

Project Report: *Network Pulse* – Network

Wei Wang (#998133951) and Yuan Feng (#996622347)
{2,487 words}

I. INTRODUCTION

Depicting a global picture of the Internet’s current status is quite costly: a set of computers have to be purchased and deployed around the world to conduct the measurement. Our application, *Network Pulse*, serves a costless approach for network measurement. We advocate to use iPhones to probe the global Internet. Our motivations reside in the facts that: *first*, almost all iPhone users are connected to the Internet; *second*, at present there are more than 21 million of iPhone users; *third*, iPhone users are distributed inherently across the world.

In particular, we are interested in measuring the round-trip time (RTT) and packet loss rate (PLR) of target websites browsed by people around the world. Our application sends out probe packets periodically to a target website, and measure the RTT and PLR based on the received feedbacks. The application then submits the measured results to the dedicated server, which will in turn send back a statistics summarizing the global measurements by analyzing all collected submissions. This project report focuses on the measurement part only, leaving the data presentation part to our partner team.

II. OVERALL DESIGN

Network Pulse adopts the canonical client-server architecture. The clients are running on iPhones, and periodically exchange information with the dedicated application server built on *Google App Engine*. Fig. 1 shows a detailed block diagram of our app. It describes a typical interaction cycle between a client and the server. A client starts to probe the Internet by sending out probe packets, HTTP requests for RTT and ICMP packets (ping) for PLR specifically. Once the response packets are received, the client starts to analyze those feedback information and extract useful statistics in its *Probing Manager* module (see Fig. 1). After accumulating enough statistics, a client submits a summarized probing report, together with its geographical location, to the server through the *Google App Engine Comm.* module. At the same time, clients can also retrieve the global measurement results processed from the server and direct them to the *Statistics Presentation Manager* module, which presents the result in a user-friendly way to users. The client repeats the processes above — probing, reporting, retrieving, and presenting — until the app is terminated by the user.

III. FUNCTIONALITIES AND SCREEN SHOTS

For this project, two measurements (RTT and PLR) regarding the interested website are supported in the client application. Also, the client can submit the measured results to the application server, from which it can retrieve the global measurement results collected from other iPhones around the world.

A. The Client

Since we are focusing on the network part, we adopt a simplified UI. As shown in Fig. 2a, the main menu offers two options, probing the network and retrieving the measurement statistics from the server. When the “Probing” button is pressed, the iPhone starts measuring the interested websites by sending out probe packets periodically (25s interval time for RTT and 5s for PLR).

