# ECE1778 Final Project: Network Pulse – UI

Yiwei Pu (#998089839), Yuefei Zhu (#998230249)
Department of Electrical and Computer Engineering
University of Toronto
{*ypu,yuefei*}@*eecg.toronto.edu*

## I. Introduction

Measuring a network path's quality is important. The main task of network measuring is to probing some criteria of quality, *e.g.,* the end-to-end delay between server and clients, the current traffic conditions of the network, *etc*. The obtained performance information could be used to enhance user experiences not only as a feedback to adjust user's terminal device but also to help network operators to provide better service. Because of its practical significance, many research projects have been conducted to collect these network quality measurements. But, the results of most of these proposals are biased, since only single computer or a small set of probing nodes are involved in these experiments because of the limited experimental funding or equipments. It is fact that the more computers involved in the probing process, the more comprehensive the results will be. As a consequence, finding a large amount of available probing nodes is one of the key challenges in measuring the network quality.

Motivated by the fact that there are a large number of iPhone users, 21 million at present [1], and those users are distributed inherently across the world, we propose to conduct the probing on the platform of iPhone in this project. By installing our probing application named Internet Pulse, each iPhone holder becomes an active probing contributor. With the permission offered by users, Network Pulse will anonymously upload the probed results as well as the user's geographical location provided by iPhone to dedicated servers. After analyzing the collected data from all online users in our backend, Network Pulse will then feed the user the assembled data. By contributing the partial information probed by himself, the user is able to obtain a comprehensive status report of the global Internet at the moment.

However, probing and processing the statistics of network condition is not the whole story, for the simple reason that, among our potential users, most of them have no background in the area of network engineering. In other words, from their point of view, the comprehensive status report is tedious, the numerical results are meaningless, and the network condition is still mysterious. According to the analysis above, we base the motivation of our application on the fact that users are curious about the network they are accessing, but no application so far has been developed to present the concept of network condition. Therefore, the selling-point of our application is that it is the first application to help users to get insights of the network condition in such an easy way.

In short, the overall goal of our application is to present the network condition to users, no matter they are professionals or have no background in the area of networks. If he is professional in network engineering, he can collect explicit numerical results of measurements and export comprehensive data report to plain text files for further data process or analysis. If he has no expertise in related areas, this application will also present a vivid and attractive sight of the current network.

## II. Overall Design

Fig. 1 shows the overall structure of our app.

- Main Menu: When a user launches the app, a main menu contains three buttons shows up. Pushing the network button enters the main part of our app, which shows animations on a map and a 3D earth. The network details and information of the app can be shown with another two buttons.

