Author: Robert Schiller
Date: 29 June 2020

Network Benchmarks for Phase2 Hardware

Overview

This report discusses networking benchmarks run on some of the hardware being considered for use by the CMS Tracker Phase2 development teams, and describes how users can run the tests themselves as new hardware becomes available. A previous report by Joshua Cepeda discussed processor benchmarks that have been integrated into a testing suite with the networking benchmarks; this suite and its accompanying documentation are available now on GitHub: https://github.com/centipeda/soc-benchmark.

The two most useful networking performance metrics are throughput and latency. Throughput is the amount of payload data that can be transferred across the network per second. The distinction between payload and total data is key because packet headers introduce some overhead, and tuning the packet size can allow more payload to be sent for the same amount of total data. Latency is a measure of the propagation time across a network. Specifically, it is typically given as the round-trip time for a packet to traverse the network and return to the host: the local host sends a packet to the remote host, which echoes the packet back, and the local host compares the timestamps between when it sent the packet and when it received the reply.

The networking benchmarks presented here include throughput for TCP and UDP communication, latency, and some other relevant metrics. The following paragraph provides more details about TCP and UDP and can be skipped for readers who are familiar with them.

Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are transport-layer protocols that run on top of the Internet Protocol (IP). They organize data from the application layer, which is then packetized by IP at the internet layer, and add a checksum to verify the received data. TCP is more sophisticated, including features like congestion control (adjusting the rate at which data is sent when the network is heavily trafficked), maximum transmittable unit (MTU) calculation to determine the most efficient packet size, and packet retransmission to prevent loss. Another important implementation difference is that TCP specifies a stream of data, whereas UDP packetizes data itself before the internet layer (where it might be further fragmented). The UDP packet size is configurable, so a significant amount of manual tuning is required to achieve maximum performance. Factors like the level of traffic on the network and the MTU of all network components can alter the most efficient packet size. UDP is typically reserved for low-latency applications where speed is more important than reliability.

Benchmarks

Iperf3

Iperf3 is a widely-used benchmarking tool for measuring TCP and UDP throughput. A remote host operates in *server* mode as the receiver, and the local host is the *client* and sender.

Iperf3 version 3.1.7 was used for all results presented here. Iperf3 is a complete rewrite of the original iperf benchmarking suite, so all future tests should be done using iperf3. Additionally, although the specific version of iperf3 is not likely to greatly impact results, version 3.1.7 should be used whenever available for maximum consistency.

Ping

Ping is a very basic utility that sends an ICMP packet to a remote machine and records the round-trip time. This provides a measure of network latency.

Scripting

A repository is available on GitHub to automatically run benchmarking tests (the networking tests discussed here as well as general purpose processing tests Whetstone, Dhrystone, and Coremark). Some minimal setup is necessary to ensure the correct packages are installed, but the script makes it simple to run the benchmarks and generate statistics. When available, the results can be tabulated manually in the repository, so comparisons between the hardware can be made. Full documentation of the setup and command line flags is in the repository as well.

Results

For the following results, CPU usage is the average percentage of CPU time used by iperf3 and its child processes. Jitter (UDP) is the mean deviation in latency (iperf3 does not attempt to establish a synchronized clock between hosts, so latency is biased in general; the bias vanishes in jitter). Packets are labeled with an index, so missing indices are counted as packet losses (UDP). For TCP, packets are retransmitted if necessary, so there is no loss. Throughput is the amount of data successfully transferred per unit time.

