

Client Code

```
/* measure monotonic time */
uint64_t delta;
struct timespec start, end;
clock_gettime(CLOCK_MONOTONIC, &start); /* mark start time */

/* send test buffer */
while (so_far < TEST_BUF_SIZE) {
    ret = send(server_fd, tx_buffer + so_far, TEST_BUF_SIZE - so_far, 0);
    if (0 > ret) {
        printf("Error: send failed with ret %d and errno %d \n", ret, errno);
        return -ret;
    }
    so_far += ret;
    printf("\t [send loop] %d bytes, looping again, so_far %d target %d \n", ret, so_far, TEST_BUF_SIZE);
}

printf("OK: buffer sent successfully \n");
printf("OK: waiting to receive data \n");
// receive test buffer
so_far = 0;
while (so_far < TEST_BUF_SIZE) {
    ret = recv(server_fd, rx_buffer + so_far, TEST_BUF_SIZE - so_far, 0);
    if (0 > ret) {
        printf("Error: recv failed with ret %d and errno %d \n", ret, errno);
        return -ret;
    }
    so_far += ret;
    printf("\t [receive loop] %d bytes, looping again, so_far %d target %d \n", ret, so_far, TEST_BUF_SIZE);
}
clock_gettime(CLOCK_MONOTONIC, &end); /* mark the end time */

delta = BILLION * (end.tv_sec - start.tv_sec) + end.tv_nsec - start.tv_nsec;
printf("<<BENCHMARK>> %llu\n", (long long unsigned int) delta);
```