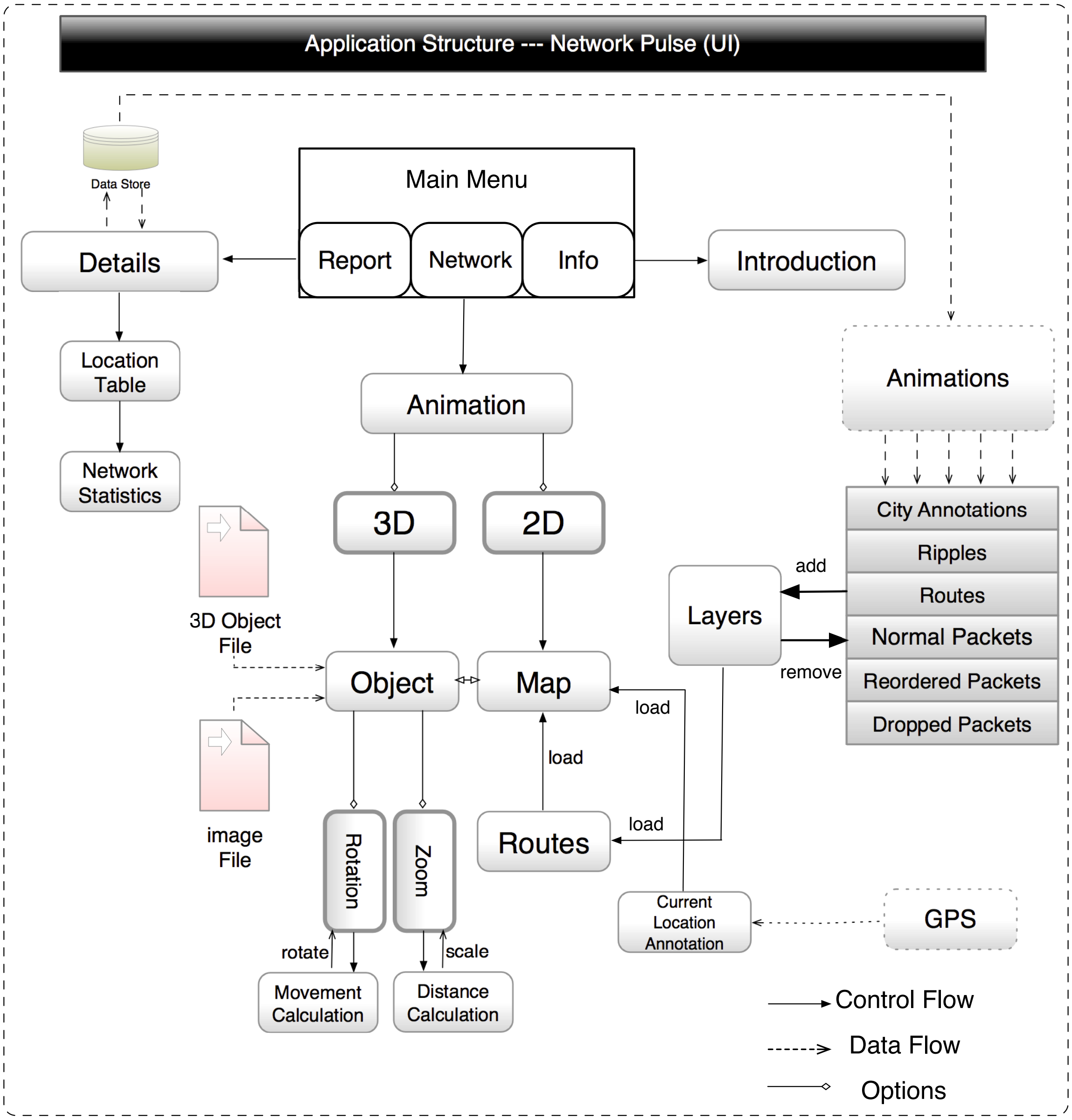
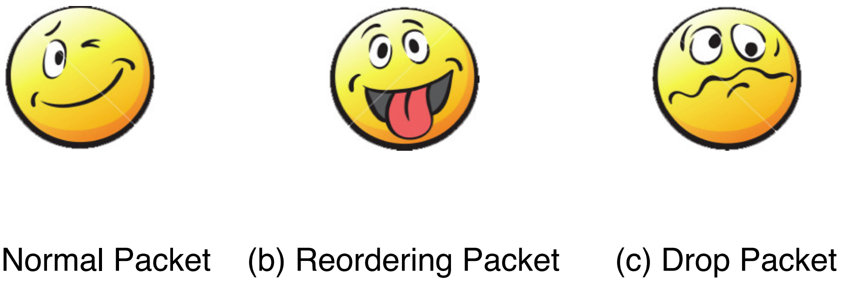


Fig. 1: A simple example in which the interferences and virtual bids are given.

- 2D: After the network button is pressed, a map is loaded, on which a mapView is added as one subview. The current location of the user is pin-annotated, which is added as another subview of the mapView. To develop animations, another view called “mapRouteLayerView” is designed, on which we dynamically add/remove different kinds of layers. Further, each kind of these layers contains one or two animations.
- 3D: With the navigation bar, a user can switch to a 3D earth. This 3D earth is drawn by *Blender* [2], and exported to a .obj file then transformed to a c-head file containing an array describing this 3D object. By importing this head file and a world map for texturing, a 3D earth can be loaded and manipulated in our app. Rotation and zoom are implemented for single touch and multi-touch correspondingly.
- Details: After the report button is pressed, a table containing different locations is shown. Selecting one location navigates to another table which shows the user the recorded delays, packet loss rate and packet reordering rate from the user’s current location to the selected location. These records,



(a) Normal Packet (b) Reordering Packet (c) Drop Packet

Fig. 3: Faces VS Packets

are also stored into the iPhone's file system.

- Introduction: An alert introducing this app will shown after the information button is pressed.

## III. Functionalities

In this section, we will introduce how the app works with different functionalities.

