Advanced Services Development Team - Xena Networks - Product Evaluation Report

Introduction

There has been keen interest in the R&E community for a network test tool that provides the ability to accurately characterize the performance a 10 Gigabit Ethernet (10 GE) network at a reasonable price point. MCNC had the opportunity to perform a product evaluation of Xena Networks' compact series network testers, which, based on the company's marketing material, appear to potentially meet that criteria. The objective of this report is to evaluate those claims and provide an appropriate report to Internet2 and the US R&E community.

Background

The availability of compatible network performance testers when testing high-speed inter-institutional network links is rare and generally cost prohibitive; this limitation often leads to the use of commodity systems and open source tools, of which the reliability and reporting leaves some room for improvement. Working jointly with Internet2 and DANTE, we were fortunate to be able to evaluate network test equipment from Xena Networks as a potential low-cost, high-performance alternative solution, as compared to devices provided by industry leaders for testing and certifying high-speed network infrastructure. We received several units; one in a live configuration, in support of an interoperability test of 10 GE equipment which interconnected disparate vendor devices across a point-to-point Ethernet over SONET circuit slated to carry production traffic, and another in a lab environment, in a back-to-back, single loop configuration. Our peers at DANTE were performing an evaluation of the same product concurrently, with a similar purpose.

The test units were introduced during an already in-progress interoperability test, which was being executed using commodity systems and standard open source test tools; as a result, not all features were able to be tested, based on the limited availability of compatible network equipment in the local test environment and the limited scope and timeline of interoperability testing.

Specifics of units under evaluation

All units involved in testing were XenaCompact series M2SFP+. The units were small form-factor (1U, half-depth) rackmount chassis, and included a single standard (IEC C-14) AC power connection, a single management network interface, and an option card equipped with two SFP+ transceivers; the testers used in the live interoperability test were equipped with MergeOptics TRX10GDP0310 long range (1310nm) transceivers, while the bench units were equipped with MergeOptics TRX10GVP2010 short range (850nm) transceivers.

In both cases, the units came installed with the current server software release, and a companion CD that included all available documentation, current clients and server software. Throughout the interoperability portion of the testing, the units were tested using the 68,18 of the Xena server release (though note the section below regarding release updates).

evaluation areas covered

Basic hardware setup and software (GUI client) control of the test set

The unit for interoperability testing was received pre-configured by the vendor with a management address (coordinated before delivery), and we were able to connect it to the network without issue. Once the management interface was properly connected, we were able to verify proper connection of the test interfaces via the Xena Manager GUI. It should be noted that the physical installation was being done 'blind', that is, one engineer performed the physical installation in NY, while another verified the working state remotely in NC.

The bench unit arrived configured with the default RFC1918 management address; this was relatively simple to change, by following the steps outlined in the very brief, but straightforward 'Getting Started' guide included with the documentation. The basic procedure involved connecting to the management port using an Ethernet crossover cable, connecting to the chassis using the Xena Manager client, and reconfiguring the chassis as desired (e.g. changing the management network configuration, name, description, password)

On both units, the test ports were not clearly labeled; regardless, it was relatively trivial to identify each port through trial and error. It would have been useful to either label the ports physically, document a procedure for identifying the ports, or provide a means of identifying the ports dynamically, perhaps in the same manner that chassis identification can be achieved through flashing the LED's.

Software installation

The Xena chassis are capable of being managed using both GUI and scripted interfaces. The primary method of controlling the chassis is via the Xena Manager GUI client, which runs under Microsoft Windows. The client itself has quite a small footprint (289KB), and no installation was required per se, other than to download it from the vendor's web site, or retrieve it from the provided companion CD. Making the initial connection to a chassis involved a one-time operation of creating a new testbed, adding a new remote chassis, and providing the IP address and password for the desired remote target. This configuration information was saved across manager restarts, and reconnecting to the chassis was achieved either by restarting the Xena Manager client, or manually reconnecting to the chassis within the client. Current chassis configuration and present test state (if any) was automatically discovered upon connection. A single Manager instance can be connected to multiple chassis simultaneously. The chassis also support multiple simultaneous connections, allowing multiple clients (GUI or scripted) to connect and control the chassis, inspect an in-progress test, etc. Resources can be reserved at the port, module or chassis level in order to prevent test collisions; reservations can be relinquished by the initiating tester or any other user logging in to the chassis.

