UDP Implementation on BitBox

In Project1, every communication between BitBox peers are TCP based, but here an UDP based BitBox are implemented. Compared with TCP, UDP is connection-less and unreliable, making the implementation very different from the TCP version.

First of all, for a file sharing system based on UDP, an application layer reliable implementation is necessary. Here we introduced the retry and timeout, as if a request do not acknowledge it’s corresponding response within the timeout limit, it will be retransmitted with a number of retry times. However, this is not enough for a reliable file transfer. Since there are also sequences for the events, such as the create of directory should be executed before the creation of files in that directory. But we didn’t define sequence numbers in our implementation so such situations will only be handled by the synchronizing of events every certain time. But this is not a complete solution for all cases. Similar situations also happen with file bytes transmission, but cannot be resolved by synchronizing events. If some file byte is lost during transmission of all retried times, the file transmission will be failed with a temperary (bitbox) file, which causes errors for further synchronizing events of the same file create, making the file in a failure until the temporary file is deleted.

Secondly, we didn’t implement segmentation or window size in the application layer for UDP, therefore the packets size is set to be below 8192 in UDP mode. Besides, the timeout used in UDP mode is a fixed value, instead of a changeable RTT as in TCP mode, hence making the re-transmission not in its most efficiency.

Also, the protocols defined in TCP mode do not suit well for UDP case. For example, the handshake process will be different as handshake after success handshake will no longer get an invalid protocol response, as the handshake response may be lost during transmission and the handshaking peer will send it once again.

In general, UDP lacks the reliable functionality of TCP, but it is generally more efficient in transmission under a good connection environment. Also the UDP sends and receives all packets from same socket, but TCP holds a sockets for each connection to a peer, hence UDP will be more resource efficient.