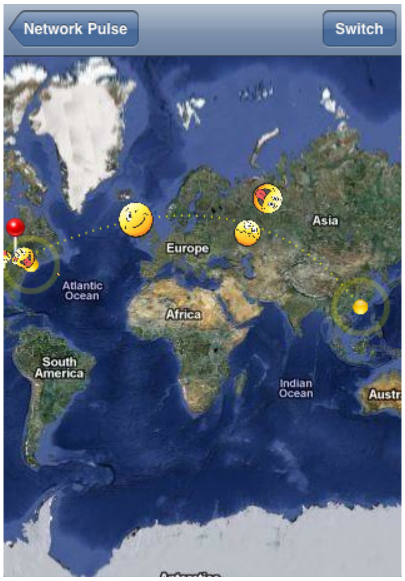


Fig. 2: Animations on a map

1) 2D animations: As a key functionality of this application, the task of adding animation is to present the condition of links between the user endpoint and remote destination servers. To implement this feature, our design is based on the Core Animation and layer design. Besides the technical implementation, the main challenge is to find out a proper way to express the explicit input experimental results, including 1) end-to-end delay, 2) packet loss rate, and 3) packet reordering rate, in an easy-to-understand way. Our solution is listed as follows.

(a) We create three types of facial expression to represent the three kinds of packets;
(b) For each class of packet, we design a moving path according to the specific characteristic of corresponding packet;
(c) The value of the delay is reflected as the duration of the movement of the normal packet;
(d) The packet loss rate and the packet reordering rate is proportional to the possibility of the occurrence of corresponding packets, which means the higher value of this measure is, the more chances of corresponding packets will appear.

For example, in Fig. 3.(a) a normal packet is presented as "happy Tom", who always keeps smiling and moving stably and smoothly from user endpoint to the location of remote servers, while in Fig. 3.(b) the "naughty Jack", who is making face, represents the reordering packet. On his way to destination, he prefers to walk around and, therefore, falls behind when he comes back. This is similar to the behaviour of real reordering packets; and in Fig. 3.(c) "sad Bob" is a spokesman of drop packet. He is so weak that he cannot reach the predefined destination just like the behaviour of packets dropped because of congestion in the network.

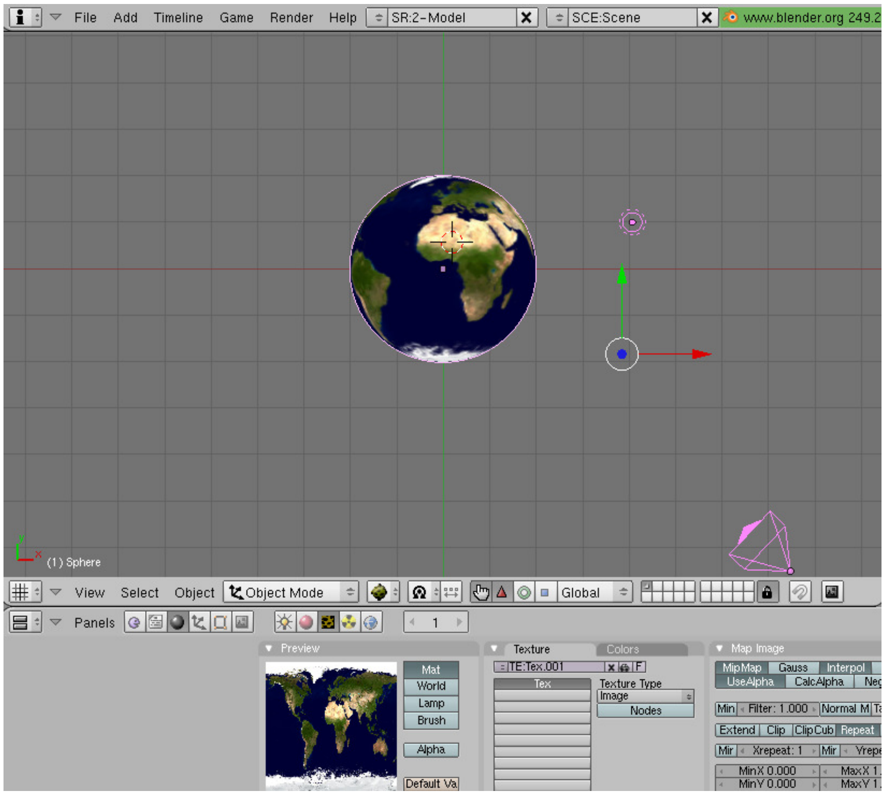
2) 3D earth implementation: Previously, another feature we proposed to accomplish was to have animations on a 3D earth. We planned to locate the user's current location on a self-designed and imported 3D earth, as well as to locate the locations of remote servers. Then we can implement similar animations with links and different kinds of moving facial expressions along the links. However, due to time limitation

and the difficulty of turning location information (i.e., longitude, latitude) to coordinates on a 3D sphere capable of rotation and scaling, we just implemented the rotation and scaling part.

