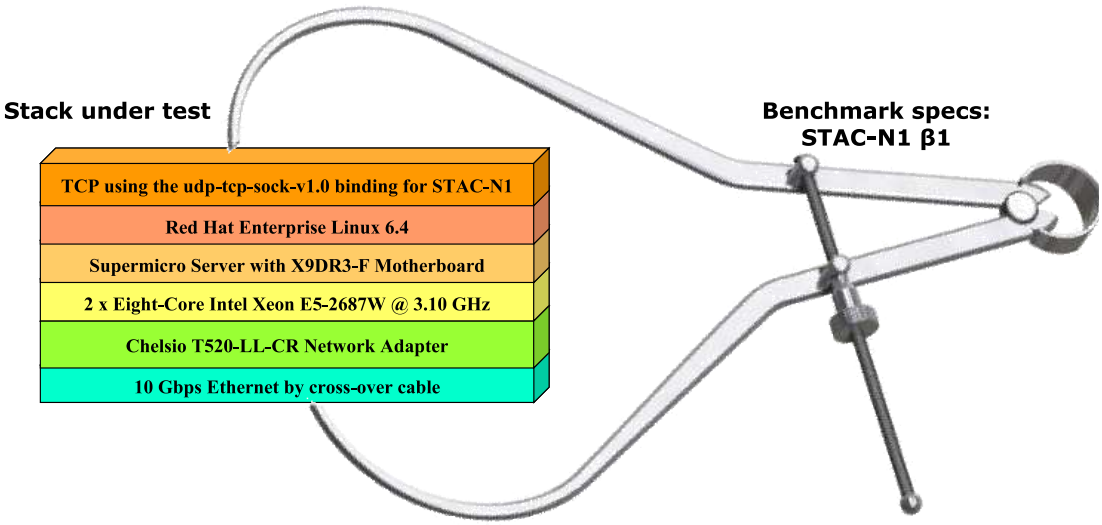
TCP over 10GbE using Chelsio WireDirect TOE on RHEL 6.4 with Chelsio T520-LL-CR Adapters on Supermicro Servers
(SUT ID: CHE130914)

STAC-N1™ Benchmarks β1

Tested by: STAC
Test date: 14 Sept 2013

Report v1.0.0, 27 Sept 2013



STAC BENCHMARK COUNCIL

THESE TESTS FOLLOWED STAC BENCHMARK SPECIFICATIONS PROPOSED OR APPROVED BY THE STAC BENCHMARK COUNCIL (SEE WWW.STACRESEARCH.COM). BE SURE TO CHECK THE VERSION OF ANY SPECIFICATION USED IN A REPORT. DIFFERENT VERSIONS MAY NOT YIELD RESULTS THAT CAN BE COMPARED TO ONE ANOTHER.

STAC® SECURITIES TECHNOLOGY ANALYSIS CENTER

Disclaimer

The Securities Technology Analysis Center, LLC (STAC®) prepared this report at the request of Chelsio Communications . It is provided for your internal use only and may not be redistributed, retransmitted, or published in any form without the prior written consent of STAC. All trademarks in this document belong to their respective owners.

The test results contained in this report are made available for informational purposes only. STAC does not guarantee similar performance results. All information contained herein is provided on an "AS–IS" BASIS WITHOUT WARRANTY OF ANY KIND. STAC has made commercially reasonable efforts to adhere to published test procedures and otherwise ensure the accuracy of the contents of this document, but the document may contain errors. STAC explicitly disclaims any liability whatsoever for any errors or otherwise.

The evaluations described in this document were conducted under controlled laboratory conditions. Obtaining repeatable, measurable performance results requires a controlled environment with specific hardware, software, network, and configuration in an isolated system. Adjusting any single element may yield different results. Additionally, test results at the component level may not be indicative of system level performance, or vice versa. Each organization has unique requirements and therefore may find this information insufficient for its needs.

Customers interested in analyzing their own environment using the same methodology used in this report are encouraged to contact STAC or visit www.STACresearch.com/harnesses.

Contents

Summary

STAC-N1 uses the STAC-N toolset to test a network stack using a market data style workload. STAC-N1 is designed to provide insightful network benchmarks that are neutral with respect to vendor, network API, and network transport. STAC-N1 can test many different combinations of network API, I/O mode, network stack, operating system, and system hardware. This report documents the results from running STAC-N1 tests on a "stack under test" (SUT) consisting of TCP over 10GbE using Chelsio WireDirect UDP with Chelsio T520-LL-CR adapters on Supermicro overclocked servers running RHEL 6.4.

The key results are tabulated in the STAC Report Card on the following page, and Section 8 provides a detailed analysis of latency and throughput. Of these results, Chelsio wishes to highlight the following:

For all tested message rates from 100K to 1.4M msgs/sec, the mean latency did not exceed 3.9 microseconds, and 99.9th percentile latency did not exceed 16 microseconds.

Section 1 contains an overview of STAC-N1, while Section 6 contains details on the STAC-N1 binding used in this report. Section 7 contains a high-level overview of the SUT configuration. Details on the configuration are available in the STAC Vault to qualified members of the STAC Benchmark Council at http://www.stacresearch.com/node/15253 . This includes a STAC Configuration Disclosure and may also include product-specific diagnostic files (e.g., Red Hat sosreport).

Note that because STAC-N1 is not tied to a particular network API, it can be used to compare stacks using different APIs (for example, UDP/Ethernet vs RDMA/Infiniband). However, STAC-N1 is often used to compare different stacks using the same API (for example, UDP with one vendor's NIC and driver vs UDP with another vendor's NIC and driver). When making the latter type of comparison, it is essential that the SUTs you are comparing used the same STAC-N1 binding. It is also essential that they used the same functions in the API. For example, there are several I/O modes for sockets (writev, sendrec, epoll, etc.) as well as for RDMA (read, write, sendrec). If you want to compare network stacks, you should make sure that the SUTs used the same I/O mode. By the same logic, if your goal is to compare different APIs using the same platform, it is essential that you make sure the SUTs you're comparing used the same NIC, OS, server, etc.

Note that each benchmark has a unique identifier. If you are comparing these results to other STAC-N1 Benchmark results, make sure the identifiers match exactly. If they do not, they cannot be fairly compared.

These benchmark specifications and supporting tools are under the guidance of the STAC Network I/O SIG, a sub-group of the STAC Benchmark Council. To participate in this group, please see www.STACresearch.com/nio.

STAC-N REPORT CARD
CHE130914

Spec ID	Description	VALUE	MEAN	MEDIAN	99P	MAX	STDV
STAC.N1.β1.PINGPONG.LAT1	SupplyToReceive Latency (Hybrid) at base rate in the 1:1 setup (µsec)		**3.5**	**3**	**4**	**63**	**0.2**
STAC.N1.β1.PINGPONG.CPU1	Consumer CPU utilization at base rate in the 1:1 setup (core equivalents)	**Active Cores = 3**	**3.00**			**3.00**	
STAC.N1.β1.PINGPONG.CMEM1	Max memory used by the Consumer application process, minus the memory allocated by the STAC Library, at base rate in the 1:1 setup (MB)	**1**					
STAC.N1.β1.PINGPONG.CMEM2	Max memory used by the Consumer application process at base rate in the 1:1 setup (MB)	**356**					
STAC.N1.β1.PINGPONG.CPU2	Producer CPU utilization at base rate in the 1:1 setup (core equivalents)	**Active Cores = 3**	**3.00**			**3.00**	
STAC.N1.β1.PINGPONG.PMEM1	Max memory used by the Producer application process, minus the memory allocated by the STAC Library, at base rate in the 1:1 setup (MB)	**0**					
STAC.N1.β1.PINGPONG.PMEM2	Max memory used by the Producer application process at base rate in the 1:1 setup (MB)	**561**					
STAC.N1.β1.PINGPONG.TPUT1	Highest successful supply rate in the 1:1 setup (msg/sec)	**1,400,000**					
STAC.N1.β1.PINGPONG.LAT2	SupplyToReceive Latency (Hybrid) while running at PINGPONG.TPUT1 (µsec)		**3.9**	**4**	**6**	**97**	**0.5**
STAC.N1.β1.PINGPONG.LAT3	SendToReceive Latency (Hybrid) while running at PINGPONG.TPUT1 (µsec)		**3.8**	**4**	**5**	**28**	**0.4**