The Manager provides access to all features of the Xena equipment, including power control, test stream profile creation, packet filtering, capture, and test statistics/histograms. Initiating a simple test stream was a matter of following this basic recipe:

- reserve ports on each target chassis
- create, configure and enable a test stream on a port at one 'end' of the test path
- initiate the stream, either for a fixed packet count, or infinite

It was also possible after following this recipe to save the port configurations and later restore them, on a port-by-port basis; this feature was useful for repeated tests or in situations where multiple administrators were using the chassis for different test scenarios.

After some initial setup, we encountered a few non-intuitive options that needed to be explicitly set before some of the statistics would be generated, but with some brief guidance from the vendor, were able to run some test streams with meaningful results.

The other option for controlling a remote chassis is through a scripted command interface, which provides access to all but a few options provided by the GUI Manager. The interface is accessed by connecting to a chassis on `TCP:22611` using a variety of network scripting interfaces (e.g. telnet, expect, Tcl) including through a provided sample client. Aside from the few known limitations of this interface vs. the Manager, and once the command syntax was better understood, running sets of repeated and/or timed tests with some parameter adjustment using the scripted interface was preferable to using the GUI Manager. In particular, capturing intermediate values during a test run was only practical using this interface, as well as anything that required automation. For the majority of our interoperability testing, we relied on the scripting interface, using `perl`'s `Net::Telnet` module. See below for more detail on the scripting interface.

Software/Firmware upgrade

The software upgrade process was a relatively simple operation. We received an updated release archive from Xena, and were able to complete the upgrade using the provided instructions. There are several components within the chassis that can potentially be upgraded; the chassis firmware, the chassis server software, and the xenadriver. The upgrade process involved uploading and applying required components, and then powering the test unit off/on. The updated release also included a corresponding update for the Xena Manager; because of feature changes and potential incompatibilities introduced in the firmware/server releases, Xena requires that Manager be run only against a particular corresponding release. Once the upgrade was complete, we were able to validate a successful update by running the corresponding release of the Manager client and checking the release versions. As an added visual cue, when running a Manager client against a mismatched server version, the chassis version number was highlighted in red to indicate an incompatibility.

One unexpected issue was encountered during the upgrade process; all of the configuration detail retained in the Manager about all connected chassis, which collectively is referred to as a 'testbed', was not preserved post-upgrade; in order to reconnect, that information would need to be manually re-entered for each target chassis. Fortunately, there is a feature for exporting/importing the testbed configuration between upgrades; using this feature, it was possible post-upgrade to launch the older version of the Manager, export the testbed configuration, and then import it into the current Manager release.

One other item that seemed relatively confusing was the firmware/server/manager version numbering scheme. For example, the version numbers for the various components from two different point-in-time releases were as follows:

release	firmware	server	xenadriver	Xena Manager
Feb 2009	18	68	n/a	82
Mar 2009	28	74	20	92

Both releases were provided as complete archives, with all subsequent required component versions included, as well as release notes that included upgrade instructions that reference the relevant versions. Regardless, given that the chassis sub-components and Manager client need to be upgraded in lock-step, it might be more intuitive if these release numbers were somehow aligned; in a larger multi-chassis installation, this version numbering scheme has the potential to introduce version mismatches.

Packet and stream construction options

Stream creation was fairly straightforward. Transmission profiles created through the GUI allowed rate control by specifying a relative percent of the available circuit bandwidth, packets per second, or Megabits per second; specifying one of those parameters automatically adjusted the others. Additionally, several packet parameters could be adjusted, including packet and header lengths, as well as control over fixed/variable packet lengths. Options for modifying the header and payload contents were available, however since our testing was focused on some very specific throughput characterization, we did not have an opportunity to fully exercise that feature. The creation of multiple streams was supported; per-stream statistics were possible by tagging a stream with a test payload id (`TID`).

Frame size sweeping, jumbo frame support

Defining variable packet length within a stream definition was supported. The variation modes include Incrementing, Butterfly, and Random. The incrementing mode performed simple step increments between a defined Min/Max range, adding 1 byte to the previous increment until the Max limit was reached. Butterfly and Random modes produced variable frame sizes in either a sinusoidal or pseudo-random pattern. An incrementing or variable load feature, similar to the variable packet size feature, would have been useful for generating increasingly larger packet streams to get a rough baseline of the zero-loss throughput rate, or else to simulate client throughput variability. In addition, a predefined RFC 2544 test suite would have been useful; for the interoperability testing we were able to achieve this through scripting. Based on feedback to the vendor, RFC 2544 testing will be implemented in a forthcoming release. Jumbo frames (9000 bytes) were fully supported in our interop testing; the tester apparently supports packet lengths up to 16383 bytes, though we had no gear on which to test frames of this size.

