	0	
	Oversized frames received	0	
	Frames with so num err received	0	
	Butes cent	4 679 020 224 000	
	Bytes received	4,679,020,224,000	
	Good bytes received	4 679 020 224 000	
	Bad bytes received	4,075,010,114,000	
	Butes with CBC err received	0	
	Bytes with Ext err received	0	
	Average frame length received	9.600	
	Average delay	33.74 usec	
	Minimal delay	33.61 usec	
	Maximal delay	33.83 usec	
	Snd Throughput	89.34 Gbps	
	Rcv Throughput	89.34 Gbps	
	Disruption time	0	

Figure 8. Error testing phase 1



Figure 9. Error testing phase 2

rdware	Flows Configuration Fa	ults Errors S	tatistics Capture Log Test	
anare	i nono   comgaradon	Terroro   c	addaed   captare   cog   rest	
~	Fault Name	History	Current	
	OPT_FAULT_LOS	0	0	
	OTU FAULT LOS	0	0	
~7	OTU FAULT LOF	0	0	
3	OTU_FAULT_OOF	0	0	
-	OTU_FAULT_LOL	0	0	
	OTU_FAULT_AIS	0	0	
	OTU_FAULT_LOMF	0	0	
	OTU FAULT TIM	1	1	
	OTU FAULT BDI	1	0	
	OTU FAULT IAE	0	0	
	OTU_FAULT_BIAE	1	0	
	ODU_FAULT_AIS	1	1	
	ODU_FAULT_OCI	0	0	
	ODU_FAULT_LCK	0	0	
	ODU_FAULT_TIM	0	0	
	ODU_FAULT_BDI	1	0	
	ODU_FAULT_MSIM	0	0	
	ODU_FAULT_PLM	0	0	
	ODU_FAULT_CSF	0	0	

Figure 10. Error testing phase 3

## ElasTeq Automation Block Diagram

The Elasteq automation is done be connecting to a telnet port. The initial setup must be done using the Elasteq GUI, and then and outside scripts can be used to connect to the telnet port. Through this port you can inject commands and receive the output

#### **Automation Flow Diagram**

After successfully completing the manual testing process, I then have made the whole testing and verification of the optical card using TEQ into automation. Due to confidentiality concerns, I am not able to disclose the actual code which I have written for the automation process. I am able to show the automation flow diagram of how the event takes places and the logic used.



Figure 11. Automation Block diagram



Figure 12. Automation flow diagram

The flow starts at the Event Proc which is for the Feature test-type. Then the input which is the Marray, goes into the Basic card Event. There are two different events performed at the Basic card Event which are: Test card assignment and the Test port assignment (event performed in all ports with all port-type) scenarios. Then in the For Each Port Type, a switch is executed for each test-event with randomly generated port-number. The switch-cases are given as below:

Case 1: Port-Protection; Input: mArry; key used in array: prtProt.

Flow: If (\$CardArr (prtProt) is true then port-protection is supported for corresponding card. Configure same random port on slot u2 and then configure port-Protection to test +VE. Configure different port on u2 than that of port configured in slot u0 and then test -VE scenario. If applicable configure Y-Protection and verify them too.

Case 2: PRBS31; Input: mArry; key used in array: fabType;

Flow: If (\$CardArr (fabType)) is FAB or NON-FAB cards, then configure value PRBS31 to attribute pattern-tx/rx. On all port-type. Verify the changes in implicit configuration for attribute pay-loadexpected/sent. For EDFA & PASSIVE card we can't configure PRBS31!

**Case 3:** GCC; Input: mArry; key used in array: linePty, clntPty; Flow: If (\$portType) is OTU\* then Perform GCC-operation accordingly else configure only loopback. And verify them accordingly.

**Case 4:** XC; Input: mArry; Key used in array: xcInternal or xcExternal;

Flow: If (xcExternal = explicit) 1) configure slot u2 and same port with same port-type.2) Configure slot u2 with port-type OTU\*(do de-multiplexing if required). And configure XC in above scenario and verify them. If (xcInternal = implicit) Verify the implicit configuration. If (xcInternal = explicit) Configure the line port with OTU\* (do de-multiplexing if required). Configure XC between line port& Client port and verify them too.

**Case 5:** Configure RO and RW attribute and verify them. If (\$port-Type = OTU\*) then de-multiplex it and verify its odu container for RO and RW attribute changes also.

Verify the de-multiplexed interface in case of FAB and NON-FAB & PASSIVE card.

Here we create expected-string for both implicit & RW-RO atttibute changes configuration expectedstring for each port-type. Its applicable to demultiplexed odu in both line & client port.

## 5. Results

The testing process is successfully tested with the help of the GUI. Once the connection between the TEQ and the optical card is maintained and traffic flow is correct, different errors are injected into the optical card. Proper verification is done with the injected errors. Finally, the whole process is automated so that the whole process can be without any human intervention and is resistance to any delays or human-mistakes.

## 6. Conclusion

The proposed in-house technology developed by ECI TELECOM is the product of the most innovative and diligent minds coming together in order to tackle on the most glaring problems in present times. DLW-NIC has been a scalable solution for verifying and qualifying the protocols of modern communication network, but one the most glaring drawbacks of this product is the cost of the product. This hardware is very expensive. Therefore, ECI TELECOM has come up with the solution of a cost effective way of verifying the performance issues in the communication network with respect to traffic.

#### Acknowledgments

We would like to express gratitude toward Dr. Renuga Devi for her support and inspiration for this project. We would like to extend our appreciation to our guide Dr. M. Jasmine Peemena Privadarsini for understanding our fantasies and supporting us in this project from the beginning. We are fortunate to be a part supporting families and companions without whom this review would not have been finished successfully.

#### References

[1] Jorge López Vizcaíno, Yabin Ye, Tamara Jiménez, 2016 OTN switching in protected transport networks: Spectral, cost and energy efficiency evaluation *Proc. Of IEEE Conference on Indian control on Communication (ICC)*.

[2] Arley Salvador, Diego Carvalho, Cleber Nakandakare, Eduardo Mobilon, Júlio C. de Oliveira, Dalton S. Arantes 2014 100Gbit/s FEC for OTN protocol *Proc. Of IEEE Design architecture and implementation results*" *in Telecommunication Symposium (ITS)*.

[3] Serge Melle, Abhijeet Deore, Onur Turkcu, Satyajeet Ahuja, Steven J. Hand 2013 Comparing optical & OTN switching architectures in next-gen 100Gb/s networks Proc. Of *Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC), IEEE*, March, 2013.

[4] ITU-T G.798 (2006), *Characteristics of optical transport network hierarchy equipment functional blocks*.

[5] ITU-T G.872 (2001), Architecture of optical transport networks. Hezi Eliyahu and Oren Mayer, "TEQ [ECI PROPRIETRY]-Test Equipment Specifications", November 7,2016

[6] Hezi Eliyahu, "TEQ [*ECI PROPRIETRY*] "Automation-test Equipment Automation Specifications", July 26,2016.