D1.2 Network Load Balancing

Ronald van der Pol, Freek Dijkstra, Igor Idziejczak, and Mark Meijerink

SARA Computing and Networking Services,
Science Park 121, 1098 XG Amsterdam,
The Netherlands
June 2010
ronald.vanderpol@sara.nl,freek.dijkstra@sara.nl,
igor.idziejczak@sara.nl,mark.meijerink@sara.nl
http://nrg.sara.nl/

1 Introduction

SARA's contribution to the GigaPort3 network research is in the area of demanding applications. There is still a large gap between the bandwidth that wide area networks can provide and the speed at which a single server can send and receive data on the network. SARA is working on identifying the bottlenecks in the the overall end-to-end performance and trying to remove those bottlenecks in order to get a higher throughput between applications running on different servers.

This document describes the network load balancing analysis and tests that SARA has done in order to overcome the limited capacity of a network interface card. The capacity of network cards is and always has been more limited than the capacity of links in the wide area network. The aim in this research is to combine the capacity of multiple network cards in a single server to allow for a single stream that exceeds the capacity of each individual network card.

This work builds on server scalability tests previously performed by SARA [1].

The test setup is described in section 2. Section 3 provides the test results and section 4 summarises the conclusions.

2 Test Setup

The tests described in this document were done with two servers. Table 1 lists the specifications of the two servers

	Server A	**Server B**
Brand	Dell PowerEdge T610	Dell Precision WorkStation T3400
CPU	Intel Xeon X5550 @2.66GHz	Intel Core Duo E6550 @2.33 GHz
Memory	6 GiB (3× 2 GiB) DDR3 @1333 MHz	2 GiB (2× 1 GiB) DDR2 @667 MHz
Disk	2× 2.5" 250 GB 7200 RPM HDDs 6× 2.5" 160 GB Intel X25-M SSDs	1× 2.5" 250 GB 7200 RPM HDD
Network	2× Intel 10GE UTP NICs 1× Myricom 2 port 10GE SFP+ NICs	2× Intel 10GE UTP NICs 1× Myricom 2 port 10GE SFP+ NICs
OS	Linux 2.6.31 (64 bit)	Linux 2.6.31 (64 bit)

Table 1. Specs of the servers used in the experiments

Whilst the servers could each carry up to four 10 Gb/s Ethernet interfaces, only two of these interfaces were used at the same time, due to limitations of the PCI express slots in the servers.

In addition to the 10 Gb/s links, the servers were connected to the regular Internet on a 1 Gb/s interface (not shown). This was to ensure that the ssh session to manage the tests would not interfere with the actual test itself. Most experiments were carried out by connecting the Ethernet interfaces on the Myricom NIC to an Arista 7124S Ethernet switch, as shown in figure 1. The switch was configured with two separate VLANs, resulting in two logical links between the servers. The links could be used individually, or be grouped together using link bundling, forming one logical 20 Gb/s link.

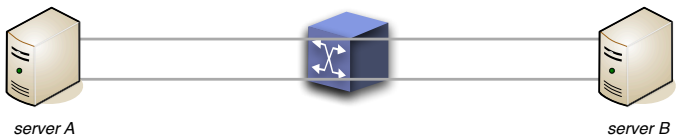

Fig. 1. Experimental setup.

The link bundle was configured to deploy a round robin policy, meaning that packets would evenly be distributed over the two links, as opposed to more common policy to distribute the data based on check-summing the source and destination addresses. The round robin policy ensured that even a single stream could exceed the 10 Gb/s capacity of a physical link.

The disadvantage of the round robin policy is possible packet reordering because data of a single stream is distributed over two physical links. To study this effect, an additional setup was created so that one of the two links would traverse a 17 ms loopback from Amsterdam to Geneva and back, while the other link would retain its < 1 ms latency.