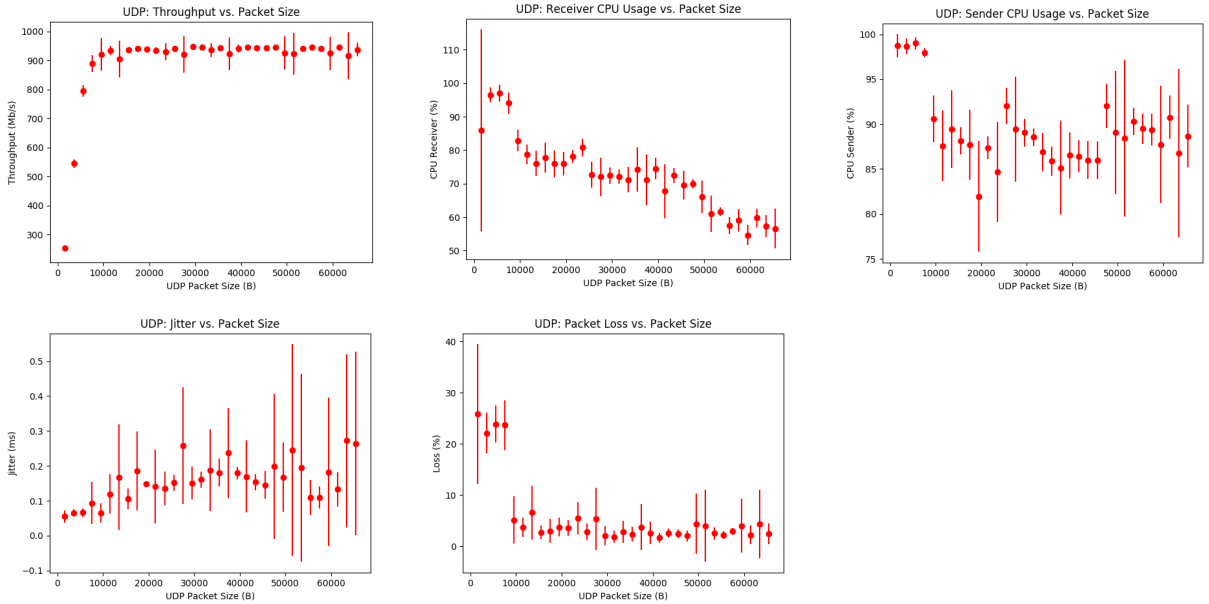
For the UDP tests, there are several command line flags that must be set to ensure interpretable results. They are listed in the documentation on the GitHub repository and are discussed at greater length in the iperf3 manual. The size of the UDP packets is configurable, so the tests are run for a range of packet sizes up to the maximum of 65,507 bytes.

TIF: Serenity 13 => Serenity 16

- Intel Atom E3825 @ 1.33 GHz
- 1 Gb/s Ethernet link (MTU: 1500 MB)
- CentOS 7
- Latency (RTT): 0.97 ± 0.13 ms

Throughput (Mb/s)	937.89 ± 0.91
Total Data Transferred (GB)	1.17 ± 0.05
CPU Usage by Sender (%)	28.78 ± 5.522
CPU Usage by Receiver (%)	11.05 ± 10.07
Retransmissions (# of packets)	0 ± 0

Iperf3 benchmarks for TCP communication between TIF Serenity boards. Statistics are generated from 10 runs.



