Team Rap: Anna Blume (201907691), Mathias Weller (201907948) & Rasmus Jørgensen (201909451)

Hand-in 1 (DISSY)

Testing:
We have tested the system manually by opening peers in different terminals, writing messages, and checking that they show up at every other peer as expected. We have checked a chain consisting of 3 peers and a branch where 2 peers connect to the same peer. We have also checked that the peers don't resend a message which has been sent before, and that a new peer gets all the messages already sent on the network. Additionally, we have tried running peers on two different computers and confirmed that this also works as expected. We have tested concurrency by spamming identical messages on two peers both connected to a third and confirming that the message is still only sent once from the third peer.
Additionally, we have refactored the system to make use of a peer struct with a run method, and interfaces for some of the methods, making it possible to isolate and test separate methods in the future.

Eventual consistency - all clients receive all messages:
Our system sends all messages via TCP, which guarantees that messages are eventually delivered. When a peer (a) is started it adds the peer that it connected to (out_conn) to its set of connections. Whenever a new peer connects to a, a also adds that new peer (in_conn) to its set. Whenever a message arrives at a peer that has not seen the message before (which it keeps track of in the messagesSent set) it broadcasts to all the connections in its set of connections. A message can arrive through both the command line (input by the user) or via one of the connections the peer has. These are treated the same and printed if it is a new message (if it's not present in the messagesSent map). Since new peers that are added to the network always end up in another peer's connections set it will always get any new messages. Old messages are sent to the newly connected peer by the peer that it connects to. This "old" peer simply sends all the messages in its messagesSent set to the new peer. This means that, if no peers leave the system so it becomes unconnected, once no new messages are input into the system, all the peers will eventually get all the messages.