Exercise #3: Low-Latency Protocols

General

The goal of this exercise is to compare the Rendezvous protocol to a trivial "eager" case.
Please refer to Lecture #2 for additional information on Eager vs. Rendezvous.

Write a C/C++ single-threaded application with Verbs API: The processes (at least 2, test with 2 and 4) connects in a ring to implement the following collective communication functions:

1. Reduce Scatter
2. All Gather
3. All Reduce

```
int connect_process_group(char *servername, void **pg_handle);
/*Connect processes*/
int pg_all_reduce(void *sendbuf, void *recvbuf, int count,
                        DATATYPE datatype, OPERATION op, void
*pg_handle);
int pg_close(void *pg_handle); /* Destroys the QP */
```

*Note that `pg_handle` is a struct you define, create and use with every API call.

- One CAN use RDMA Write with Immediate for signalling completion
- Implement pipelining to overlap communication and computation
- Use RDMA Read or RDMA Write for large messages zero copy on the all-gather phase