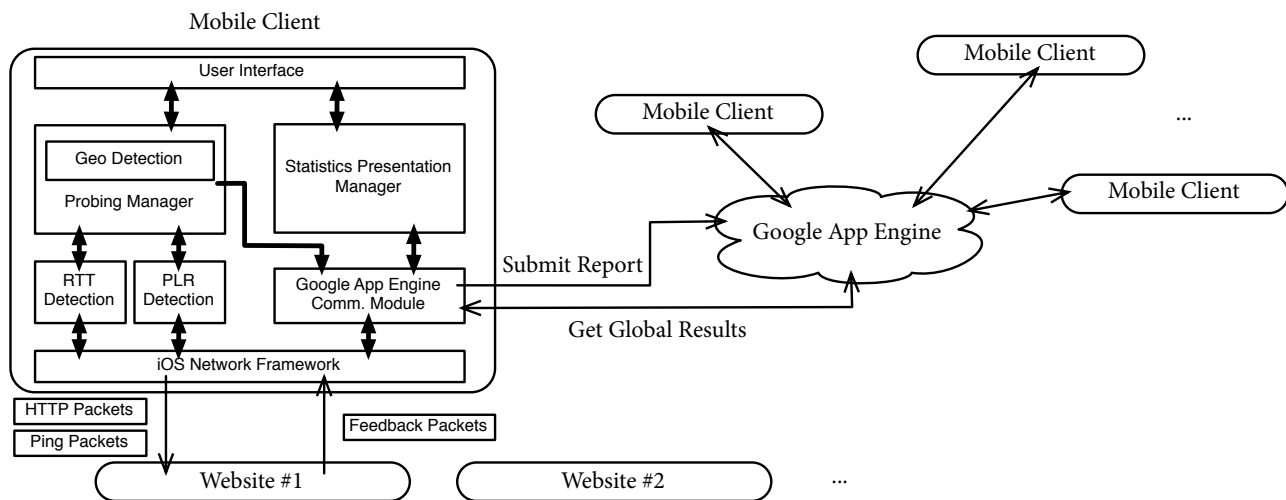
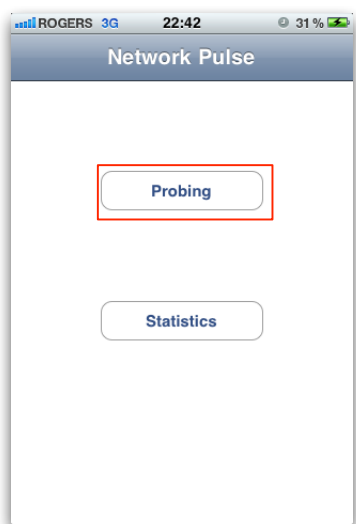
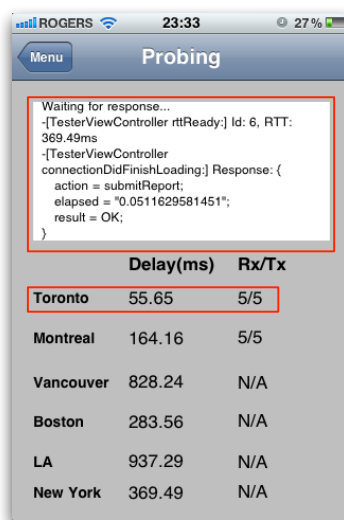


Fig. 1. The block diagram of *Network Pulse*.

The user will see the latest probing results, i.e., RTT and PLR, measured by her iPhone from a simple UI shown in Fig. 2b. For simplicity, the current implementation measures six predefined websites located in six different cities, but the user can also change to probe her interested website if she wishes. The measured RTT and PLR as well as the iPhone’s geographical location will also be submitted to the app server for further statistics. As illustrated in Fig. 2b, such uploading process is observed in the log window, where the measurement has been successfully reported to the server.



(a) The main menu.



Log Window: see log messages.

Latest Probing Results: RTT is 55.65ms without lost packet so far. The results are based on the latest probing from this iPhone to the dedicated websites in Toronto.

(b) The UI of network probing.

Fig. 2. The UI of network probing.

The user can also have a general picture of the global statistics collected from other iPhones around the world. By pressing the “Statistics” button, the user will see a list of websites. As shown in Fig. 3a, six websites are measured at the current implementation. The user can choose



(a) The menu of the measured websites. (b) The RTT of the selected website. (c) The PLR of the selected website.

Fig. 3. The UI of retrieving network statistics.

any of them and will be directed to a view of detailed statistics, where a tab in the bottom is used for switching between RTT and PLR, as shown in Fig. 3b and 3c, respectively. Meanwhile, the iPhone will connect to the server and retrieve the global statistics regarding the selected website. The measurements collected from the same city are clustered and averaged, with the result shown as an entry of the list. Since the only client is located in Toronto, we only see the statistics averaged over her reports, as depicted in Fig. 3b and Fig. 3c.

The working flow of RTT and PLR measurements are given in Fig. 4 and Fig. 5, respectively.