1. Background on the STAC-N1 Benchmark specifications

Organizations in the financial markets remain focused on reducing the latency and jitter of network communication while maintaining sufficient headroom for anticipated traffic peaks. The ability to respond to events quickly and predictably–especially during bursts of activity–continues to be crucial to profitability. At the same time, infrastructure vendors continue to deliver new and optimized solutions to these challenges. In 2012, trading organizations and vendors in the STAC Benchmark Council created a new methodology and toolset called STAC-N to test network solutions under financial market workloads. The objective was to establish a vendor-independent benchmark standard that would accommodate the current and future variety of APIs and implementations. While network performance had always been fundamental to certain STAC Benchmarks (e.g., STAC-M1 for market data feed handler solutions, STAC-M2 for market-data distribution middleware solutions), STAC-N represented the considerable group of trading applications that integrate directly with network APIs, without intervening third-party software. At first, the STAC-N tools were used only to produce research that stayed within the STAC Benchmark Council; but in 2013, STAC began making some of these benchmark results public.

The key requirements informing the development of STAC-N were to:

- provide a vendor-neutral, level playing field across numerous network APIs that nevertheless allows for configurations that exploit each API to its fullest
- provide a detailed latency analysis
- provide statistics on throughput, CPU, and memory
- automate tests and analysis as much as possible.

STAC-N is capable of representing many different kinds of message patterns and application usage patterns. Council members using the STAC-N tools are free to change these patterns as it suits them. The first set of patterns to be proposed as a benchmark standard is STAC-N1, the methodology used in this project. Further patterns and STAC-N*x* Benchmarks may be defined in the future.

The core of the test harness is the STAC-N Library, a close derivative of the STAC-M2 Library. STAC-N uses a "reflection" methodology for round-trip time-stamping, illustrated in Figure 1. A Producer transmits a "primary" message; a Consumer consumes the message and republishes it as a "reflected" message; and the Producer consumes the reflected message. Unlike STAC-M2, the reflection ratio is 100%; that is, every message from the Producer to Consumer results in a message from Consumer to Producer. (The STAC-M2 Advanced Test Harness also supports one-way measurement, but that feature has not been incorporated into STAC-N at this point.)

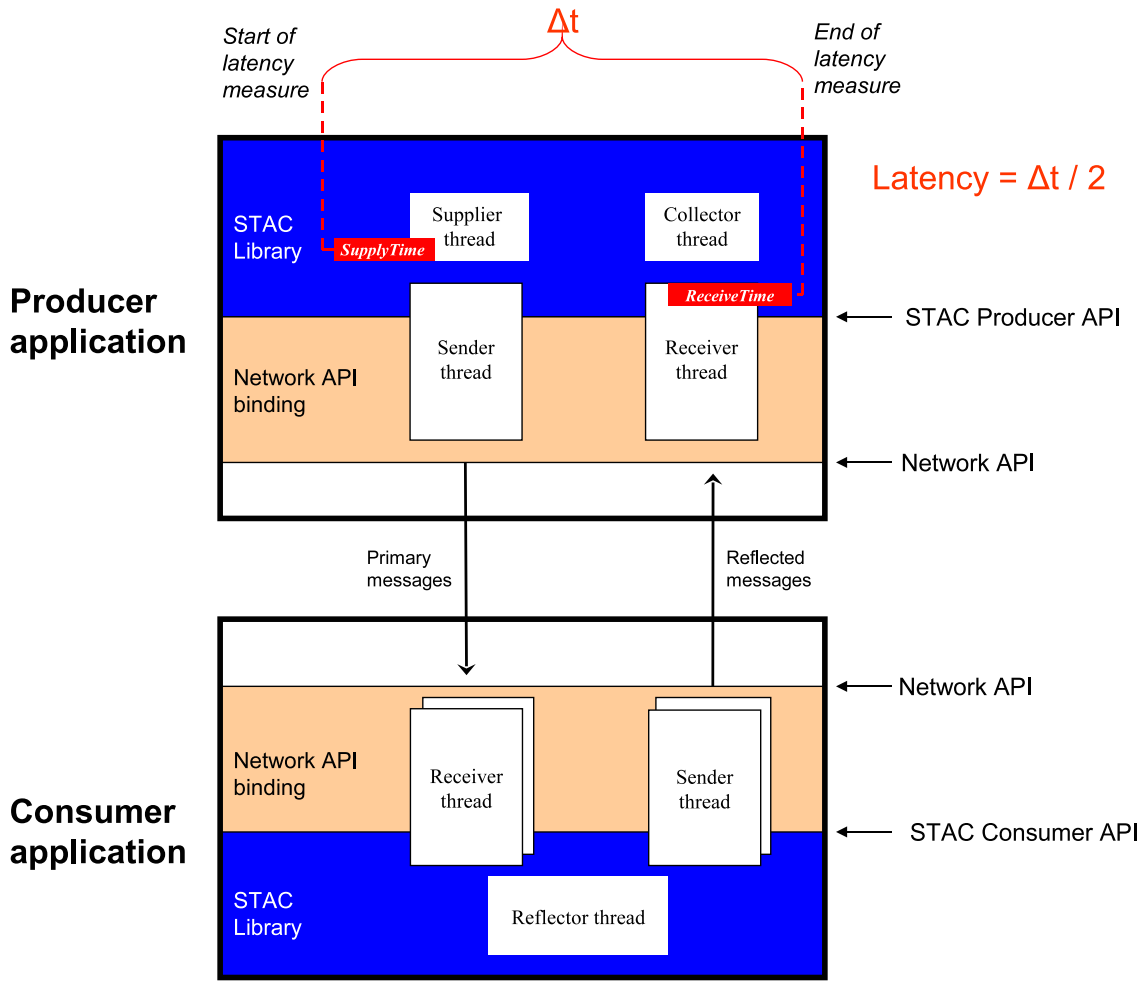


Figure 1 - Construction of STAC-N Test Clients

The STAC-N library takes a pre-defined message pattern as input. The message pattern for STAC-N1 is modeled on observed output of US equities order-book feed handlers deployed in the field, with a 232-byte payload. This message pattern is configured for a steady-state rate. The STAC-N Library supplies messages to the Producer Adapter as pointers to C structs containing a 24- to 32-byte header and the message payload.

STAC-N1 currently uses a classic "ping pong" distribution pattern. This sequence involves a single Producer and Consumer. Starting with the configured Base Message Rate (100K mps), the harness executes a pair of test runs at the given rate, then moves the rate one step size higher (Step Size = 100K mps) and repeats the process until there is a failure (see Max Message Rate, below). Once all rates have been tested, the harness automatically generates a detailed analysis report and CSV files to allow further analysis in Excel or other tools.

