Jose Arias
Discussion Questions Project 1
CS160

1. With event driven programming we are certain that part of our code will be executed only when the event takes place. So we can expect how the program will behave. However, if the events start to happen at the same time, we will get undesired behaviour. Therefore, it is critical to use different timers so the events do not overlap and interfere with the sequence of our code.
2. By having the two mechanisms in place, we have two mechanisms to check for a packet. Hence, we assure that the packet will not circulate indefinitely in the network. If we only had flooding checks, we would need to check the TTL every time the message arrives at a new node. When the TTL is zero, the packet is dropped. If we only had TTL checks, it would be possible for a package to be rebroadcast again and again. As a result, the package will circulate indefinitely in the network.
3. In the best case, assuming a simple line. A node would see a package at most two times. That is the time it first sent the message and another time when it received the message. By checking the cache, the node will drop the packet because it knows it has seen the package before. In the worst case, a node will be connected to multiple nodes. So the number of times a node will see/receive a message will be n-1 times.
4. Send the packets to the neighbor directly instead of broadcasting the message to all the neighbors.
5. Having a separate module and configuration file for the neighbor discovery and flooding task. I am starting to see how the code can become unmanageable by not having different modules. Modularization is something that I should work on for future projects. As far as the skeleton code, I think it was well structured.