5. Verification

To verify the functionalities of the system, we designed several test cases.

Test#1: To verify that the connection between Peers succeeds. We input connect order to each Peer after they have been initialized. The signal "Connection established!" is presented if this operation is completed.

Test#2: Verify Query functionality with single neighbor. Put a unique file (e.g. test.txt) in Peer 2, which is one of Peer 1's neighbors. Peer 1 activates Query(test.txt) to see if it receives a hit-query message of test.txt from Peer 2.

Input: 2. Query;
test.txt

Output: Peer 2 has this file

Test#3: Verify Query functionality with many neighbors. Put a unique file (test.txt) in folder of Peer 1's neighbors (Peer2, Peer 3, Peer 5). Peer 1 activates Query(test.txt) to see if it receives a hit-query message from its neighbors.

Input: 2. Query;
test.txt

Output: Peer 2 has this file
Peer 3 has this file
Peer 5 has this file

Test#4: Verify Query functionality in Tree architecture. Create a tree, making Peer 1 as the root and Peer 10 as one of leaf node. The depth of this tree is unknown. Put a unique file (test.txt) in Peer 10. Peer 1 activates Query(test.txt) to see if it receives a hit-query message from Peer 10.

Input: 2. Query;
test.txt

Output: Peer 10 has this file

Test#5: Verify Query functionality in Star architecture. Set Peer 1 as the central node, while other Peers only connect to Peer 1 and there is no connection between each other. Suppose Peer 2 and Peer 10 has the same unique file (test.txt). Peer 1 activates Query(test.txt) to see if it receives a hit-query message from these two nodes.

Input: 2. Query;
test.txt

Output: Peer 2 has this file
Peer 10 has this file

Test#6: Verify Obtain functionality with single neighbor. After operating the Query action, we can use Obtain order to choose one of Peers, which have the files we wanted. In fact, Obtain is an action sending file delivery request. Here, we use Peer 1 to obtain test.txt from Peer 2, as the situation described in Test#2.

Input: 3. Obtain;
Peer 2
Output: Download succeed

Test#7: Verify Obtain functionality with many neighbors. Similar as Test#6, but we firstly do the Query like Test#3. Then we can download the same file from different Peers.

Input: 3. Obtain;
Peer 2
Output: Download succeed

Input: 3. Obtain;
Peer 3
Output: Download succeed

Input: 3. Obtain;
Peer 5
Output: Download succeed

Test#8: Test querying non-exist file. Peer 1 activates Query(x.txt), assuming x.txt does not exist.

Input: 2. Query;
x.txt
Output: No such file exist

Test#9: Test obtaining file from non-exist Peer. Do the same query as Test#3 and then Peer 1 activates Obtain operation and inputs Peer x to see what will happen.

Input: 3. Obtain;
Peer x
Output: No such Peer exist