CHAPTER 11 – ASSESSMENT EVALUATION

1. Introduction

The testing of this project was performed in the network of Computing and Information Sciences at Kansas State University. The network is an open network that connects to the Internet, and students and faculty use this network together. Since this project involves communication of multiagent systems, the network is very crucial to the testing. The test used three kinds of operating systems that available in the network, Windows XP professional, Debian Linux and Sun Solaris 9. The tests were performed on both wired and wireless network. The tests were also performed during the nighttime to minimize the effect of network traffic. Furthermore, the testing was not the test of network traffic, so the size of messages was limited to 500 bytes and the maximum numbers of continuous sent (send messages without waiting for a reply) are limited to 100 messages each time. With these limits, we can guarantee that the network will not overload.

2. Test cases overview

Test cases	Scenario
1 Sending and receiving unicast message	1. one agent sends 100 unicst messages continuously to another agent.
	2. one agent sends unicast message and another agent reply with the same message 100 cycles.
2 Sending and receiving multicast message	1. one agent sends 100 multicast messages continuously to the group of two agents.
	2. one agent sends multicast message to the group of two agents, and then one of the two reply to the group with the same multicast message 100 cycles.
3 Sending and receiving broadcast message	1. one agent sends 100 broadcast messages continuously to two agents on the same network.
	2. one agent sends broadcast messages to two agents on the same network, and then one of the two reply the same broadcast message 100 cycles.
4 Encrypting and decrypting message	1. encrypt the Message object and then decrypt the Message object.
5 Subscribe to multiple group	1. one agent subscribes to three groups, another agent sends 30 multicast messages continuously per group to these three groups.
6 Sending and receiving secured unicast message	1. one agent sends 100 secured unicast messages continuously to another agent.
	2. one agent sends secured unicast message and another agent reply with the same message 100 cycles.
7 Sending and receiving secured multicast message	1. one agent sends 100 secured multicast messages continuously to the group of two agents.

	2. one agent sends secured multicast message to the group of two agents, and then one of the two reply to the group with the same secured multicast message 100 cycles.
8 Compatible with existing multiagent systems	1. Run the supplied multiagent systems built based on agentMom1.2. to check the compatibility
9 Test all the features by building multiagent systems.	1. The multiagent systems is built to check the functionality of the new agentMom package. For more detail of the scenario, please refer to the analysis section.

3. Testing Result

Test cases	Criteria	Results
1	All messages must be received.	Both scenarios are passed.
2	All messages must be received.	Less than 1% of message got lost.
3	All messages must be received.	Approximately 1% of message got lost.
4	Encrypted message must be unreadable and decrypted message must be the same as original message.	Passed.
5	All messages must be received.	Less than 1% of message got lost.
6	All messages must be received.	Both scenarios are passed.
7	All messages must be received.	Less than 1% of message got lost.
8	The supplied systems must be able to run as same as when running on agentMom1.2	Passed.
9	The built systems must be able to run as	Passed.

4. Analysis

Although, the testing was not fully passed, all the requirements are satisfied. The error rate of unreceived messages is 1% at the most for each type of communication. The lost messages are not to be the fault of the product, but the fault of the network. Packets are dropped by the network for many reasons as described below:

1. Network environment – the network environment is the most important factor in this testing. Network environment in this case includes the setup of the network, wired or wireless, number of clients in the network, the traffic and the maximum allowable of packet size. If the network is wireless, the chance of lost packet increases. If the network is busy, the chance of lost packet increases. Also, the packet will be dropped if the packet size is bigger than the allowable size. For example, most Ethernet networks set the maximum size of 1500 bytes.
2. Unreliable Protocol – as we see in the testing results, messages were lost when using multicast and broadcast type of communications. Since multicast and broadcast use datagram packet that does not guarantee of delivery, message can

easily get lost. Unlike TCP/IP, unicast communication is reliable and delivery is guaranteed. I also performed the testing during the daytime once, the results were that the packets got lost about 25%-40% due to high traffic in the network.

3. Operating Systems – the operating systems also plays a major part in this problem because different operating systems have different setting in the maximum buffer size of received data. Also, the buffer size is limited. If the buffer is full, the packet will be dropped. The processing speed may also affect this problem.
4. Router – problems from the router is the same as in operating systems. They have different size of buffer, and buffer is limited. Moreover, some router may decline to forward any multicast and/or broadcast message. It depends on the setting of the router. As far as I know, many inexpensive routers does not support multicast. This type of router broadcast the message to all clients instead of forwarding to subscribed clients.

Furthermore, the multiagent system was built to test all the functionalities provided in new agentMom. The goal of this system is to distribute the secret key to other agents in the environment so that theses agents can communicate using secured multicast communication. The system consists of one agent with the private key and three agents waiting for the private key. The scenario starts by the agent with the key sends broadcast message asking for the IP address of all the agents in the same network. When message is received, the agents send reply using broadcast message with their IP address. Then, the agent with the key sends the key to all agents based on the reply message using secured unicast communication using SSL Socket technology. Finally, all the agents can then use the secured multicast message using the received key to send and receive messages. With this scenario setup and the other testing results, all the requirements are satisfied.

In conclusion, the testing new agentMom package was quite difficult for two main reasons. The first one is the network used to perform the tests was an open network. There were many factors that could affect the results as described above. The second problem was that fact that agentMom is a framework that requires a real implementation of multiagent systems to be able to run to perform the tests. Hence, it difficult sometimes to determine the source of problems whether they are from the agentMom package or the implementation of the systems.