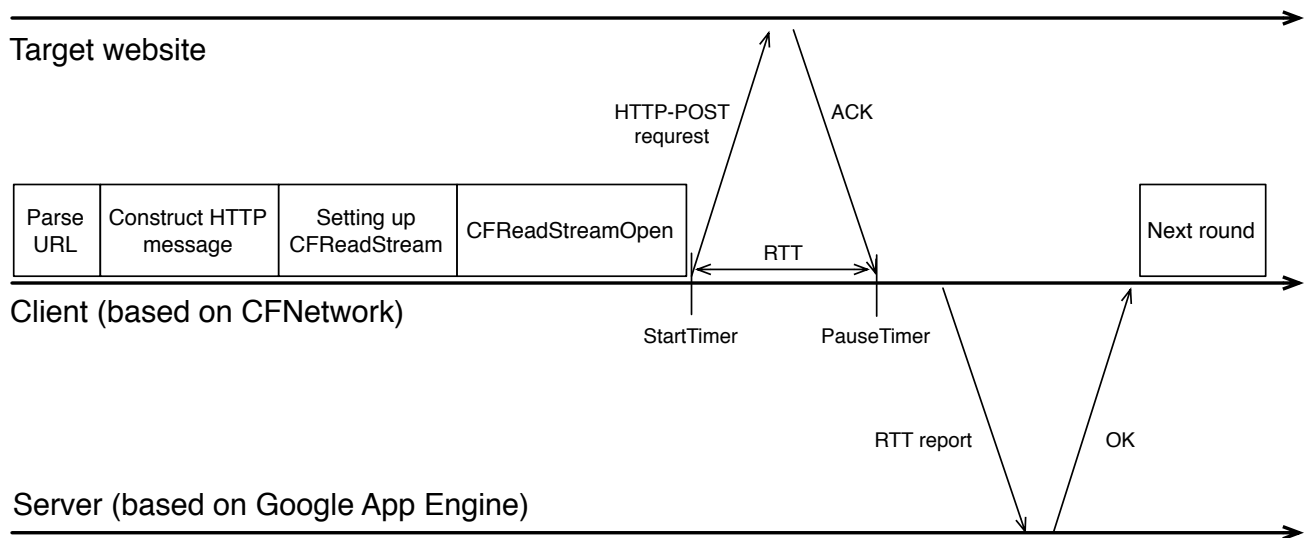


Fig. 4. The working flow of RTT measurement.

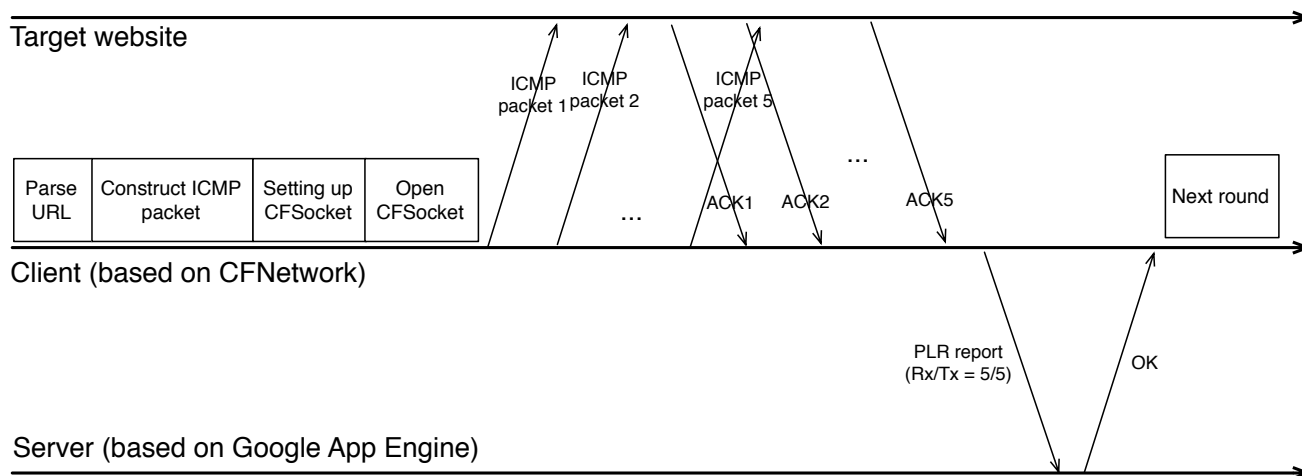


Fig. 5. The working flow of PLR measurement. Five ICMP packets will be sent for each PLR probing.

B. The Cloud Server

The main task of the application server is to collect probing results from clients all over the world, and to feed the global statistics back to them. Our application server is based on Google App Engine, for it provides a persistent yet highly scalable and distributed database interface, called *datastore*, for data storage and retrieval.

