**MessageProcessor module**

**Assumption :** Current solution for the MessageProcessor module can process all sequence numbers in memory. If all sequence numbers would not fit in memory so we could use external sort in future.

**Note: Currently class interfaces are not used due lack of time in future we can design classes with interface.**



**MessageProcessor:**

* Read files name from directory and put them into the file queue.
* Start N thread for processing UDP messages and track receiving rate thread.
* SortFile reads UDP from the file and puts the sequence number into a tree map(seq\_num, UDP offset, size, file\_name).
* Once all UDP processing is done, create an output file from the tree map.

**Directory Reader:** Read files name from directory and put them into the file queue.

**Process UDP thread :** Pop file from a file queue and process UDP messages and put **sequence number** into a tree map(seq\_num, UDP offset, size, file\_name).Thread stops once all files are processed.

**CreateoutFile :** Read sequence number in an order from sequence number map and read UDP message from input file from the offset and write the UPD message into outfile.

**Track receiving rate thread :** Control message processing rate**.**

**Testing**

**createFile :** Creating **N** input files which have UDP messages with duplicate packets.

**ReadOutFile :** Reading output file and printing sequence number and size.

**Intput folder :** All input files.

**Out folder :** Ordered UDP message file.

**Test\_output.txt :** Test output file**.**