When I saw the problem I had no idea what to do. So I decided to go through the original codes and the examples. After I reviewed the original codes I understood the basic idea of them. Then I went through the first example Calculator. This example helped me to recall the basic concepts of AspectJ. The second example makes more sense to me. I realized I just need to focus on SocketChannel.write() and SocketChannel.read(). When the SocketChannel.write() is called, the Client is going to send the message. So we need to record the time of that moment, this is the start time. Similarly, when the SocketChannel.read() is called, the Client is going to receive the message. We also record the time of that moment, this is the end time. Then the difference of the two is the time we need. So I implemented the aspect in this way. But after talked with Ali, I realized that I made I mistake there. Actually my aspect just keeps track sending and receiving, we don't know if the received message is the reply to a specific request. Since it is asychronous to Client, it is possible that the reply to the second request can come before the reply to the first request comes. We need a data structure to record the Id of the request messager. We do subtraction only when the corresponding reply is received by the Client. In order to do this, I made a little change to the code of Server part. When the Server creates the reply, we add the request message's id to its responseId. So that when the Client receives this reply, it knows which request it was replying.