3 Results

Several throughput measurements were done with the topologies described in the previous section. UDP and TCP traffic was sent between the servers with netperf. Each measurement was done with various socket buffer sizes (from 32 kB to 32 MB) and various message lengths (from 64 bytes to 30,000 bytes). Each run lasted for 30 seconds.

3.1 Single Local 10GE Link

In this test setup both servers were connected to the Ethernet switch with a single 10GE link. Both ports on the Ethernet switch were in the same VLAN and Ethernet flow control was enabled and in use on both ports. The Ethernet MTU on both servers was set to 9000. The topology of this setup is shown in figure 2.

Fig. 2. Both servers connected with a 10GE link via the Ethernet switch.

The results of sending UDP and TCP traffic with netperf from server A to server B are shown in figure 3.

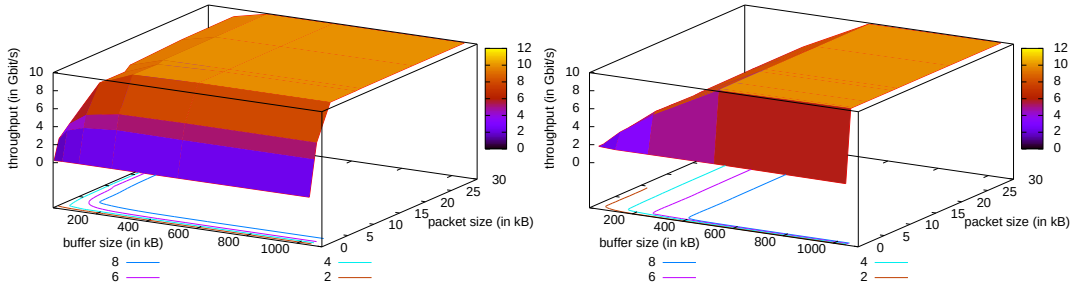

Fig. 3. UDP (left) and TCP (right) traffic from server A to server B.

The x-axis shows the various socket buffer sizes in kilobytes, the y-axis shows various packet sizes in kilobytes and the z-axis shows the throughput in Gbit/s.

The graph on the left in figure 3 is the received packet rate on the receiving server B. The graph on the right of figure 3 shows the TCP throughput. Because TCP is a reliable protocol, the packet rate at both the sending and receiving end are the same.

The graph on the left in figure 3 shows that for UDP the link can be saturated for almost all socket and packet sizes. Only when using a packet size of less than about 1500 bytes the maximum throughput is not reached. This is also true when the socket buffer size is less than about 100 kilobyte.

The graph on the right in figure 3 shows that larger buffer sizes are needed to saturate the link in the case of TCP than is the case when using UDP. The reason is that TCP needs a large enough buffer size at the receiver in order to keep the link filled with packets. When the receiving buffer is too small, the sender needs to wait for an acknowledgement of packets in transit in order to be allowed to send additional packets.

The results of sending UDP and TCP traffic the other way around (from server B to server A) are shown in figure 4.

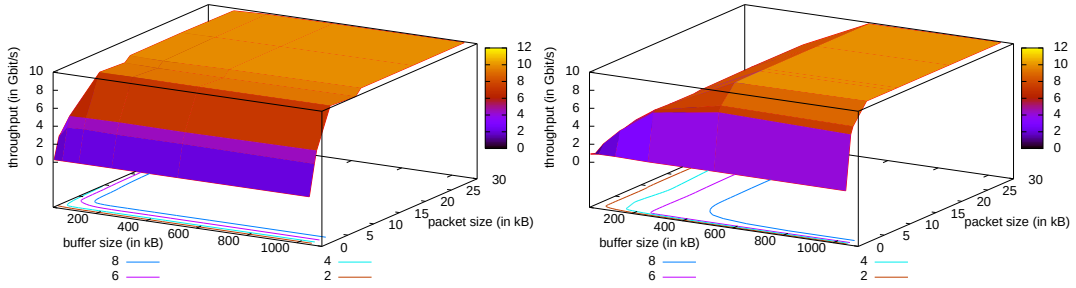

Fig. 4. UDP (left) and TCP (right) traffic from server B to server A.

