MECAC WEEKLY REPORT (April 15 – April 21)

Having setup the svn server for the MMOG, this we have worked separately on a number of issues. Utmost attention has been given to beta testing of Virtual Turkey. Beta testing this software involves installing at least 700MB of a client copy to different computers at MODSIMMER. We have first tried to manually copy the executables. As a result, we were unable to relocate the binaries as the source codes involved references to files with absolute paths. In order to mobilize binaries, we have refactored the source codes to contain relative file paths. Following sound effect file reference is given as an absolute path. We have changed this and many more file references to mobilize binaries.

```
(@"C:/Documents and Settings/117-4/Desktop/proje/StajOyun/StajOyun/Client/Songs/walking_running/Walking on Concrete 2")
```

After recompiling the client package with refactored source codes, we were able to relocate the client package to different computers at Modsimmer. Currently, we have installed the client package to two other computers within the Modsimmer network. To simulate real world MMOG experience, we have also installed the client package to our personal computers. We have realized that the system requirements for Virtual Turkey makes it unfeasible to play with 2GHZ CPU laptop. Further examination of CPU usage and memory footprint of client packages is needed to decrease the system requirements.

Umit has worked on client package as well. During our tests with the clients while listening the network with WireShark, we have noticed that the clients periodically sends position updates even when there is no position change. To verify this issue, Umit has examined the client source code, and realized that the client has a separete thread which sends the position update periodically. Following code taken from client package illustrates the issue more clearly;

```
protected override void Update(GameTime gameTime)
      UpdateNetwork();
```

Periodically, XNA engine sends an Update event to the client component. When the client component receives this event, it calls Update Network method of the client package to transmit its current position and receive position update packages. Note that this operation is done regardless the fact that the clients position has not changed in time. For an MMOG to support more clients concurrently, this approach obviously would result in bottleneck in the network traffic.

To resolve this issue, Umit has first separeted the logic for sending a position update and receiving opponents position updates within UpdateNetwork method of client component. He has then conditioned the position update to be sent when a position changes noticably. We have basically checked the distance between last position sent and the current position and thresholded this value.

After this modification to client component, we have analyzed the network traffic with WireShark again.The bandwidth usage of a single client dropped significantly, and the number of concurrent players that can be supported from a single server has increased.

As we are nearing the delivery date of the project, we have also considered possible deployment issues of Virtual Turkey. If the MMOG becomes popular among internet users, the administrators of it would need to deploy additional servers seamlessly. A simple proxy server can be used to channel traffic through different servers. In computer networks, a proxy server is a server (a computer system or an application) that acts as an intermediary for requests from clients seeking resources from other servers (1). A client connects to the proxy server, requesting some service, such as a file, connection, web page, or other resource, available from a different server. Where to deploy an additional server of Virtual Turkey is, on the other hand, not a trivial approach. We have conducted a research on network simulation to determine locations of potential servers. At first GtNets has seemed a good start point to discover network tophography. The Georgia Tech Network Simulator (GTNetS) is a full-featured network simulation environment that allows researchers in computer networks to study the behavior of moderate to large scale networks, under a variety of conditions (2).

However, GtNets needs deep integration with the code base of both client and server components and requires huge time which could easily put the project behind schedule. Instead Cinar has suggested a design for our own tool to discover network tophography from the server side only. The design didn't turn out to be huge challenge for MECAC. When an administrator clicks 'discover network tophograph' button on the server component, server executes following algorithm to trace the clients.

Algorithm 1. Network tophography discovery

Let G = (V,E) be a tophography graph
For each connected client represented by (ip:port)
    Execute tracert program from server by ip argument
    Let (R1,R2,...,RN) be a path P from server through client
    Merge G with P check for equality conditions (V.ip==P.Rx.ip)
Output G

Above algorithm can correctly determine network tophography and potentially usefull to find possible server locations. After G is obtained, we can output this information to the server administrator along with the bottleneck points. It remains as a future work to MECAC to implement this design by Cinar.

References

1) http://en.wikipedia.org/wiki/Proxy_server
2) http://www.ece.gatech.edu/research/labs/MANIACS/GTNetS/