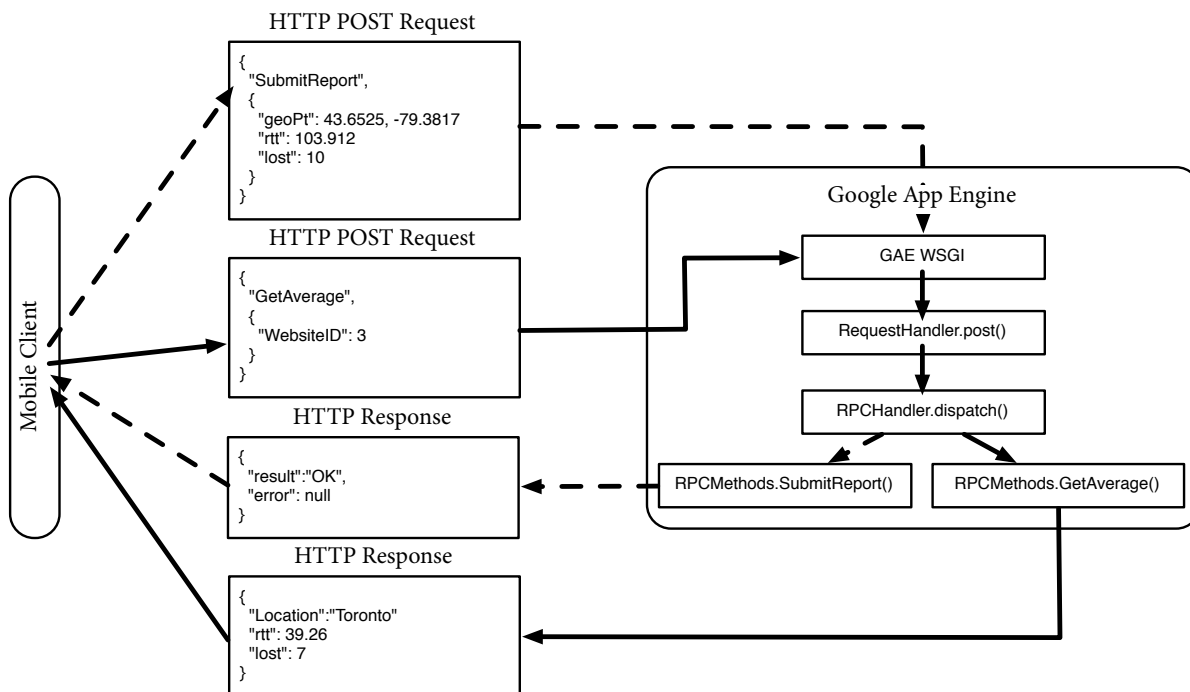


Fig. 6. The architecture of application servers.

Fig. 6 shows the working flow inside the server. We take the example when a user asks for the global statistics for a website, the data flow of which is shown by solid lines in the figure. The client sends out an HTTP POST request and the ID of the interested website, indicating that

the user's purpose is to retrieve the global statistics regarding to that website. This request is then directed by the *RPCHandler.dispatch* function to *GetAverage* function in this example. The Google App Engine then informs the user the averaged RTT and PLR via an HTTP Response message.

One of the important tasks for the server is to maintain all collected reports submitted by clients, and to update the datastore whenever a new report is submitted. Fig. 7 shows the screenshot of the Report table maintained by our servers. Each row of the table represents a report.

ID/Name	geoHash	geoPt	location	lost	rtt	timestamp	website
id=26338	dpz837kd6ch7x	43.6525,-79.3817	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	510.361	2011-04-05 14:09:11.434275	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATIM Website: name=2
id=26339	dpz837kd6ch7x	43.6525,-79.3817	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	785.855	2011-04-05 14:09:15.745996	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATYM Website: name=6
id=26340	dpz837kd6ch7x	43.6525,-79.3817	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	319.192	2011-04-05 14:09:40.526186	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATEM Website: name=1
id=26343	dpz837kd6ch7x	43.6525,-79.3817	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	1440.318	2011-04-05 14:10:28.169376	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATMM Website: name=3
id=26344	dpz837kd6ch7x	43.6525,-79.3817	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	1073.882	2011-04-05 14:10:33.369815	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATYM Website: name=6

Fig. 7. The screenshot of the Report table.