The harness computes several metrics from each test sequence, including the following:

- SupplyToReceive Latency (Hybrid) - (LAT1). Latency measured from the earliest moment a primary message is available for sending in the Producer to the moment the reflected message payload is available for consumption in the Producer (ReceiveTime minus SupplyTime in Figure 1). This result is divided by two to provide a "Hybrid Latency" approximation of one-way latency. (LAT1 is defined as the SupplyToReceive Latency while running at the Base Rate, and LAT2 is while running at the Max Message Rate).
- SendToReceive Latency (Hybrid) - (LAT3). Latency measured from the moment the Producer Adapter notifies the STAC Library that it has removed the message pointer from the queue, to the moment the

reflected message payload is available for consumption in the Producer. This result is also divided by two to provide a "Hybrid Latency" approximation of one-way latency. SendToReceive latency is measured only at the Max Message Rate. The difference between SendToReceive and SupplyToReceive Latency indicates how much of SupplyToReceive Latency is due to queuing.

- Max Message Rate (TPUT1). The highest supply rate that does not result in a Failure Event. A Failure Event is defined as the occurrence of a Primary Message that is supplied by the STAC Library and is not observed at a Consumer within 1 second, or the occurrence of a Reflected Message that does not arrive at the Producer within 2 seconds of the supply of its corresponding Primary Message. (In other words, there is a 1 second timeout in each direction.) Max Message Rate is only accurate to the step size configured for a test. In these tests, the step size was 100K mps, starting from a base rate of 100K mps. This means, for example, that the max rate for a SUT whose true max rate is 390K mps would be reported as 300K mps, because the test at 400K mps would fail.
- Consumer Memory 1 (CMEM1) and Producer Memory 1 (PMEM1). Maximum memory used by the Consumer or Producer application process, respectively, minus the memory allocated by the STAC Library. Memory measurements were obtained by sampling the "VmRSS" kernel statistic from the application-specific " /proc/<pid>/status" system file and finding the max value during the run. STAC Library memory is determined separately from the test runs, in a measurement taken of a no-op Consumer or Producer, as appropriate, on the platform. If the Consumer uses very little memory, small variations and rounding can cause this result to be zero or slightly negative.
- Consumer Memory 2 (CMEM2) and Producer Memory 2 (PMEM2). Maximum memory used by the Consumer or Producer application process, respectively. Memory measurements were obtained by sampling the "VmRSS" kernel statistic from the application-specific " /proc/<pid>/status" system file and finding the max value during the run. This includes memory used by the STAC Library.

Testing a given network API requires a binding between that API and the STAC-N Library. The binding used in this project is described in Section 6. The source code to the binding is available to qualified members of the STAC Benchmark Council. To request this, see www.STACresearch.com/nio.

This project used a binding to the TCP multicast functions in the Berkeley Sockets API. The author of the initial version was Solarflare. Chelsio updated the binding to provide multiple read/write, iomux, and blocking modes as well as an option to use TCP. See section 6.2, "STAC-N1 Binding used in this project", for the specific read/write/iomux/blocking modes used in this benchmark.

2. Product background

Chelsio Communications submitted the following information and claims about its products used in the SUT:

> *The Chelsio T520-LL-CR dual-port 10G Ethernet SFP+ WireDirect server adapter delivers unmatched message rates with low latency and jitter over standard Ethernet along with the lowest CPU utilization and power consumption, enabling the industry's best performance and scalability for financial services and other enterprise data centers.*
>
> *Chelsio's T520-LL-CR network interface card achieves all the requirements to make it ideal for low latency High Frequency Trading operations. Chelsio's custom T4 ASIC offers protocol acceleration for UDP/TCP/RDMA/iSCSI/FCoE/NFS/CIFS. This makes the T520-LL-CR an ideal Unified Wire adapter, simultaneously accelerating processing for all protocls with the same card/driver/firmware.*
>
> *Using Chelsio's WireDirect application acceleration middleware in combination with Chelsio's TOE ASIC on the T520-LL-CR to enable full operating system bypass dramatically reduces host processing overheads and enables high transaction rates while substantially reducing application latency with very low jitter. WireDirect performs network processing at user-level and is binary compatible with existing applications that use TCP/UDP with BSD sockets.*

3. Project participants and responsibilities

The following firms participated in the project:

- Chelsio Communications
- STAC

The Project Participants had the following responsibilities:

- Chelsio Communications sponsored the tests, provided the network adapters, developed the updated udp-tcp binding, supplied the servers, lab and server administration and tuned the SUT.
- STAC submitted the Chelsio-supplied binding for review by the STAC Network I/O SIG, supplied and supported the STAC-N test harness software, inspected the SUT, and executed the tests.

4. Contacts

For questions about Chelsio products, visit http://www.chelsio.com

For all other questions, contact info@STACresearch.com.

5. Benchmark Status

The benchmark specifications were developed by STAC and have been reviewed by members of the STAC Benchmark Council. They will be considered for ratification as standards once the Council has sufficient experience with them and is comfortable that no significant modifications are required. While the results conform to the procedures described in this document, they should not be viewed as the last word on the questions explored. Please note the limitations in Section 6.3 as well as the caveats throughout this document regarding interpretation of results.

6. Methodology

6.1 Specifications

This project followed STAC-N1 β1 Benchmark specifications. These benchmark specs have been proposed to the STAC Benchmark Council but not yet submitted for ratification. More information is available at www.STACresearch.com/nio.

6.2 STAC-N1 Binding used in this project

Producer binary name	udp_tcp_sock_producer
Producer binary version	udp-tcp-sock-v1.0
Producer binary configuration	Mode: tcp IOMUX: direct Send function: writev() Receive function: readv() Blocking: disabled
Consumer binary name	udp_tcp_sock_consumer
Consumer binary version	udp-tcp-sock-v1.0
Consumer binary configuration	Mode: tcp IOMUX: direct Send function: writev() Receive function: readv() Blocking: disabled
Compiler version used to build Producer/Consumer binaries	gcc 4.4.7-3

Key to binding config parameters:

- Mode: TCP sockets
- IOMUX: Network notification via direct socket call
- Send function: Socket library writev() function used to send messages
- Receive function: Socket library readv() function used to read messages
- Blocking: Send and receive functions are set to nonblocking

6.3 Limitations

- The STAC-N1 Test Harness relies on software instrumentation for measurement, which has an impact on performance. These limitations are consistent for all systems tested with this harness but are worth noting. In particular: 1) Because it is a reflection methodology, the latency measurements include some latency for the STAC-N Library to reflect messages (which is divided by 2 along with the rest of the round-trip latency); 2) reflection in the Consumer also takes CPU time, which increases the observed CPU utilization of the Consumer and may decrease throughput; 3) the Producer is required to call the STAC-N Library to record a "Send" timestamp, which consumes some CPU and may reduce the max sustainable rate.
- When deployed in a Consumer, the threading model of the STAC Library used for this version of the specifications emulates an application that requires business logic to execute in a different thread from message-consumption threads and requires serialized input to the business-logic thread from those other threads. This imposes a single-threaded bottleneck on performance of a single Consumer. Another common application pattern in the industry is for an application to scale across multiple cores by partitioning business logic by symbol and running multiple partition threads, each performing its own in-line message consumption. The latter pattern would potentially allow a single client to achieve higher throughput but this use case is not currently defined for STAC-N1. Similarly, each Producer has a single Supplier thread, which limits the total possible message supply rate for a given Producer. Specifications that allow an arbitrary number of Suppliers on a given machine may be added in the future but are not part of the current STAC-N1 definition.