We first design a 3D earth using Blender, which is shown in Fig. 4. Then with a script, we export it to a wavefront (.obj) file and then using a script to turn it into a c-head file containing an array describing this 3D object. Next, we can easily load and manipulate this object by OpenGL ES functions.

Specifically, we first use glVertexPointer(), glNarmalPointer(), glTexCoordPointer() and glDrawArrays() to load the object. Then we set up the lighting and texture a 512 × 512-pixel picture onto the sphere.

In this app, single touch and multitouch are for rotation and scaling the 3D earth respectively. For single touch, we calculate the moved distance of the finger and use the glRotatef() function to rotate. For multitouch (we only considered the two-finger case), the distance of two fingers is obtained and compared with the previous distance to get a scaling factor, and then the glScalef() function is called to dynamically scaling the object. The touch actions are all captured by the touchesBegan, touchesMoved and touchesEnded functions provided by the iPhone SDK.



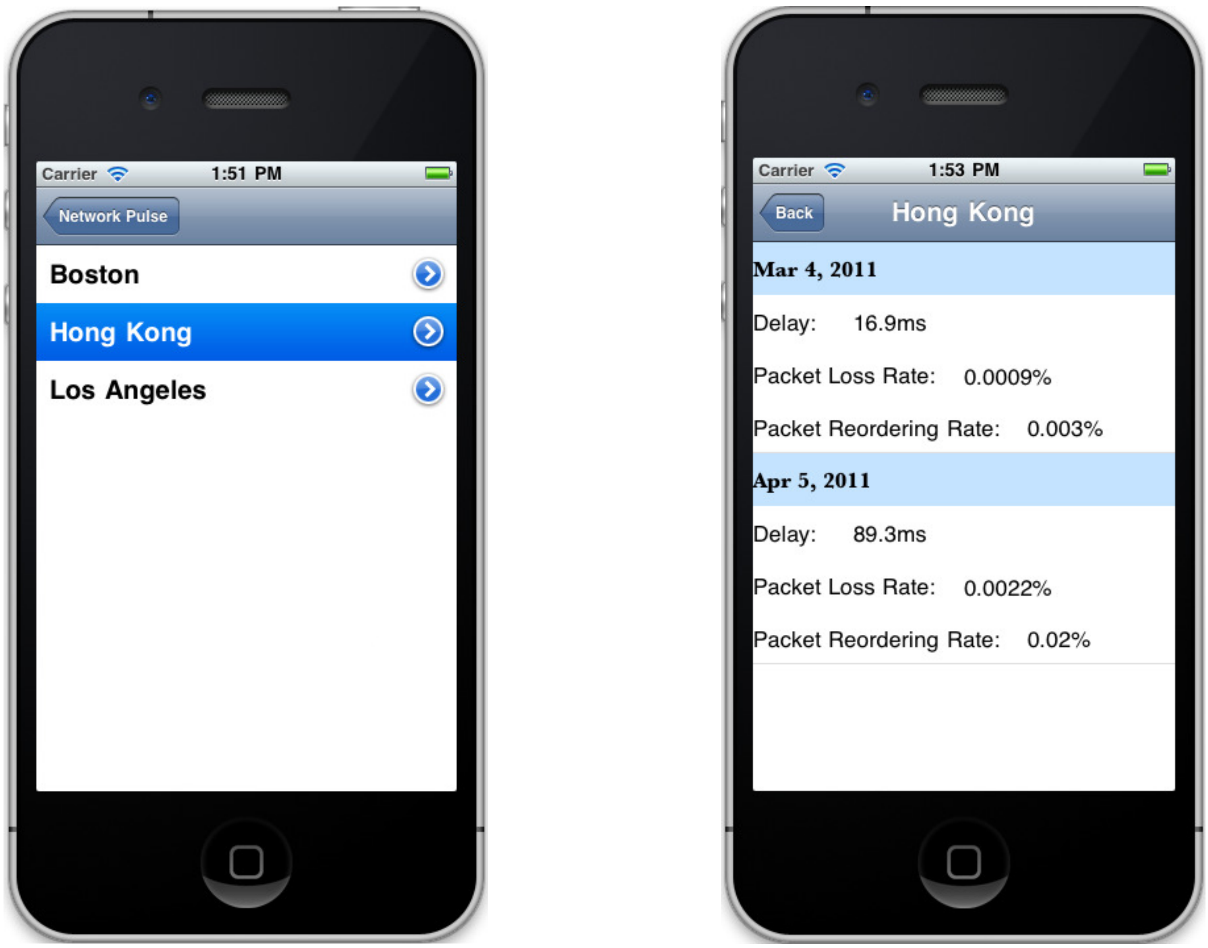
Object Mode
Global
1
TE:Tex.001
Texture Type: Image
Filter: 1.000
Xrepeat: 1
MinX 0.000
MinY 0.000

