Project 3 SKJ

Oktaviani Szamocka s17874

1. Files

This project is consisted of a few classes

1. Agent : this class is where the agent make connection TCP with relay
2. AgentUDPlisten : class that making new thread and where the UDP socket of agent run
3. ClientProcess : where the client process have run and connect with UDP Agent socket
4. Main : the class to create object of agent
5. Relay : where the relay make TCP and UDP connection to agent and to target
6. RelayThread : class that help the process of Relay
7. Message = class that going to store message between Relay and Agent (for TCP)
8. TargetUDP : class that target of message
9. Implementation
10. First we need to run **Relay class.** Then Relay run TCP socket at port 6666 (hard-coded) and waiting for agent to connect with.
11. After that we need to run **Main class** to run the agent.
12. In the Main we can see, IP of relay and IP of the target is both local host. The port of relay is permanently coded and agent will open UDP socket at port 1234, and 5678
13. After connected to relay, agent send the configuration of IP of the target of the packet and IP that will be sent the data. These two IP is basically the same which is the target of packet

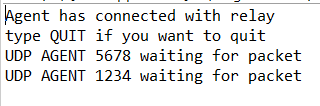
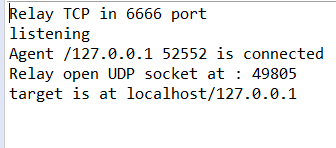


Figure 1 Agent illustration after configuration Figure 2 Relay illustration after configuration

1. After that we need to run TargetUDP class. TargetUDP is the class that later going to accept and reply back the message. UPD socket is opened in random port
2. We need to run the ClientProccess class. This class will represent the client process of agent. The UDP port will be open in random number. It will ask to fill which UDP agent port want to connect with and the udp port of the destination

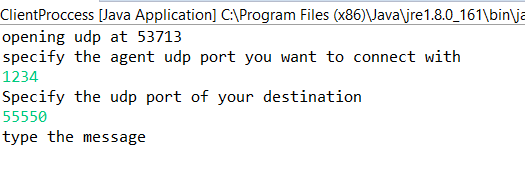
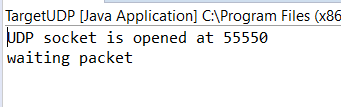


Figure ClientProccess and targetUDPillustration

1. Then clientProccess can write message to be sent to the target. The format UDP message that client write is targetport-message.
2. When agent receipt packet from their UDP socket, it save the packet port of clientproccess to know where to send the reply later (one UDP – one clientproccess). Agent also split the message and substract the target port to be parameter targetPort of object message. Besides that it also attach its udp socket so relay know from which udp port the message coming.

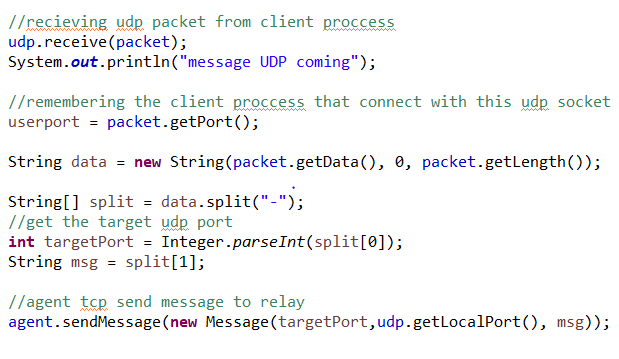


Figure 4 the code to accept UDP packet and convert to TCP message in agent

1. Then relay accept TCP message. Relay will manipulate the message to be in proper format, agentUDPPort-message. Then send the message by UDP

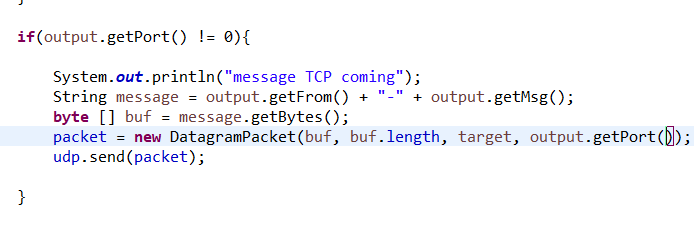
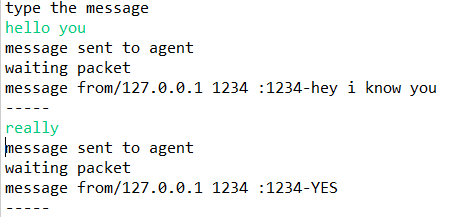
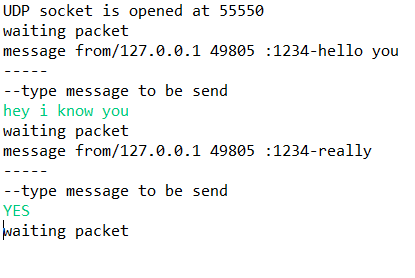
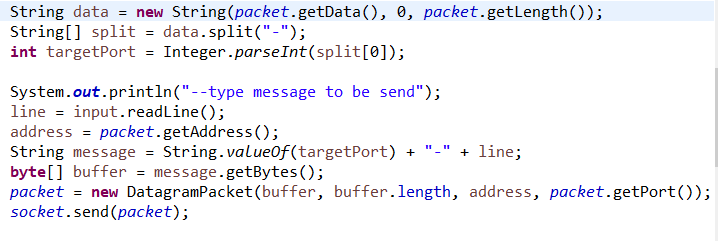
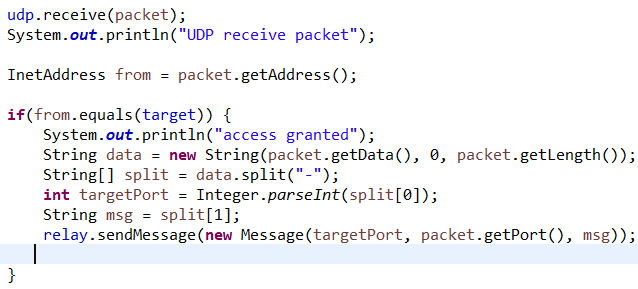


Figure 5 Relay code when converting TCP message to UDP

1. Then the target accept the packet and could reply back. Here example of conversation between targetUDP and clientproccess
2. When targetUDP reply the message, the process is quite similar. The message format will be targetPort (agent udp port) – message. Then packet is sent to Relay UDP port. Here is the code



1. Then Relay will accept the packet and will check whether the IP of sender is equal or not with the IP configuration. If yes then will process the packet and change it into TCP message



1. Then the message will be arrived at agent, then agent convert the message into datagram and send it to udp port of client process
2. Besides that, we can make multi client process working together in same time

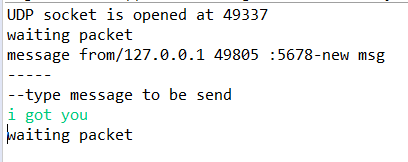
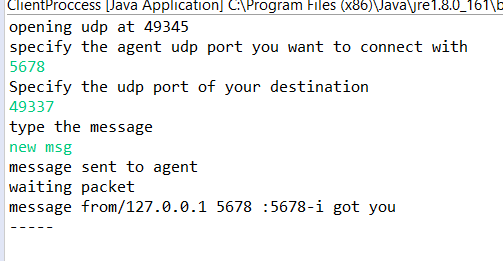


Figure 6 another client proccess connect run together with the other one

1. Agent can disconnect the connection by typing “QUI” in Main console. After that relay can listen to another agent

