A MacBook Based Earthquake Early Warning System

Shayan Mehrazarin, Bin Tang, Ken Leyba, Jianchao Han, and Mohsen Beheshti
Department of Computer Science
California State University Dominguez Hills (CSUDH), Carson, CA
smehrazarin1@toromail.csudh.edu, {btang,kleyba,jhan,mbeheshti}@csudh.edu

***Abstract*—We demonstrate an early stage of our prototypical development of an earthquake early warning system based on Apple's MacBook computers. This system, called MacSeisApp, utilizes the Sudden Motion Sensors in MacBooks to detect seismic activities. It is inspired by an existing application called SeisMac, which turns a MacBook computer into a seismograph. MacSeisApp expands on SeisMac by using Apple's Push Notifications (APN) via a dedicated server for earthquake notification. Written in both the C and Objective-C under XCode Integrated Developer Environment (IDE), MacSeisApp utilizes an open-source library to detect the vibrations via the Sudden Motion Sensor and to translate it into points to be plotted as a seismograph.**

Keywords—MacSeisApp; earthquake early warning; push notification; Sudden Motion Sensor, MacBook computer

I. INTRODUCTION

Most MacBook computers manufactured since 2006 are equipped with Sudden Motion Sensors (SMSs). The purpose of SMSs is to protect the MacBook's Hard Disk Drive (HDD) in the event that the MacBook is accidentally dropped or experiences heavy shaking and vibration. (MacBook models that feature a Solid State Drive (SSD) do not come with the SMSs.) Recently, SeisMac (http://www.suitable.com/tools/seismac.html) was developed to transform a MacBook into a seismograph by displaying real-time, three-axis acceleration graphs. Inspired by SeisMac and [1], we develop a new application called MacSeisApp, which not only displays a basic seismograph, but also aims to notify other users in nearby cities using Apple's Push Notifications (APN) via a dedicated server, therefore possibly serving as an earthquake early warning system. As shown in Fig. 1, when shakes detected, MacSeisApp communicates with the dedicated server based at CSUDH, which process the data and reports the location and time of the shake. It then communicates with Apple's Push Notification server (APSN), which sends a voice messaged-based alert to other MacBooks with the MacSeisApp.

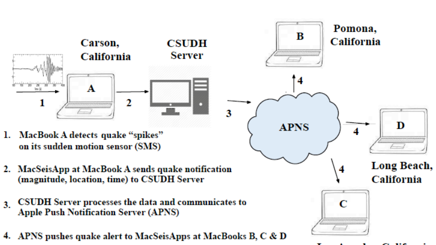

Figure 1: The General Structure of MacSeisApp.

II. AN INSIDE LOOK AT MACSEISAPP

The MacSeisApp is developed using a combination of Objective-C and C codes and utilizes the Xcode Integrated Development Environment (IDE). Objective-C contains a class called *AppDelegate*, which has two functions similar Fig. 2.

```
- (void)applicationDidFinishLaunching:(NSNotification *)aNotification {
    // Insert code here to initialize your application
    smsStartup(nil, nil);          // Used to initialize access to Sudden Motion Sensor (SMS)
    smsLoadCalibration();          // Used to load any stored calibration values
}

- (void)applicationWillTerminate:(NSNotification *)aNotification {
    // Insert code here to tear down your application
    smsShutdown();          // Shut down SMSLib once application is killed
}
```

Figure 2: The Application Delegate of Objective-C.

In MacSeisApp, the SMS is responsible for collecting shake data once every 0.1 second. The raw data collected is then translated by the open-source library *smslib* into values in three *x, y,* and *z* axes. MacSeisApp takes this data and stores it in three separate arrays with a size of one hundred. The data is then plotted to its the application window with each axis on its own separate line to depict a seismograph. The seismograph is then shifted everytime a new data is collected from the SMS. A total of one hundred plots of data is displayed at a time on each axis with the points interconnected with standard lines. The code for this scenario is shown in Fig. 3.

```
for(int i = 0; i < 99; i++)
{                                  // Shift all values back one index space
    x_axis[i] = x_axis[i + 1];
    y_axis[i] = y_axis[i + 1];
    z_axis[i] = z_axis[i + 1];
}

x_axis[99] = x;          // Insert new data at the end of array
y_axis[99] = y;
z_axis[99] = z;
```

Figure 3: Data Collection and Shifting in SMS.