- The message sizes of STAC-N1 are unlikely to push a high-bandwidth link to saturation. However, the maximum message rates of the harness observed in the lab exceed most of the relevant message rates for a single network interface in today's financial trading. This can be measured using an inline binding that directly connects the Supplier and Collector queues on a single machine without using a network stack. On a 2010-generation Westmere processor, such a "loopback" harness sustained over 3.3 million messages per second. Nevertheless, enhancements that allow higher rates across a single interface are under consideration for the future.
- The STAC-N test harness enforces an in-order requirement on messages. For market data, such ordering must typically be performed somewhere before reaching the application business logic, so the harness captures the performance impact, if any, of this requirement.
- The philosophy behind STAC-N1 bindings is to design each binding optimally for each API while making the same assumptions about use cases. However, these bindings may have different authors, thus they may have subtle differences in implied use cases. Reader's comparing the performance of different bindings should keep this in mind.
- The STAC-N1 sockets binding used in this project (udp-tcp-sock-v1.0) is derived from the UDP binding developed by Solarflare, with enhancements by Chelsio that were submitted for review by the STAC Network I/O SIG. As opposed to the sockets bindings used in previous projects (reports available in the STAC Vault), this is a unified binding for both TCP and UDP. One advantage of this is to make it fairer to compare TCP and UDP results, since the previous TCP and UDP bindings were different code bases. Chelsio spent considerable effort optimizing the unified binding, but the results that it yields are not identical to the previous bindings. Compared to the previous bindings, udp-tcp-sock-v1.0 has lower latency outliers at message rates below 400K mps and higher outliers above 400K mps when using the same network stacks (measured on a Sandy Bridge EP server with RHEL 6.4). The results of this comparison are available to members of the STAC Network I/O SIG at www.stacresearch.com/node/15246. The source code to both sets of bindings is also available in the STAC Vault.
- As described in Section 1, the Max Message Rate is only accurate to the granularity of the step size.

7. Stack under test

7.1 Overview

As shown in Figure 2, the SUT hardware consisted of two Supermicro servers with Chelsio T520-LL-CR network adapters linked by a pair of cross-connect cables (i.e., there was no switch). This configuration is described in more detail in the sections that follow.

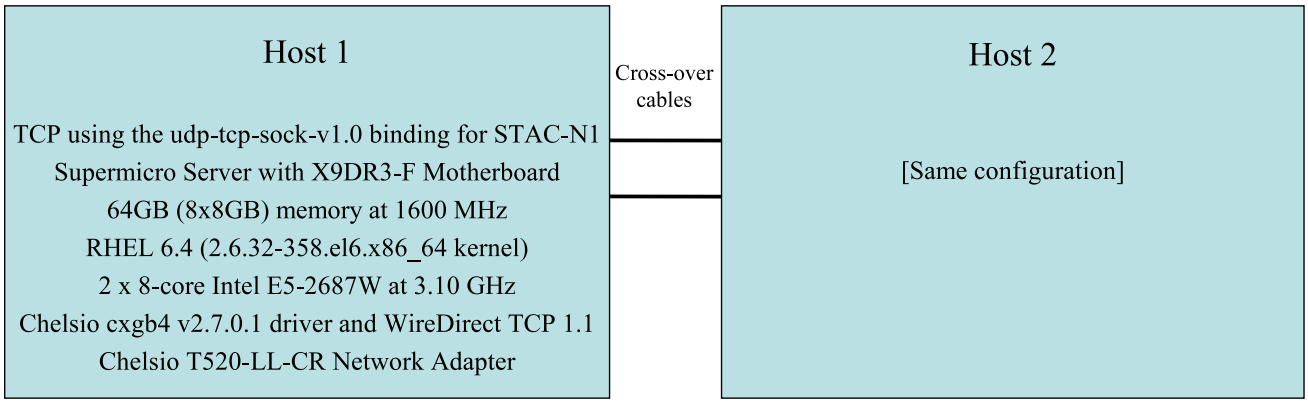


Figure 2 - SUT configuration

In these tests, the system was configured to send messages via:

TCP over 10GbE

7.2 Configuration Details

Details of the SUT configuration are available in the STAC Vault to qualified members of the STAC Benchmark Council at http://www.stacresearch.com/node/15253 . This includes a STAC Configuration Disclosure as well as the Red Hat sosreport

8. Detailed latency analysis

8.1 Section contents

The sections that follow represent latency statistics in a number of ways to facilitate analysis. These tables and charts were created automatically by STAC report-generation tools from the test data.

First is a chart of latency vs throughput at a range of throughputs (with just a few latency statistics per throughput). The range was determined by the max message rate of the system (TPUT1). See Section 1 for details about this rate.

Next are tables and charts that provide an in-depth look at latency when running two specific message rates:

- Base rate. This rate is used for all systems tested, in order to provide a common basis for comparing latency.
- Max rate (TPUT1). This rate will, in general, vary from system to system. The latencies at this rate are very important for understanding the performance profile of this system, but they are not directly comparable to those of a system with a different TPUT1.

Each analysis for a given message-supply rate contains:

- A tabular summary of latency statistics across all test runs.
- A graphical summary of the same.
- A histogram plotting frequency of occurrence against latency values from 1-10,000 microseconds using a logarithmic scale for latency that ranges from 1-microsecond to 1,000-microsecond buckets.
- Time plots of MEAN and MAX latency over the course of one test run, using 100-millisecond intervals. NOTE: Time plots are taken from just one run (the first run). For this reason, they may not correspond exactly to the summary statistics, which are taken across both runs. For example, if the max latency occurs in the second run, it will not appear in the time plot.