When comparing the graphs for both traffic directions several observations can be made. The effect of socket buffer size is roughly the same for both directions. However, UDP performs

slightly better for small packets when sending from server B to server A. Server A is the faster of the two servers and therefore it can keep up better with the incoming UDP packets than server B. For TCP, traffic in both directions performs equally for all packet sizes.

3.2 Single 10GE Link with 17 ms latency

UDP and TCP traffic was sent over a link with a large roundtrip time. This topology is shown in figure 5.

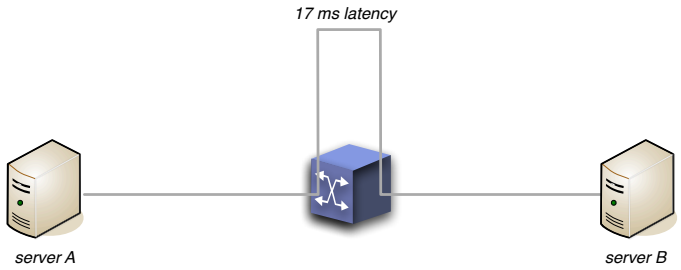

Fig. 5. Both servers connected with a 10GE link with a 17 ms latency.

The results of traffic sent from server A to server B are shown in figure 6. The graph on the left shows the UDP traffic received at server B and the graph on the right shows the TCP throughput.

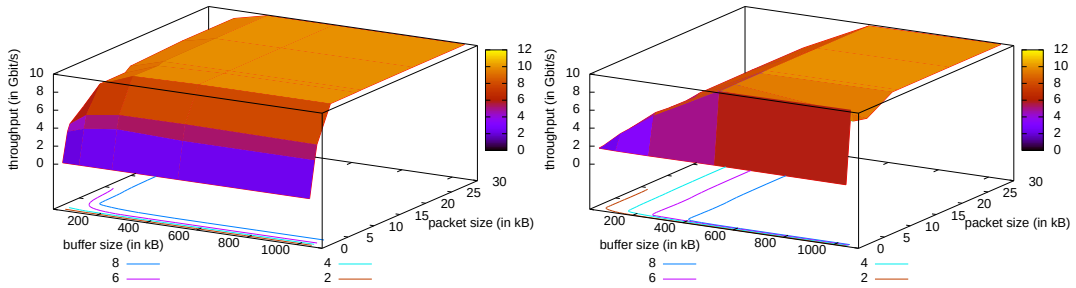

Fig. 6. UDP (left) and TCP (right) traffic from server A to server B over a link with 17 ms latency.

When comparing the results shown in figure 6 with the results shown in figure 3 the following observations can be made. The results for UDP are almost identical. This is to be expected when the packet loss between the short local connection and the longer connection are the same. UDP has no feedback mechanism; packets are just sent from A to B and the distance of the link has no influence.

For TCP it is expected that larger socket buffer sizes are needed when the latency increases. This can not be observed in the graph on the right of figure 6. In theory, an increase in latency from 0.02 ms (local connection) to 17 ms would also result in an 850-fold increase of the socket buffer for reaching the same throughput. It is not clear why this is not reflected in the

measurements. Also, the reason for the dip in the TCP throughput for small packet sizes is not clear.

3.3 Ethernet Channel with Two Equal Length 10GE Links

In this test setup both servers were each connected with two 10GE ports to the Ethernet switch. On the Ethernet switch two VLANs were configured. Each server had one 10GE port connected to one VLAN and the other 10GE port connected to the other VLAN, as shown in figure 7.

Fig. 7. Experimental setup for bonding of two equal length links.

The Ethernet channel was setup between the two servers, not between a server and the Ethernet switch. The Ethernet switch was transparent with respect to the Ethernet channel. Load balancing between the two individual links of the Ethernet channel was done via round robin load balancing.