III. DESIGNING THE SEISMOGRAPH INTERFACE

The seismograph interface for MacSeisApp was implemented using OpenGL. Our MacSeisApp seismograph (shown in Fig. 4(a)) looks a lot smoother than the SeisMac (shown in Fig. 4(b)). This is because the time interval used in SeisMac appears to be much longer than the interval of 0.1 second used in MacSeisApp. In order to ensure that the data and the seismograph interface are updated once every 0.1 seconds, an object of type NSTimer from Objective-C is used. The timer, in particular, is used for the following three functions:

- Shifting existing content in array to the previous index, and data at index 0 is overwritten,
- Adding new data to the end of the array at index 99,

- Redrawing the interface each time with the new contents of the array.

(a) SeisMac (b) MacSeisApp

Figure 4: Seismograph Interfaces.

IV. Calculating Magnitude From Data

Earthquakes are commonly measured based on the Richter scale. The data that is collected from the SMS for each of the three axes, however, is measured in *gravities (g)*, where the *x-axis* and *y-axis* range from -1.0g to 1.0g and the *z-axis* ranges from 0.0g to 2.0g. As gravity is a form of acceleration, *gravities* can easily be converted to *meters per square second* (m/s^2), where 1g is equal to 9.81 m/s^2.

The severity of an earthquake is also assessed by the Mercalli scale. The magnitude of an earthquake can be approximated on the Richter scale using a combination of a rating on the Mercalli scale and acceleration or gravity values (shown in Table 1, http://www.geography-site.co.uk/).

Table 1: Relationship Between Gravity/Acceleration and Magnitude

Richter Scale Value	Approximate Gravity (*g*)	Approximate Acceleration (m/s^2)	Approximate Mercalli Scale Value
5.0	± 0.03	± 0.29	V
5.4	± 0.05	± 0.49	VI
6.1	± 0.1	± 0.98	VII
6.6	± 0.25	± 2.45	VIII
7.3	± 0.5	± 4.91	X
8.0	± 0.75	± 7.36	XI
≥ 8.2	± 1	± 9.81	XII

The seismograph interface for MacSeisApp displays the data in the form of gravities (*g*). By using the conversions from Table 1, MacSeisApp can determine the approximate Richter scale value for 5.0 and above as long as the value of gravities (*g*) from both the *x-axis* and *y-axis* are greater than 0.03g. The Richter scale value is later used in the notification message in MacSeisApp in the event that an earthquake of magnitude 5.0 and above is detected.

Additionally, MacSeisApp calculates the average gravities value for the last one hundred values collected from each axis and takes the absolute value of the difference between the newest value and the average. The purpose of this calculation is to reduce the likelihood of "false alarms" and to account for cases where the MacBook is placed on an uneven surface.

V. Using Push Notifications in MacSeisApp

MacSeisApp utilizes Apple's Push Notification Service (APNS), which requires no interaction by the end-user to initiate notifications. It is used to get the end-user's immediate attention, as it is delivered instantly. The general idea and flow of the APNS in shown in Fig. 5.

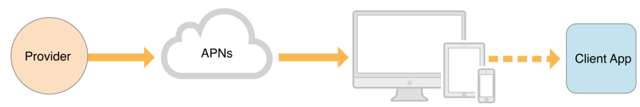

Figure 5: Structure and Flow of Apple's Push Notification Service (source: https://developer.apple.com/).

Suppose that there is a user running MacSeisApp located in the city of Torrance, California, detects a "spike". A push notification (shown in Fig. 6) is generated to notify users in nearby cities immediately. Those users will receive the push notifications seconds before the seismic wave reaches them. This is because data and network speeds are a lot faster than the speeds at which seismic waves travel through the ground.

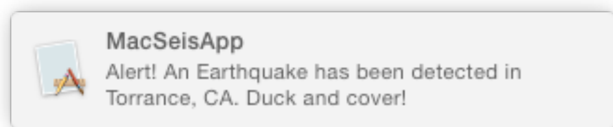

Figure 6: A Sample Push Notification From MacSeisApp

<u>"Triggering" algorithms.</u> We developed a "Triggering" algorithm on CSUDH server that has the ability to differentiate between artificial or man-made shaking and legitimate seismic activity. It sends a message to Apple Push Notifications Server only if there are at least three end-users in one location that simultaneously detect some shakes. The rationale is that if multiple users detect spikes simultaneously, it is very likely that some real seismic activity has occurred.

VI. Future Work

One feature to be added would be a counterpart application for iOS where users can also receive push notifications on their iPhone or iPad devices as well. This could become a very effective way to instantly notify a great amount of users over multiple platforms. We will also improve the triggering algorithm by considering real earth quake models.

Acknowledgment

The research was supported by NSF Grant CNS - 1419952.

References

[1] E. Cochran, J. Lawrence, C. Christensen, and R. Jakka. The quake catcher network: Citizen science expanding seismic horizons. Seismological Research Letters, 80(1):26–30, 2009.