Burst profile options

It was possible to define a stream that contained 'bursty' traffic; that is, rather than sending packets in a continuous stream, sending them in some pre-determined inter-packet spacing, through tuning per-burst packet count and burst density. The bursting count was limited to a range of 0-10 packets, though burst density was tunable as a percentage, where full density meant that all burst packets were delivered back-to-back with no inter-packet spacing, and lower percentages would distribute the packet spacing at increasingly even rates. For the interoperability testing, we performed a baseline burst profile test, and then repeated the test using two 50% bandwidth concurrent streams, one with a profile using the 'normal' evenly-distributed packet spacing, and another with a full-density burst profile, in order to simulate the interaction between multiple customer flows. Constructing such a stream using our commodity systems was not practical, therefore this was a useful feature.

Multiple flow testing, including per-VLAN flows

Testing specific flows could be achieved either by modifying the packet content, which involves adding specific field modifiers to a stream, or by applying filters to a given stream, which could then be applied as a rule to trigger a packet capture, or simply to report statistics for a given flow within a stream. Filtering was possible on one of multiple variables, including Ethernet/IP address, VLAN id, type of service field; as the devices involved in the interoperability test were intended to be used to provide circuit services to customers who might perform multiple levels of VLAN tagging (e.g. Q-in-Q), we made use of the stream filter feature in order to verify throughput based on several tagging scenarios.

Layer 2-7 testing, IPv4/v6 support

The test units in this case were used purely for Layer 2 testing, in a non-shared point-point Ethernet configuration. The test unit does support testing on a Layer 3 network as well (i.e. for testing in a routed environment), though we had no such configuration in our production or lab testing. Integration with Layer 4+ testers is not supported directly, but instead must be achieved through physical test network design, by adding simulated or real L4-7 application streams into the test path, some level of application performance can be observed. We did not have the opportunity to evaluate IPv6 support, though Xena claims to provide v6 support in stream header manipulation, packet filtering and capture.

Throughput testing

Using the Xena test units for interoperability testing, we were able to perform sets of tests to validate the throughput and loss rates that produced consistent and reliable results. During one phase of testing, we observed an asymmetric throughput behavior under certain test conditions; in order to test that the behavior was not an artifact of the tester or a passing anomaly, we ran tests using identical parameters in two independent lab configurations. In each case, the lab results were statistically identical, and we were able to drive the testers at near full wire speed, which validated our confidence in the results of the field test and led us to further investigate the anomaly. Having the capability of performing repeated tests with a high degree of confidence also allowed us to quickly narrow the scope of testing until we were able to identify the source of the observed results.

Latency testing and clock synchronization options

Many latency and throughput testing models rely on high-resolution clock sources for accuracy when tagging/identifying test payloads. The Xena tester does not currently support an external clock source, though it does support latency measurement within a single chassis. A mechanism for calibrating a chassis in order to compensate for any internal latency is also provided. This does allow latency measurement to be achieved when testing in a port->DUT->port configuration within a single chassis. Support for external clock synchronization via GPS is a feature on the product roadmap, which will allow latency testing across multiple chassis.

Packet capture capabilities

Packet capture is supported, but was not evaluated.

Scripting options

As outlined above, scripting is supported using standard scripting interfaces capable of issuing network commands (perl, expect, Tcl, Java, etc.), as well as through a provided sample client. The scripting language is documented in the Xena Scripting Specification, which is available for public download. The scripting commands are fairly straightforward, and for our evaluation and testing, we were able to operate without the need to refer to the specification document. An online help facility was also available, which appeared to be fairly complete and self-explanatory. Using this, we were able to construct a few sample scripts, as well as a complete test script that calculated a zero-loss throughput rate at various packet lengths.

Using the sample client, it was also possible to load small code snippets and receive output; for example, after connecting to a chassis, the following code could be used to logon, set an owner, reserve all ports, then reset all statistics:

xena_logon

```
; log on to xena_chassis (192.168.15.100)
;
c_logon "super_secret_pass"
c_owner "cli_test_user"

; reserve ports 0/0, 0,1
0/* p_reservation reserve

; reset ports
0/* p_reset
; clear all stats
0/* pt_clear
0/* pr_clear
```

To clear all statistics, reset all ports, release all reservations, logoff and disconnect, the following snippet could be used:

xena_logoff

```
; clean up, logoff
;
; clear all stats
0/* pt_clear
0/* pr_clear
; reset ports
0/* p_reset

; release ports
0/* p_reservation release

; logout, disconnect
c_logoff
c_disconnect
```

These code snippets could be loaded through the sample client by opening and loading an external file, cutting and pasting, or inputting the commands directly in the client. The scripting client could only be connected to a single chassis at once, which provided only limited utility in a multi-chassis scenario; as outlined above, using the scripting interface vs. the Manager was essentially required in order to achieve some specific test reporting. Fortunately, the same commands used in the scripting client could be used in an external scripting interface, so using the scripting client for prototyping and performing ad-hoc chassis manipulation was quite useful. Providing some sample scripts such as the above or in one or more scripting languages with the client would have reduced the time to learn the scripting language. Providing a client that is capable of managing multiple chassis for coordinated testing might also be useful.