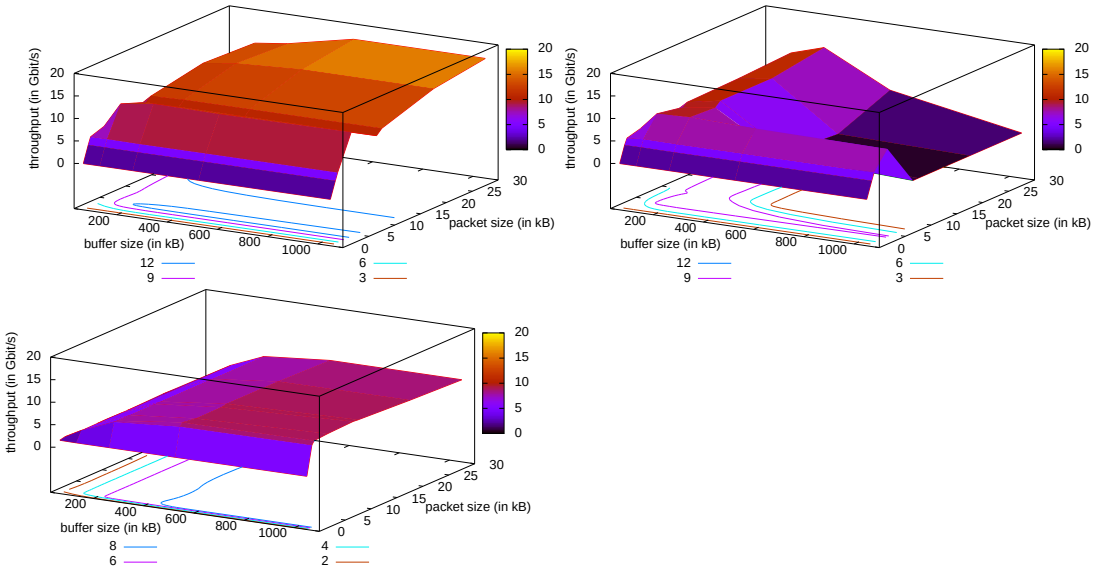

Fig. 8. Traffic from server A to server B over a 2x 10GE Ethernet channel (top left: UDP sending rate; top right: UDP receiving rate; bottom: TCP throughput

Figure 8 shows the results of sending traffic via the 20 Gbit/s Ethernet channel from server A to server B.

The top left graph in figure 8 shows the rate of packets sent by server A, while the top right graph in figure 8 shows the results of the rate of packets received by server B. Comparing both pictures clearly shows that the throughput measured at the receiving end is much lower for large socket buffer sizes and packet sizes than what is sent by server A. Indeed it seems that the performance collapses above a certain sender throughput (roughly at 12-13 Gb/s). Other measurements have shown that the packets arrive at the receiving interface, but are dropped somewhere on the way to the netperf application. It is not clear yet where exactly these packets are dropped. Our hypothesis is that the CPU of the receiving server can not keep up, and is dropping packets in the kernel. Since the kernel has no means to signal the sender to slow down, it it flooded, and the performance collapse. Unfortunately, this hypothesis could not yet be confirmed by additional measurements, and further analysis needs to be done.

When looking at the packet rate received by server B, the top right graph in figure 8, the graph contains a small ridge. Irrespectively of the socket buffer, there is a small ridge around 8000 byte IP packet size and a small valley around 9000 byte IP packet size. This can possibly be explained by the effect of fragmentation. The Ethernet MTU on the Ethernet channel was set to 9,000 bytes. The IP packet sizes that were sent during the measurements consisted of a range of increasing sizes of 64, 1472, 4096, 8984, 10k, 20k and 30k bytes. The throughput is increasing for IP packet sizes from 64 to 4096 bytes and dropped for IP packet sizes from 8984 and higher. 8984 bytes plus UDP header does not fit anymore in the 9,000 byte Ethernet MTU. Therefore, from IP packet size 8984 and up, the IP packets are fragmented. Since the load balancing mechanism deploys a round robin policy, this means that 8980 of the 8984 bytes of each IP packet end up on one link, while the other 4 bytes of each IP packet end up on the other link. This has been verified by other experiments, which showed the same Ethernet frame rate, but radical different bandwidth on each link. Probably, the additional overhead for fragmentation does not outweigh the reduced overhead due to larger packet size for this IP packet sizes which are only marginally larger than the Ethernet segment size.