Fig. 4: Design a 3D earth with Blender

3) Detailed reports: For users who want to explore more in terms of the network environment statistics, they can switch to the tables containing the details, which is illustrated in Fig. 5. With such a hierarchical structure, users can pick one specific location and see the temporal changes of the network measurements.

## IV. Lessons Learnt

This course is rewarding. On one hand, we know how to develop applications on the platform of iphone and Xcode. Besides the topics in class, we have learnt the basic knowledge of Core Animation and OpenGL ES. Based on these knowledge, we are able to design and implement animations both in 2D and 3D spaces. On the other hand, we learnt how to manage a project. We have experienced how to set milestones with spiral methods, how to divide projects by estimating workload and how to collaborated



Boston

Hong Kong

Los Angeles

**Mar 4, 2011**

Delay: 16.9ms

Packet Loss Rate: 0.0009%

Packet Reordering Rate: 0.003%

**Apr 5, 2011**

Delay: 89.3ms

Packet Loss Rate: 0.0022%

Packet Reordering Rate: 0.02%

(a) First level table containing locations.

(b) Measurements for current location to Hong Kong.

Fig. 5: Detail reports.

with team-mates closely to reach the target before deadlines. These experiences, without no doubt, have greatly improved and prepared our skills in a real, large software project.

However, during our developing process, we also made some mistakes, e.g., we underestimated the workload of some tasks. So, if starting again, we would like to re-divide the project and start earlier to develop the animation in 3D space. Besides, we would pay more attention to present the numerical results in more ways, on which users can make their own choice. We believe “more choice, more fun”.

## V. Contributions by Group Members

What did each group member contribute to the project is listed below.

TABLE I: Contributions by group members

| Work | Member |
|---|---|
| Building the overall structure of the app and preparing the canvas (i.e., a mapView and a mapRouteLayerView) for layering and animations | Yuefei Zhu |
| Designing functions turning network statistics into parameters for animations, calculating routes for “packets” move within | Yiwei Pu |
| Implementing animations on the mapView | Yiwei Pu |
| Drawing a 3D earth with Blender, exporting it into the Xcode project and implementing rotation and scaling with OpenGL ES | Yuefei Zhu |
| Table construction and file persistence | Yuefei Zhu |

## VI. Future Work

In the next step, to augment our application, we have two main tasks. One is to combine the user interface part with the module for measurement. To achieve this task, two obstacles have to be overcome. The first is to implement the interface function, whose task is to pass the control signal and measurement results to the interface. The second is to make the process of measuring more controllable, e.g., users can choose not only the number but also the exact location of the destination servers.

The other is to complete the feature of 3D animation. Specifically, we have to find out the methods to convert the real location to the coordinates on the sphere of the 3D earth. Furthermore, we have to transplant the path for animation from 2D space into the 3D space.

By installing our application, each iPhone holder becomes an active probing contributor. With the permission offered by users, this will anonymously upload the probed results as well as the user's geographical location provided by iPhone to dedicated servers. After analyzing the collected data from all online users in our backend, the server will feed the user the assembled data. Such data can be used as feed back for not only user terminals to adapt to the current network but also for network operator to provide better services.

Motivated by research purposes, this application is difficult to be commercialized, since users are tend to be unwilling to upload data for the fact that each byte of data uploaded will be charged by network operator, like Rogers. Even with this disappointing fact, it is still possible for this application to be popular, if attractive user interface is designed and more rewarding functionalities are added.

## References

[1] "Number of iPhone users." [Online]. Available: http://www.tech24hours.com/2009/05/number-of-iphone-users.html
[2] "Blender." [Online]. Available: http://www.blender.org/