Statistics and test reporting options

The Manager client provided several options for presenting test stream statistics. Gradual updates of in-progress tests, and summary values for completed tests were available via the statistics pane; these were broken down by individual port, stream and test payload ID (`TID`). Completed results from multiple test streams could be retained for comparison. Statistics included relevant traffic counters (Mbits/sec, packets/sec, Byte and packet counters), as well as several error counters (lost packet count, sequence / misorder / integrity errors) and min/avg/max latency measurements. The summary results from completed tests could also be exported to CSV. Some limited graphing of TX/RX inter-frame gap or packet length statistics was supported, though this feature was not fully explored. Support for capturing test detail directly into a Microsoft Excel document via the Visual Basic for Applications API is slated for an upcoming release.

We found that finer-grained control of retrieving mid-test statistics was better achieved using the scripting interface; this provided the flexibility to query and capture a wider range of metrics, at relatively high intervals, with output to whatever format we were capable of scripting. For the most part, we limited our tests to outputting CSV suitable for import and graphing via Excel.

Overall impressions

Overall, the test units performed as advertised. We encountered no problems with hardware or software failures, and all units produced consistent results throughout the evaluation and testing. The compact series tester was simple to place, set up and configure; the small form-factor of this unit lent itself well to quick placement and setup in a remote network exchange. Shortly after placing one unit for the interoperability test, we were able to connect to and manage the unit, create a test stream, and perform some simple tests without much issue or need to consult the documentation in-depth. Some of the features in the provided client tools were not completely intuitive at first, but were relatively simple to understand after consulting either the documentation or receiving instruction from the vendor. More comprehensive documentation and scripting examples would have been useful, particularly regarding some of the more advanced features. We did find a few features to be lacking, such as integrated RFC2544 testing, the ability to effect change on multiple streams/ports/devices, or a few hard limits within the testing parameters, but our impression based on our longer-term experience is that the test units and provided client tools are stable and suitable for performing network performance baselining as delivered. Our expectation is that the product will continue to mature at a healthy pace. We were not able to thoroughly exercise and examine all features of the test units, based on limited availability of compatible network devices with which to test and the limited time frame of the evaluation.

As a company, we found Xena Networks easy to work with, both in their willingness to work with us on the product evaluation as well as in their flexibility and responsiveness throughout the evaluation and testing. As the testing and evaluation progressed, we were able to engage them for guidance on usage of the test units as well as provide feedback on issues encountered during the testing. Our primary interactions were through regular status calls, though both the sales and engineering representatives were available via email and instant messenger as well; in all cases, the level of responsiveness and technical expertise was superb. Working in collaboration with our peers from DANTE, we provided feedback on a number of specific issues as well as suggestions for improvement; in several instances, the vendor received our feedback and implemented fixes or changes in relatively short order, or scheduled the improvements for incorporation in advance releases. Specifically, several features of interest for which we provided feedback are slated for releases beyond the versions tested, including global chassis control (specifically providing the ability to stop/start all streams from one location), automated RFC2544 testing, and more granular reporting (through the Microsoft Excel VBA interface).

Summary

The Xena Networks compact series network testers are quite suitable for network performance characterization and baselining, and based on advertised pricing, as a viable and potentially attractive alternative to more costly and elaborate network testers. For the purposes of our interoperability testing, where a very specific set of features was required, the Xena testers were easily superior as compared to using dedicated commodity hardware, both in terms of performance, consistency, and features, and helped us to achieve and complete a successful set of targeted tests in a more efficient and timely fashion than had been achieved previously. The units produced repeatably consistent measurements, which increased confidence in the observed results. The product was not completely issue-free, and is still maturing, but the responsiveness from the vendor to our issues and suggestions for improvement was reassuring and balanced out any identified concerns. More information about Xena Networks can be found on their website - www.xenanetworks.com.

Add Comment

Added by Tom Throckmorton , last edited by Tom Throckmorton on Jun 24, 2009 08:55

Labels: (None) EDIT