The bottom graph in figure 8 shows the result of sending TCP traffic from server A to server B. This TCP throughput is much better than what is achieved with UDP. The traffic rate does not decrease for large socket buffer and packet sizes. In fact, it it around 9 Gbit/s whenever the socket buffer size is larger than about 500 kB.

Figure 9 shows the same result but now in the direction of server B to server A. When comparing the results presented in figure 9 to those in figure 8 it is clear that the results for traffic from server B to server A are much better than the results for traffic from server A to server B. Server A is the faster of the two servers. The better results could be explained by the assumption that the packet receiving process on an Ethernet channel needs more resources than the packet sending process on an Ethernet channel. This is probably the case because reassembly must be done at the receiving end.

There is still a drop in throughput when sending from server B to server A for larger packet sizes. This can be explained by the fact that larger IP packets are fragmented into more Ethernet frames than smaller IP packets. So the Ethernet frame rate is larger when using larger IP packets and thus put more stress on the receiving server which needs to reassemble all those frames into IP packets again.

3.4 Ethernet Channel with Two 10GE Links of Different Lengths

Figure 10 show the experimental setup where two links to unequal length are bonded. Figures 11 and 12 respectively show the results of the tests from server A to server B and from server B to server A.

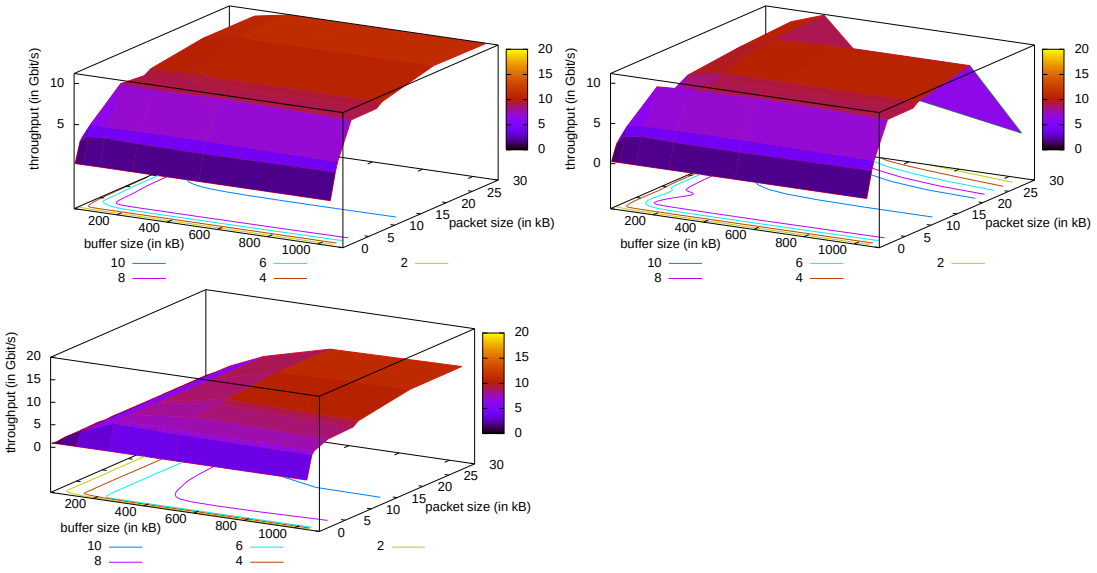

Fig. 9. Traffic from server B to server A over a 2x 10GE Ethernet channel (top left: UDP sending rate; top right: UDP receiving rate; bottom: TCP throughput