8.2 Test sequence PINGPONG

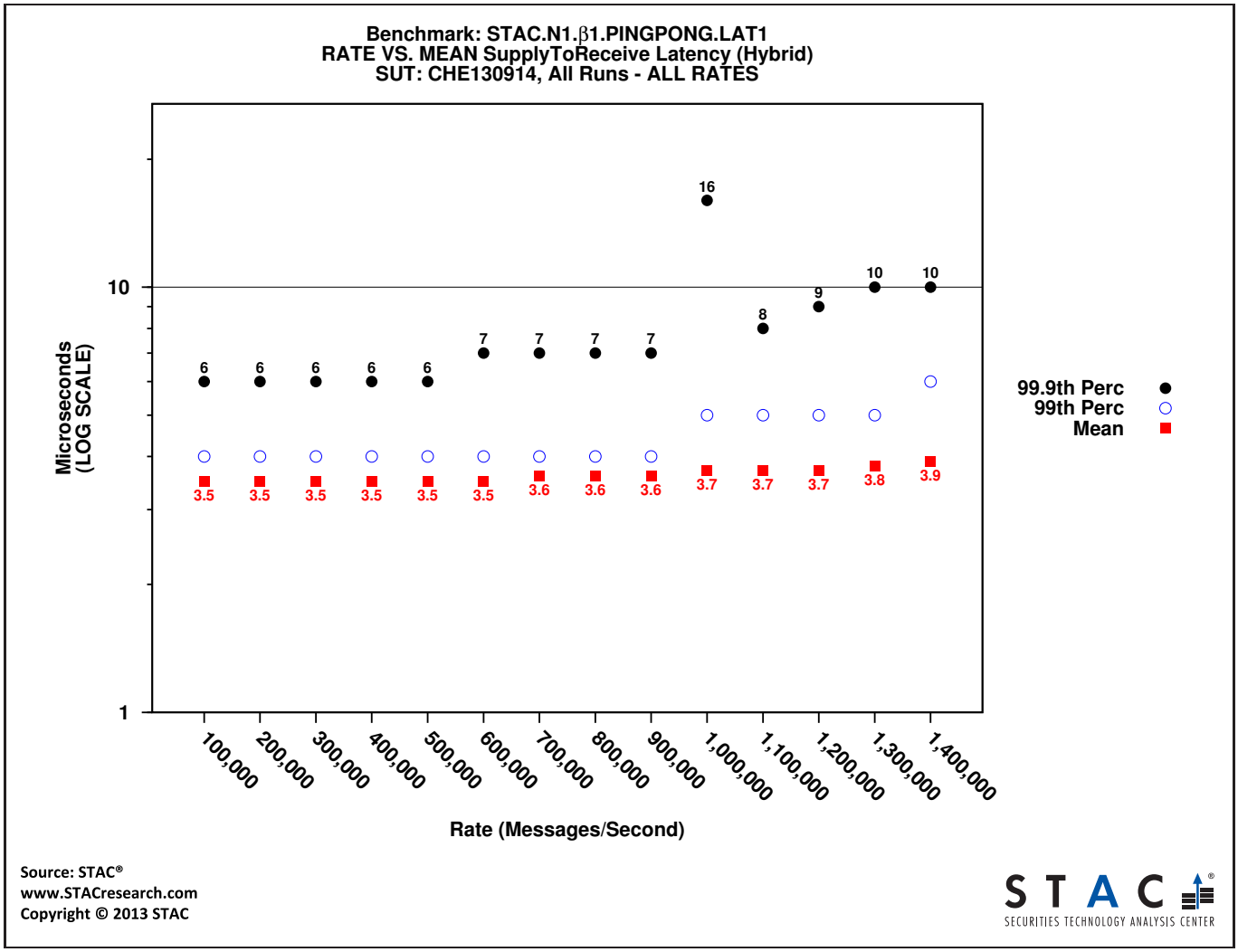


Figure 3

8.2.1 BASE RATE

8.2.1.1 Summary stats for BASE RATE runs in test sequence PINGPONG

STAC Benchmark(TM): STAC-N1.β1.PINGPONG.LAT1 SUT: CHE130914						
RATE (messages per second)	Latency statistics (microseconds)					
	Mean	Median	Std Dev	Minimum	Maximum	99th Perc
100,000	3.5	3	0.2	3	63	4
Supply-Rate Jitter, Run 1: 0.00%			Supply-Rate Jitter, Run 2: 0.08%			

