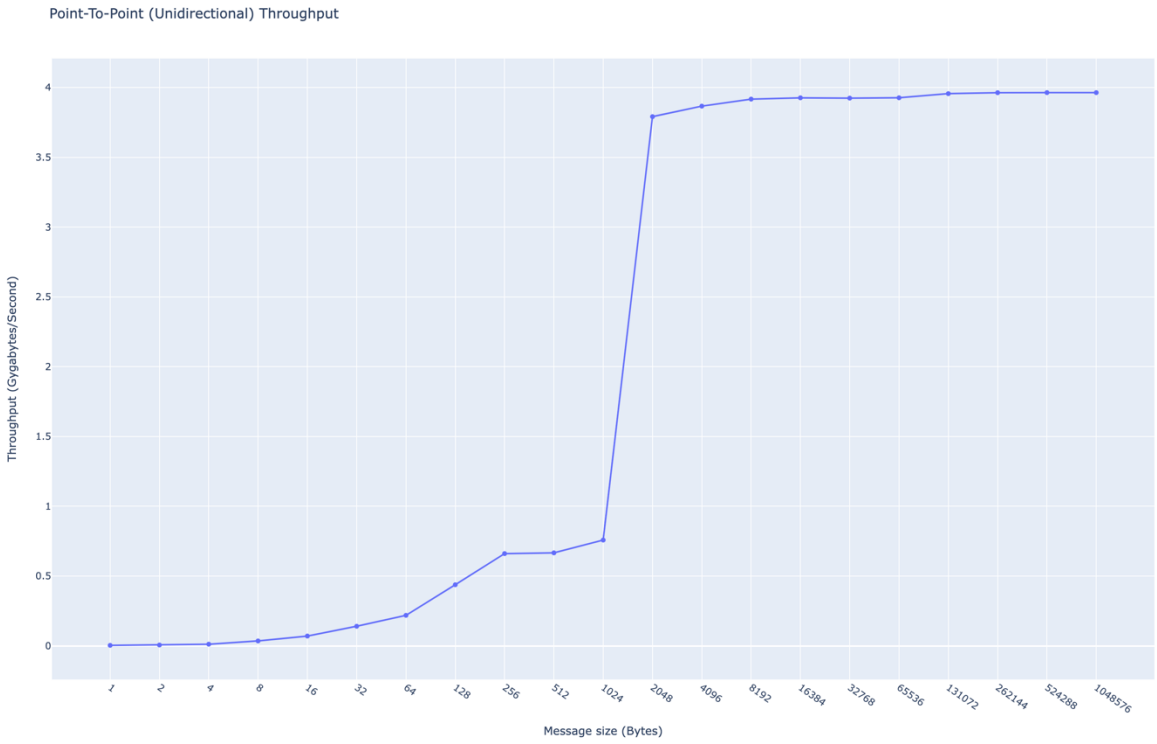
Exercise 2 Workshop In Communication Networks - Verbs API Throughput

Authors: *Yosef Edery* and *Leon Gershko*

The goal of the implemented code was to apply the Verbs API for to measure point-to-point (unidirectional) throughput between two machines for an exponential series of message sizes, ranging from 1 byte to 1MB.

The performance results were as follows:

```
1        0.003560        Gigabytes/Second
2        0.006076        Gigabytes/Second
4        0.011888        Gigabytes/Second
8        0.035161        Gigabytes/Second
16       0.070242        Gigabytes/Second
32       0.140365        Gigabytes/Second
64       0.218717        Gigabytes/Second
128      0.437915        Gigabytes/Second
256      0.660842        Gigabytes/Second
512      0.666309        Gigabytes/Second
1024     0.757967        Gigabytes/Second
2048     3.791192        Gigabytes/Second
4096     3.866430        Gigabytes/Second
8192     3.916728        Gigabytes/Second
16384    3.926040        Gigabytes/Second
32768    3.923778        Gigabytes/Second
65536    3.926328        Gigabytes/Second
131072   3.955934        Gigabytes/Second
262144   3.962851        Gigabytes/Second
524288   3.963447        Gigabytes/Second
1048576  3.963273        Gigabytes/Second
```



To measure throughput, we have set: the following values:

- 50,000 iterations per message
- 5,000 Work Requests / Queue (both send and receive queue)
- 5,000 warm-up iterations

As we can see from the graph, we obtained a convergence to 4 gigabytes/second (exactly 3.963273 for a size of 1Mb).
We can also appreciate how for message sizes below 2048 bytes, the throughput remains relatively low, but it jumps significantly for larger message sizes, especially at 2048 bytes. This is because, for smaller message sizes, the communication overhead dominates the time required for sending the message, which can limit the overall throughput.
Using TCP sockets, we obtained a pick of 117 bytes/microsecond, that means that with RDMA verbs we got more than **33 times better performance than using TCP sockets**.