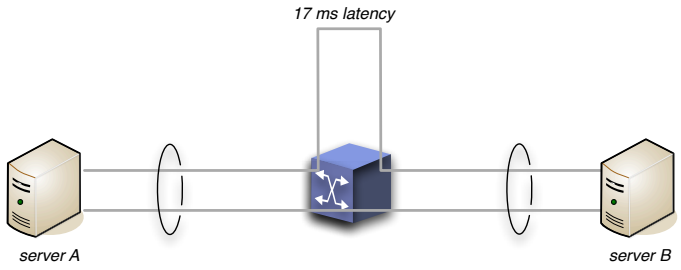

Fig. 10. Experimental setup for bonding of unequal length links.

The sending side for the UDP traffic (top left in figures 11 and 12) show a ridge near an IP MTU of 9000 bytes, similar to the result with bonding of equal length links (figure 3). As with that test, the ridge can be explained by fragmentation and uneven distribution of data on the two links.

Similarly, the results show a clear drop in performance with near 100% packet loss above a certain throughput of the sender.

The performance of TCP has collapsed due to the difference in round trip time. The throughput from server A to server B is less then 1 Gb/s, and the throughput from server B to server A is negligible.

The results were not expected. For small IP packet sizes, there is no fragmentation on the Ethernet layer, and the difference in latency accumulated to packet reordering at the IP layer. For larger IP packet sizes, there is fragmentation, and the difference in latency should already be handled during reassembly of the packets at the Ethernet layer (and would not result in packet reordering on the IP layer). However, there is no difference in the results with and without fragmentation. Instead, the only parameter that influences the performance is the buffer size.

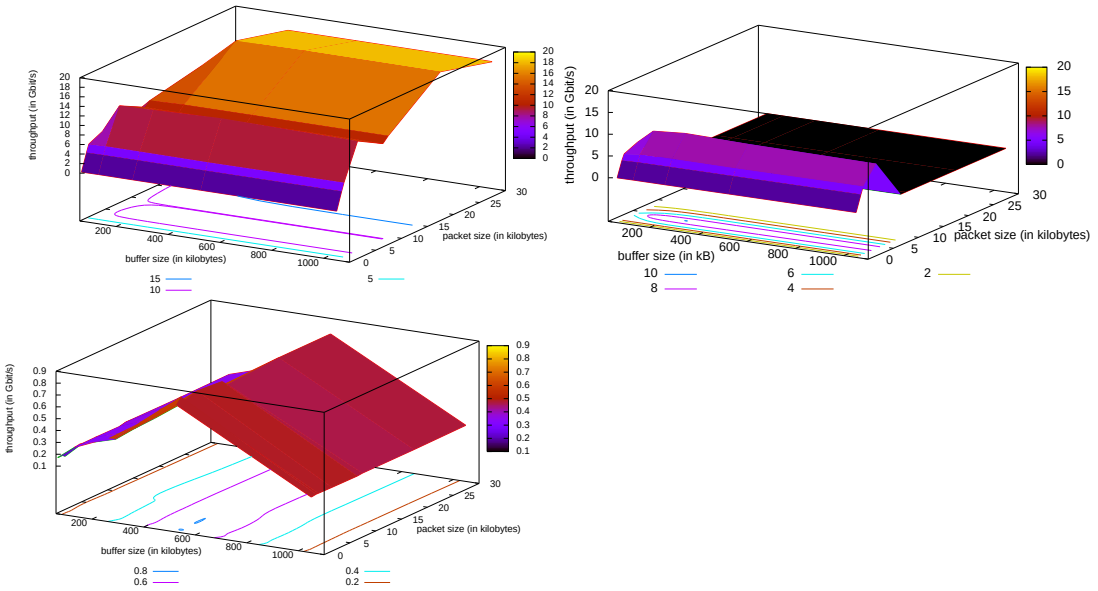

Fig. 11. Traffic from server A to server B over a 2x 10GE Ethernet channel (top left: UDP sending rate; top right: UDP receiving rate; bottom: TCP throughput)