Figure 4

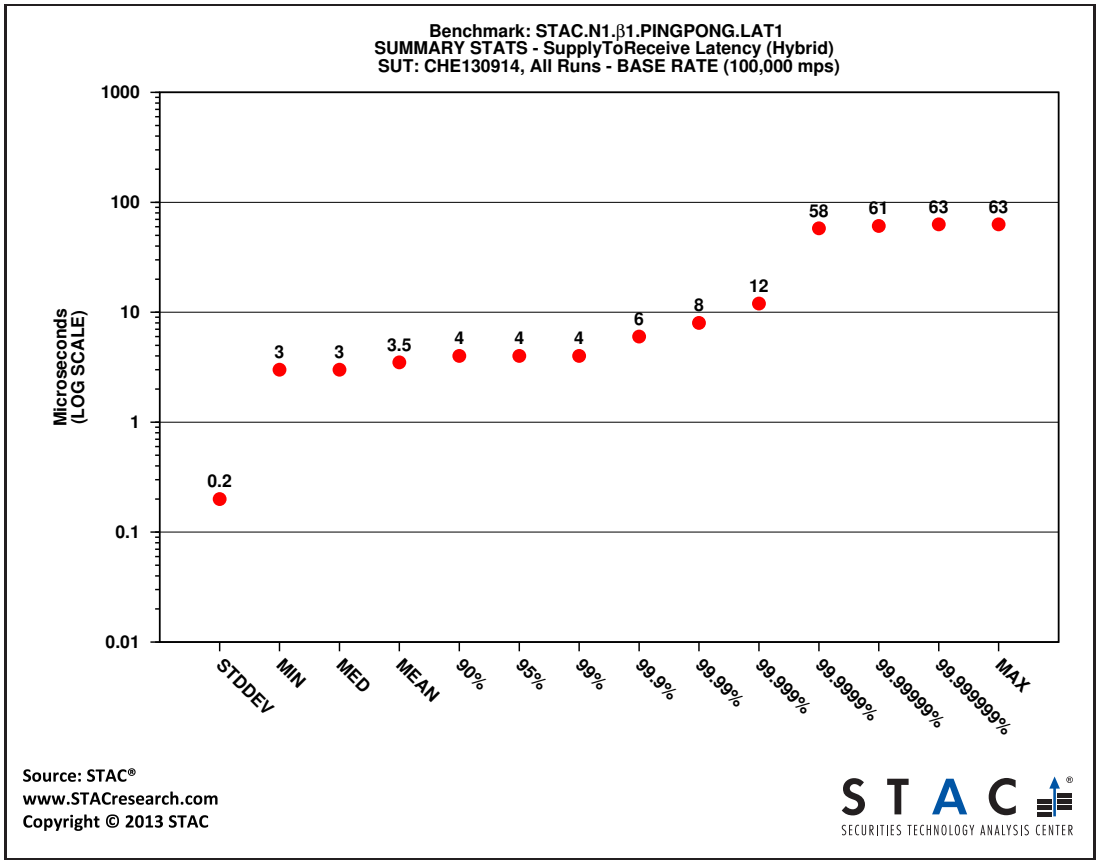


Figure 5

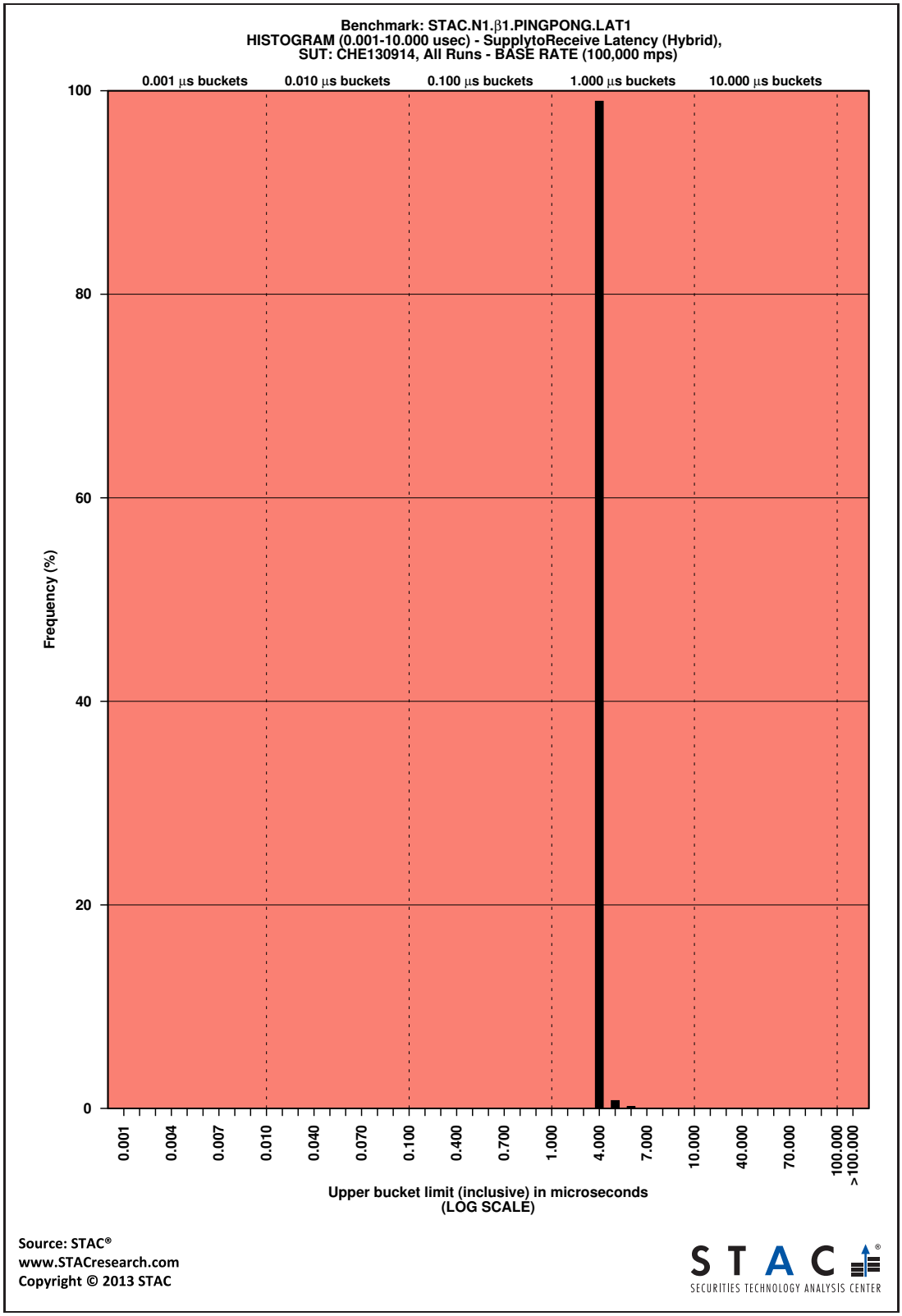

Benchmark: STAC.N1.β1.PINGPONG.LAT1
HISTOGRAM (0.001-10.000 usec) - SupplytoReceive Latency (Hybrid),
SUT: CHE130914, All Runs - BASE RATE (100,000 mps)
0.001 µs buckets
0.010 µs buckets
0.100 µs buckets
1.000 µs buckets
10.000 µs buckets
Frequency (%)
100
80
60
40
20
0
0.001
0.004
0.007
0.010
0.040
0.070
0.100
0.400
0.700
1.000
4.000
7.000
10.000
40.000
70.000
100.000
>100.000
Upper bucket limit (inclusive) in microseconds
(LOG SCALE)
Source: STAC®
www.STACresearch.com
Copyright © 2013 STAC
STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Figure 6

8.2.1.2 Example stats for BASE RATE runs in test sequence PINGPONG

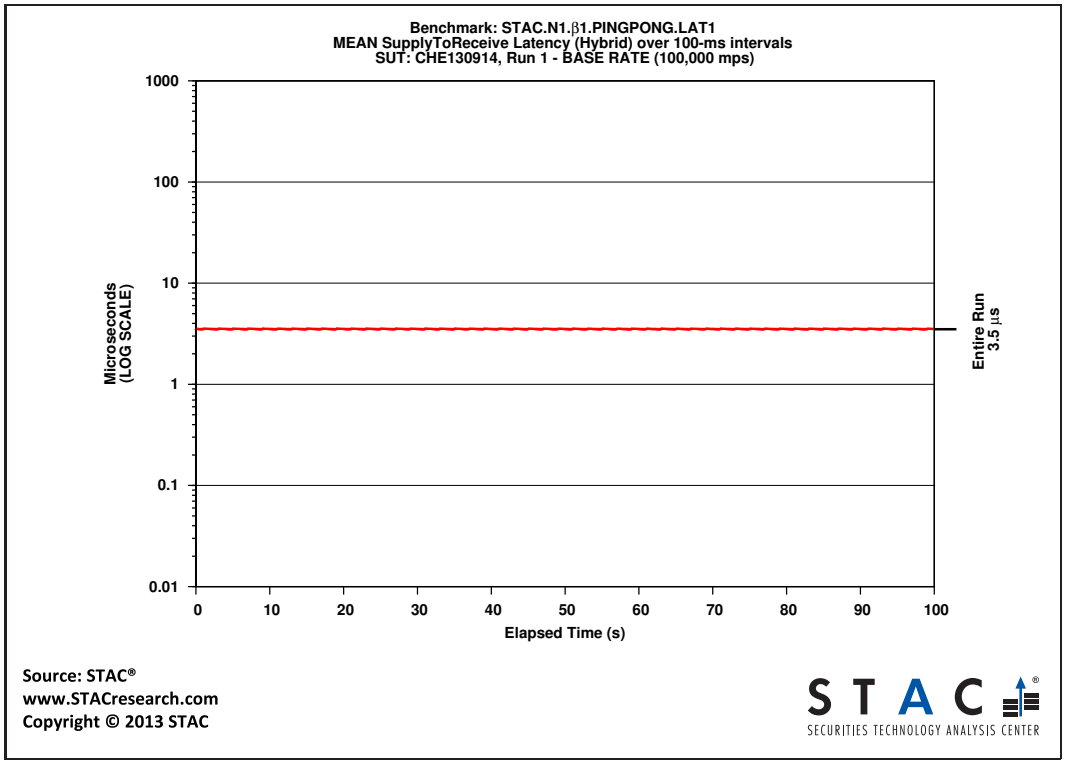


Figure 7 - Mean Latency over time at BASE RATE

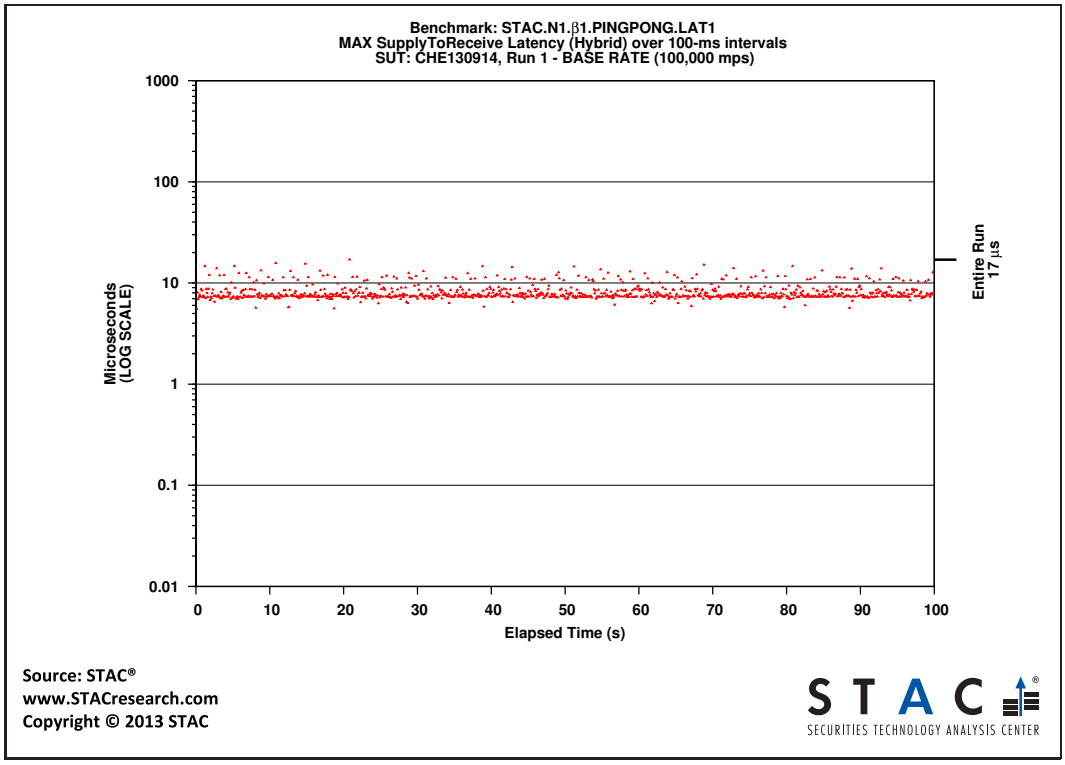


Figure 8 - Max Latency over time at BASE RATE

8.2.2 HIGHEST SUCCESSFUL RATE

8.2.2.1 Summary stats for HIGHEST SUCCESSFUL RATE runs in test sequence PINGPONG

STAC Benchmark(TM): STAC-N1.β1.PINGPONG.LAT2 SUT: CHE130914						
RATE (messages per second)	Latency statistics (microseconds)					
	Mean	Median	Std Dev	Minimum	Maximum	99th Perc
1,400,000	3.9	4	0.5	3	97	6
Supply-Rate Jitter, Run 1: 0.04%			Supply-Rate Jitter, Run 2: 0.05%			