Another important task for the application server is to get the average measurement results of different websites. To be specific, the server will return the averaged RTT and PLR computed in the recent hour for each location. The server maintains a table called *Snapshot*, the screenshot of which is shown in Fig. 8.

ID/Name	count	location	lost_avg	rtt_avg	timestamp	website
id=26365	12	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	936.93	2011-04-05 15:10:00	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATQM Website: name=4
id=26366	11	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	1017.41109091	2011-04-05 15:10:00	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATEM Website: name=1
id=26374	6	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	16.6666666667	1037.709	2011-04-05 15:15:00	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATEM Website: name=1
id=26382	9	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	1436.77966667	2011-04-05 15:20:00	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATUM Website: name=5
id=26388	5	agxuZXR3b3JrcHVsc2VyEAsSCEvxY2F0aW9uGPbIA gw Location: id=42102	0.0	948.9746	2011-04-05 16:30:00	agxuZXR3b3JrcHVsc2VyDgsSB1dlYnNpdGUiATYM Website: name=6

Fig. 8. The screenshot of the Snapshot table.

For each website, the average of RTT and PLR are computed every five minutes. The average is computed over all reports submitted from the same location and is stored in the ScreenShot table. Therefore, a website might have a set of average statistics each of which corresponds to a specific location. For example, the first row stores the average measurement results for Website #4 at the location with ID 42102 at time 15 : 10 : 00 on April 5, 2011. By doing so, when the user requests for the global measurement results for a website, we just need to search the most recent 12 Snapshots for this website in the Snapshot table and calculate the average.

IV. LESSONS LEARNED

Spiral model is crucial for the success of a project. It advocates stepping progressively, instead of making a giant leap. It turns out that our plan is too ambitious at the very beginning. We have designed our own protocol that can measure the RTT, PLR and packet reordering rate at one probe. However, soon we realized that modifying low-level network protocol on iPhone is quite challenging, and is officially discouraged by Apple. We changed our protocol and built it based on *CFNetwork*, an official library encapsulating the low-level operations. We started from the basic HTTP communications and added new measurement features based on previous achievements. Though the current implementation is not as powerful as the original design, it is doable and has been successfully delivered.

Teamwork is also an important thing we have learned during the cooperation. We thought we have done well in allocating tasks and deciding responsibilities. The communication between group members is smooth and in time, especially when facing technical problems.

Should we have a chance to revisit everything, we would 1) try to step progressively, and 2) keep on working smoothly between each other.

V. CONTRIBUTIONS OF GROUP MEMBERS

The spirals of our project and contributions of each group member is summarized in Table I.

TABLE I
PROJECT SPIRALS AND GROUP MEMBERS' CONTRIBUTIONS.

Project Spirals	Accomplished By
Project Proposal	Yuan
Project Plan	Wei
Spiral 1: devices being able to measure RTTs by sending legitimate HTTP requests	Wei
Spiral 2: building up servers on the Google App Engine; servers being able to process dummy RTTs to get the average value	Yuan
Spiral 3.1: devices being able to send the summarized RTTs to servers and retrieve the processed RTT from servers	Yuan
Spiral 3.2: devices being able to measure PLRs by sending ping packets	Wei
Spiral 4.1: servers being able to process PLRs to get the average value	Wei
Spiral 4.2: building up a plain but user-friendly interface	Yuan
Project Report	Wei and Yuan

VI. FUTURE WORK

Packet reordering rate is currently unsupported in our implementation. The challenge resides in the fact that low-level network programming are officially discouraged by Apple. Figuring out a new implementation based on provided network library is the first thing to do in the next step.

To attract users to participate the measurement, the application has to present the statistics in a fancy way, which is implemented by our partner team. We plan to combine two team's work into a complete project.

Finally, we plan to release the application and use it to probing the Internet. By analyzing the collected data from different places over the world, we can get a more in-depth understanding of the global network environment.