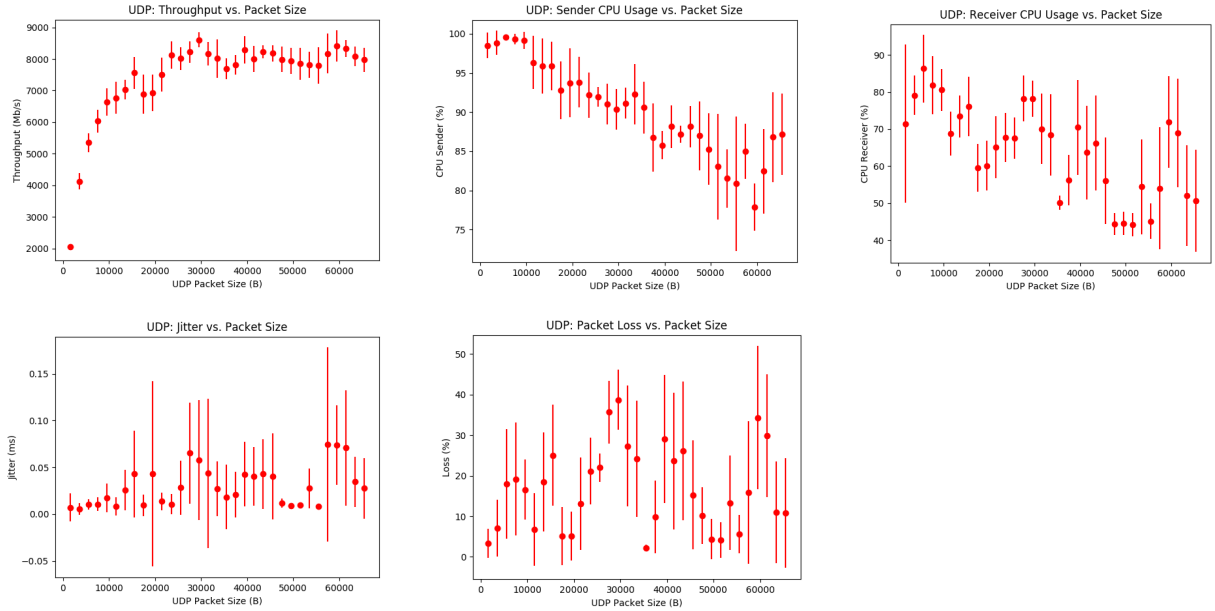
iperf3 UDP test results on TIF Serenity boards. Each test run was 10 seconds, and statistics are generated from 10 runs at each datagram size.

Notre Dame CRC: FE01 => FE02

- Intel Xeon E5-2680 @ 2.50GHz
- 10 Gb/s Ethernet link (MTU: 1500 MB)
- RHEL v. 7.8
- Latency (RTT): 0.07 ± 0.02 ms

Throughput (Mb/s)	9320 ± 51
Total Data Transferred (GB)	10.8 ± 0.1
CPU Usage by Sender (%)	13.19 ± 4.08
CPU Usage by Receiver (%)	16.17 ± 6.23
Retransmissions (# of packets)	67 ± 41

Iperf3 benchmarks for TCP communication between head nodes on the Notre Dame Center for Research Computing cluster. Throughput is presented in Mb/s for consistency. Statistics are generated from 10 runs.



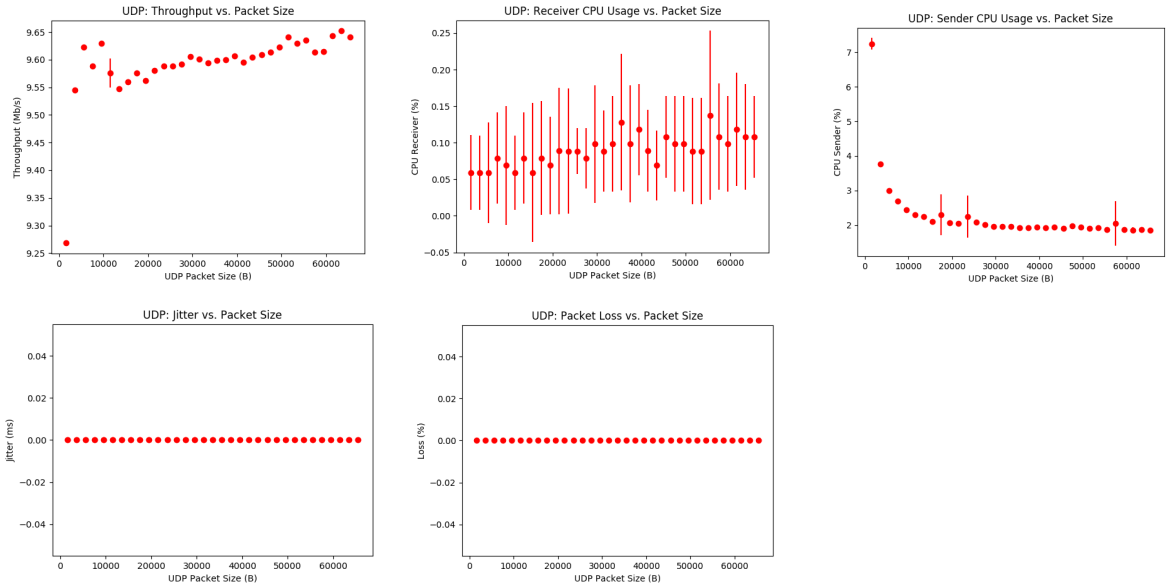
iperf3 UDP test results on Notre Dame CRC head nodes. Each test run was 10 seconds, and statistics are generated from 10 runs at each datagram size.