Figure 9

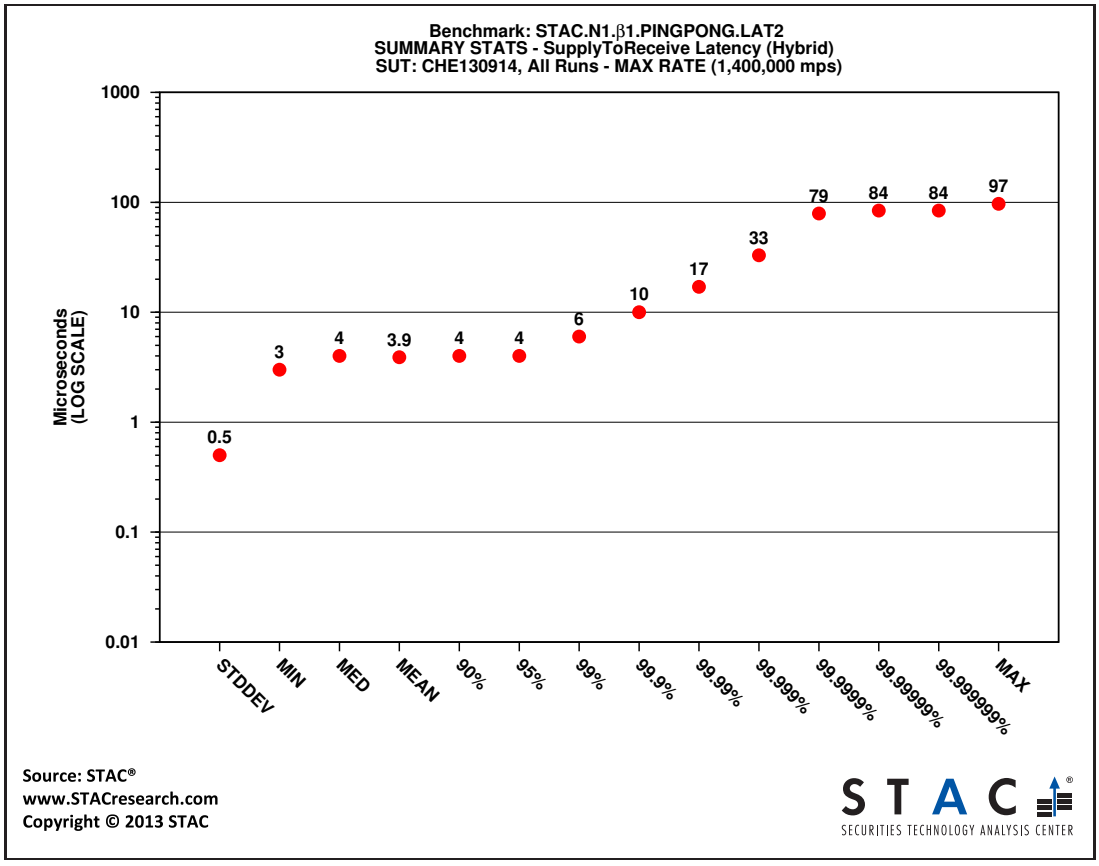


Figure 10

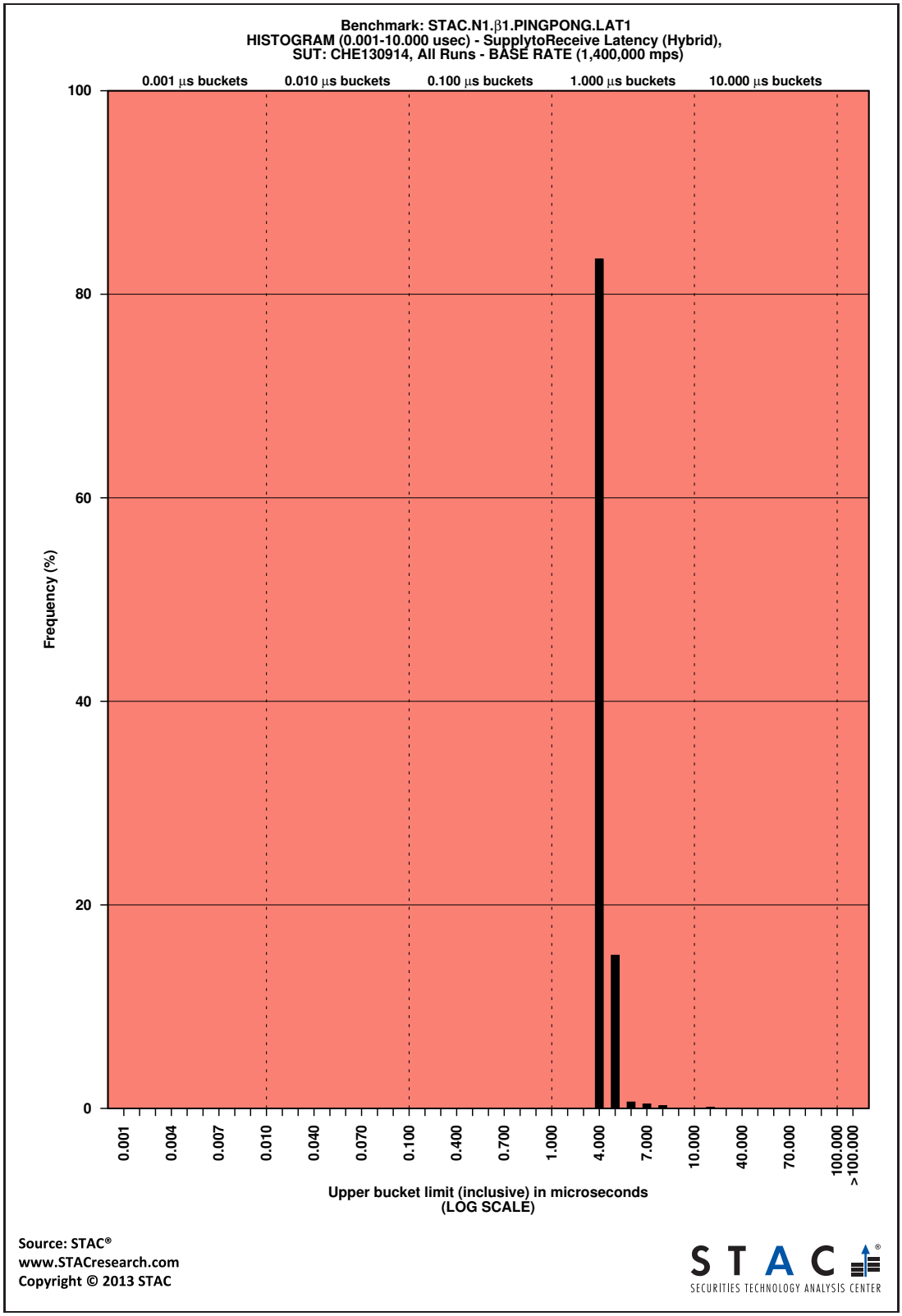

Benchmark: STAC.N1.β1.PINGPONG.LAT1
HISTOGRAM (0.001-10.000 usec) - SupplytoReceive Latency (Hybrid),
SUT: CHE130914, All Runs - BASE RATE (1,400,000 mps)
0.001 µs buckets
0.010 µs buckets
0.100 µs buckets
1.000 µs buckets
10.000 µs buckets
100
80
60
40
20
0
Frequency (%)
0.001
0.004
0.007
0.010
0.040
0.070
0.100
0.400
0.700
1.000
4.000
7.000
10.000
40.000
70.000
100.000
>100.000
Upper bucket limit (inclusive) in microseconds
(LOG SCALE)
Source: STAC®
www.STACresearch.com
Copyright © 2013 STAC
STAC
SECURITIES TECHNOLOGY ANALYSIS CENTER