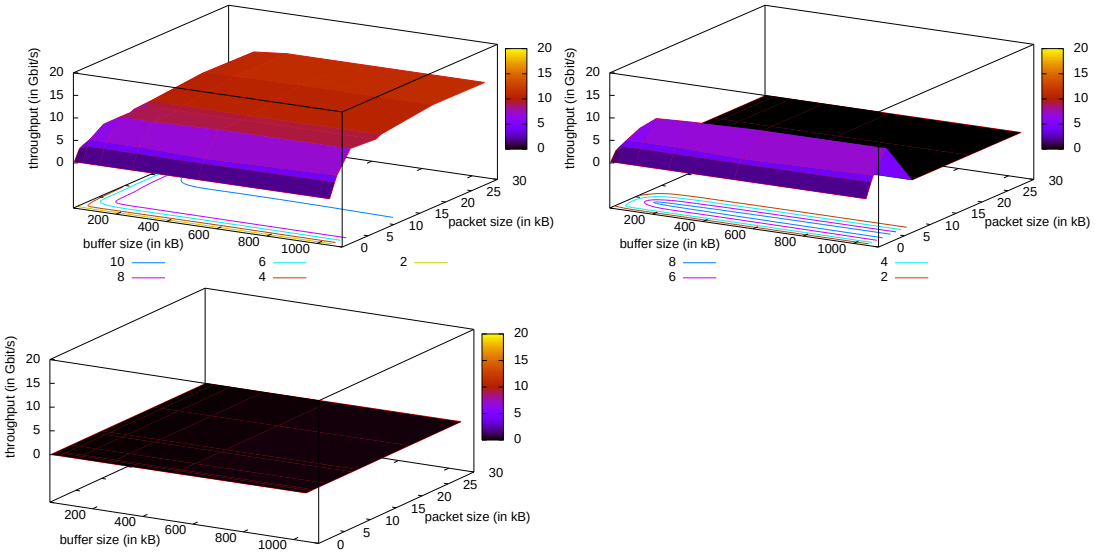

Fig. 12. Traffic from server B to server A over a 2x 10GE Ethernet channel (top left: UDP sending rate; top right: UDP receiving rate; bottom: TCP throughput)

For TCP, the buffer needs to be at least the round trip time times the throughput. For 20 Gb/s, this would be 20 Gb/s $\times$ 34 ms = 85 MByte. Since the buffer sized used in this experiment are much smaller than that, it is expected that this causes the poor throughput. This does not explain the collapse in performance for a buffer size larger than 800 kilobytes. Further examination is required to understand the cause of these effects.

4 Conclusions

The measurements shown in this document show that saturating a single 10GE link is possible. However, saturating an Ethernet channel of two 10GE links proves to be much harder.

The measurements done for this document provided a huge amount of information and insight about the various buffers, tuning parameters, hardware implementations, etc. that influence the end-to-end performance.

The reason for not being able to reach 20 Gbit/s is not clear. Further investigations are required to examine the exact cause (in the driver, kernel or application) of the packet loss and decreased performance above a certain socket buffer size or packet size. A main reason for limited performance is probably the limited capacity of server B. Further investigations will be done with faster hardware.

References

1. Ronald van der Pol and Freek Dijkstra, D1.1: Server Scalibility
   `https://noc.sara.nl/nrg/publications/RoN2010-D1.1.pdf`
2. Dell PowerEdge T610 Technical Guidebook
   `https://noc.sara.nl/wiki/images/6/62/Server-poweredge-t610-tech-guidebook.pdf`
3. Intel 5520 Chipset and Intel 5500 Chipset Datasheet `http://www.intel.com/Assets/PDF/datasheet/321328.pdf`
4. Intel Ethernet Server and Desktop PCI Express Adapters
   `http://download.intel.com/network/connectivity/products/prodbrf/252454.pdf`