TIF: Apollo 05 => Serenity 16

- *Sender*: ARM Cortex-A9 @ 800 MHz; *Receiver*: Intel Atom E3825 @ 1.33 GHz
- 10 Mb/s Ethernet link (MTU: 1500 MB)
- CentOS 7
- Latency (RTT): 0.66 ± 0.20 ms

Throughput (Mb/s)	8.99 ± 0.24
Total Data Transferred (GB)	0.0107 ± 0.0001
CPU Usage by Sender (%)	0.84 ± 0.18
CPU Usage by Receiver (%)	6.3 ± 1.53
Retransmissions (# of packets)	0 ± 0

Iperf3 benchmarks for TCP communication between head nodes on the Notre Dame Center for Research Computing cluster. Statistics are generated from 10 runs.



iperf3 UDP test results between TIF Apollo and TIF Serenity 16. Each test run was 10 seconds, and statistics are generated from 10 runs at each datagram size.

Discussion

The TCP tests ran well straight out of the box, delivering sensible results with no modifications. The UDP tests were somewhat less straightforward, since the default settings for iperf3 resulted in significantly lower throughput than TCP. The bandwidth setting was increased to match the theoretical maximum bandwidth for the link, and the throughput was recorded at various datagram (UDP packet) sizes. At low datagram sizes, the CPU usage is high for both the sender and receiver, since many datagrams must be produced to transfer data. As the datagram size increases, the throughput increases quickly and plateaus at a maximum value, while CPU usage declines. As expected, the maximum throughput for UDP after tuning was, in most cases, slightly higher than for TCP.

The TCP performance of networking between the TIF Serenity boards is consistent with a theoretical 1Gb/s maximum bandwidth. Iperf3 records payload throughput, so the overhead incurred by TCP/IP headers is not included in the results. This accounts for some of the difference between the theoretical bandwidth and the measured throughput. For UDP, the packet loss dropped dramatically in between 7500 and 9500 MB datagrams. The reason for this is unclear, especially since the network uses standard 1500 MB Ethernet frames.

On the current version of the TIF Apollo board, the Ethernet connection is physically limited to 10 Mb/s, which is reflected in the throughput measurements. Interestingly, for UDP the CPU usage is quite low, and both jitter and packet loss are identically zero. These results are generally explained by the low bandwidth, but the jitter measurement still seems anomalous. No good explanation has been found for this, so it is likely that either the measured jitter is less than the precision of iperf3 (not impossible- most of the network is running far below its

capacity and it was likely empty apart from the iperf3 tests), or there is simply a bug in the program.

To assess the effect of CPU speed on network throughput (since the SoC modules are relatively underpowered), the iperf3 tests were also run on the head nodes of Notre Dame's Center for Research Computing cluster. The graphs show the same behavior.

These benchmarks are intended to give a picture of the hardware's networking capabilities under close to ideal conditions. The level of external traffic on the networks was not accounted for, but it was likely limited in most cases, with the exception of the tests between the Notre Dame CRC head nodes. A busy network would generally produce greater packet loss for UDP, along with lower throughput for both UDP and TCP.

When reporting results, it is important to specify the hardware for both the client and the server, since both machines affect the results. For UDP servers with less powerful CPUs, for example, the socket buffers might be filled before the system has time to handle the data, resulting in more dropped packets than for more powerful servers.