Figure 11

8.2.2.2 Example stats for HIGHEST SUCCESSFUL RATE runs in test sequence PINGPONG

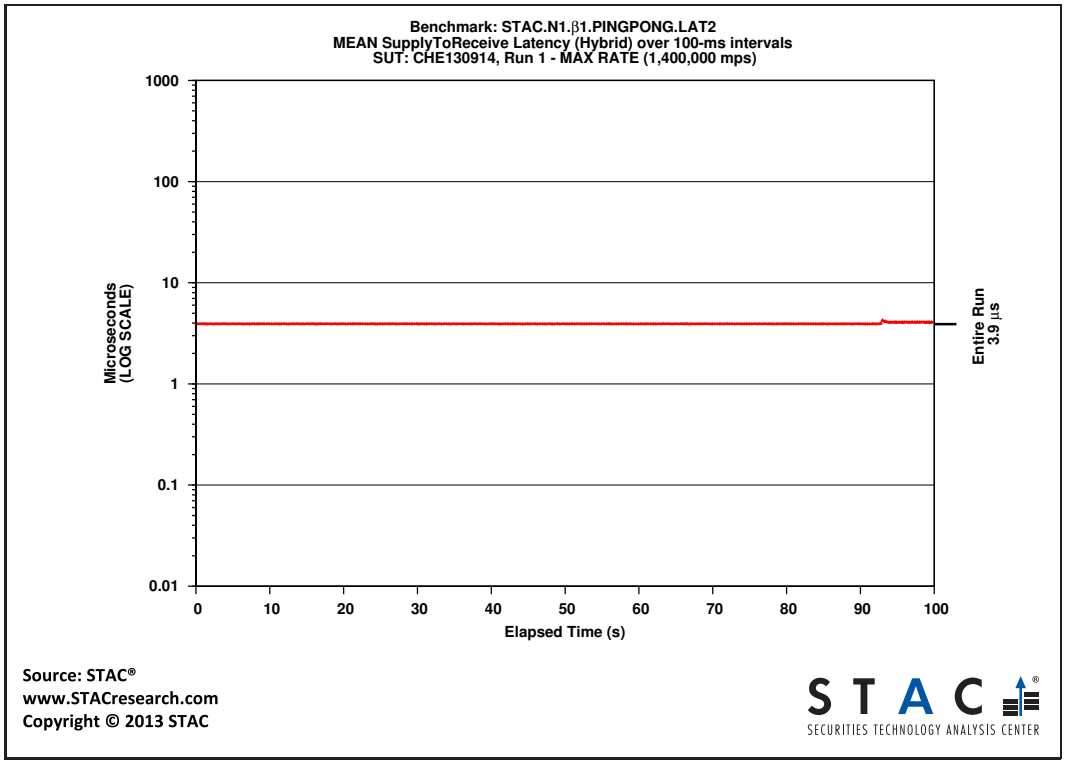


Figure 12 - Mean Latency over time at HIGHEST SUCCESSFUL RATE

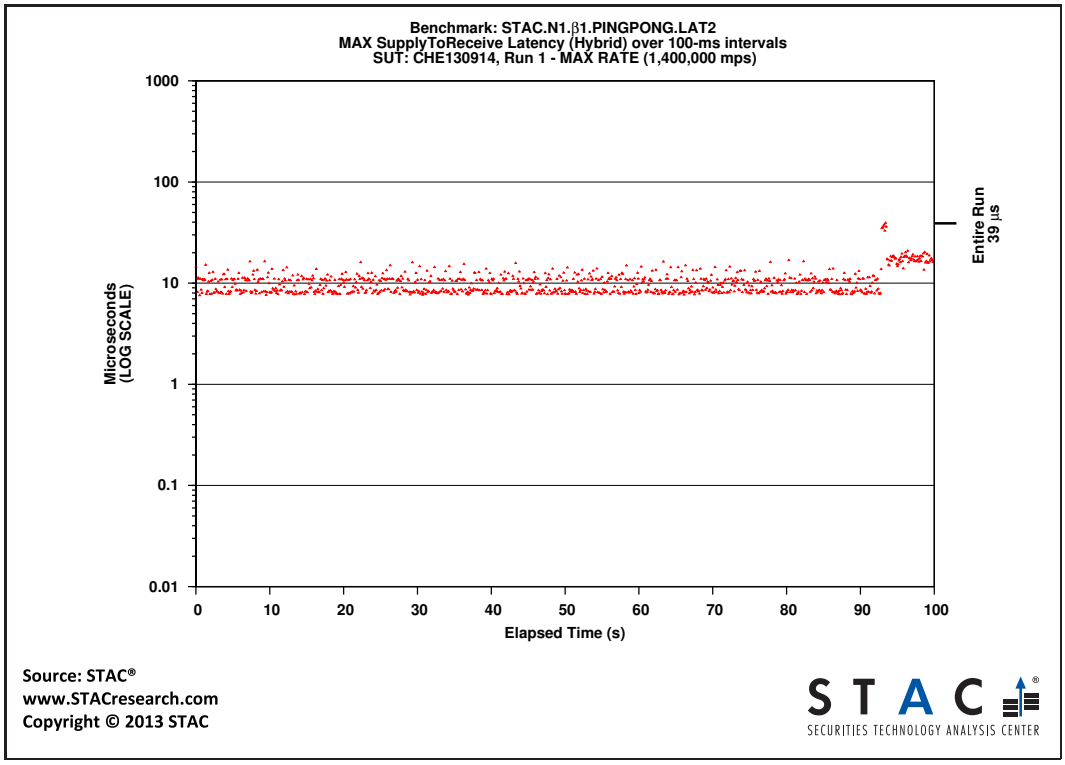


Figure 13 - Max Latency over time at HIGHEST SUCCESSFUL RATE

9. Vendor Commentary

None.

10. STAC Notes

None

About STAC

STAC is a technology-research firm that facilitates the STAC Benchmark Council™ (www.STACresearch.com/members), an organization of leading trading organizations and vendors that specifies standard ways to assess technologies used in the financial markets. The Council is active in an expanding range of low-latency, big-compute, and big-data workloads.

STAC helps end-user firms relate the performance of new technologies to that of their existing systems by supplying them with STAC Benchmark reports as well as standards-based STAC Test Harnesses™ for rapid execution of STAC Benchmarks in their own labs. End users do not disclose their results. Some STAC Benchmark results from vendor-driven projects are made available to the public, while those in the STAC Vault™ are reserved for members of the Council (see www.STACresearch.com/vault).

To be notified when new STAC Benchmark results become available, please sign up for free at www.STACresearch.com.