TO 0014 Smart Phone Based In-Vehicle Driver Warnings for Pedestrian Mid-Block Crossings

Performance Test Plan and Report

Version: 2.0

11/03/2017

Leidos

# Revision History

|  |  |  |
| --- | --- | --- |
| Revision | Date | Comment |
| Version 2.0 | 11/03/17 | Added data and conclusions from second round of performance testing |
| Version 1.0 DRAFT | 07/21/17 | Initial draft |

1. Contents

[1. Revision History 2](#_Toc497489610)

[3. Background 4](#_Toc497489611)

[4. Testing Goals 4](#_Toc497489612)

[5. Testing Personnel 4](#_Toc497489613)

[6. Testing Equipment 4](#_Toc497489614)

[7. Testing Location 4](#_Toc497489615)

[8. Test Methodology 5](#_Toc497489616)

[9. Data Analysis 5](#_Toc497489617)

[10. First Performance Test Results 6](#_Toc497489618)

[11. Conclusions 9](#_Toc497489619)

[12. Potential Improvements 10](#_Toc497489620)

[13. Second Performance Test Results 10](#_Toc497489621)

[14. Second Performance Test Conclusions 13](#_Toc497489622)

# Background

For Saxton Transportation Operation Laboratory Task Order 14 Leidos developed two software applications for the purpose of driver notification with regards to pedestrians crossing at mid-block crosswalks. These applications (one installed in a cloud server, one installed on two or more Android devices) communicate over the network, access device GPS data, and record data for later analysis during the process of normal operation. These applications will be used as part of an academic study concerning the efficacy of such a system at alerting drivers to the presence and intent of pedestrians at a mid-block crossing.

# Testing Goals

The goal of executing this test plan and analyzing the resulting data is to quantify the unknown factors of GPS inaccuracy in the Android devices, the expected values of network latency and how that value may vary, and to ensure that the server application is capable of effectively recording all the data that it receives. These metrics will be used to ensure that the application software and computer hardware are capable of meeting the needs of the study as well as to provide additional data that may be useful during the study’s data analysis.

# Testing Personnel

Pedestrian – Participant who will use the Android application’s pedestrian role to request a crossing and then cross at the midblock crossing.

Driver – Participant who will use the Android applications driver role to be alerted of the pedestrian’s intent to cross and stop to allow the crossing.

# Testing Equipment

The performance testing will utilize the same equipment as is planned to be used in the experimental evaluation. As such the Android devices will be one Samsung Galaxy S7 and one Samsung Galaxy Tab E. The server side infrastructure will be one t2.micro instance hosted in Amazon’s EC2 ElasticBeanstalk cloud. One vehicle will also be used by the driver participant in the testing.

# Testing Location

The performance testing for the TO 14 applications will take place at the Turner-Fairbanks Highway Research Center’s (TFHRC) Connected Vehicle Highway Testbed (CVHT). This is the same location that the experimental testing for the study will be conducted, so the cellular tower the Android devices will communicate with will be the same and the latency should be comparable between performance testing and the experimental testing. The route used for the performance testing will be a subset of the experimental route, from the TFHRC main parking lot through both intersections by means of Innovation Drive. Data will be collected for the driver device from the parking lot to the far side of the westmost intersection and from one side of Innovation Drive to the other for the pedestrian device.

# Test Methodology

This performance test aims to closely mimic the experimental conditions to provide data about how the application performs during use. As such the performance testing relies on a singular test method executed multiple times on both devices to generate data for later analysis. The methodology for a single test is as follows:

1. Install the STOL TO 14 app on both the Samsung Galaxy S7 and on the Samsung Galaxy Tab E
2. Install the Samsung Galaxy Tab E in a vehicle and bring both devices and both test participants to the Turner-Fairbanks Highway Research Center midblock crossing for which the application is configured.
3. Position the pedestrian (carrying the Samsung Galaxy S7) several meters away from the midblock crossing.
4. Position the vehicle (with the Samsung Galaxy Tab E) in the TFHRC main parking lot.
5. Both testers open the app on their respective devices and select their respective roles.
6. Have the pedestrian start the serverside data collection by using the “Start logging” menu option.
7. Have the pedestrian enter the pedestrian geofence.
8. The driver should begin to approach the midblock crossing from the parking lot.
9. Before the vehicle completes the right turn at the newer TFHRC intersection, the pedestrian presses the “Request Crossing” button.
10. The vehicle drives normally along the route between the two TFHRC intersections until the alert is received.
11. When the alert is received the vehicle will slow to a stop to allow the pedestrian to pass.
12. After the pedestrian has crossed the driver will proceed through the midblock crossing and following intersection.
13. Once the driver has cleared the TFHRC older intersection, the pedestrian will stop serverside data logging via the “Stop logging” menu option.

This test case will be executed three times with the normal device configuration and three times with the devices swapped roles (i.e. Samsung Galaxy Tab E as pedestrian and Samsung Galaxy S7 as driver). Once these tests have been completed, the log files will be downloaded from the server and analyzed.

# Data Analysis

The data analysis for the performance testing will be conducted by Ruby scripts parsing the Comma-Separated Values (CSV) log files generated by the server application. The server will produce two files for each test run – one for events and one for data reports – but only one (the data file) will be used in this analysis. The data report file includes timestamped latency and location accuracy data reported by each device at all times logging is enabled during a test run. This data is configured to be captured at a 5hz rate and is stored as it arrives at the server. The Ruby script will parse the CSV structure and compute the averages and standard deviations for the following performance parameters:

**Latency:** Defined as the time it takes for a device to encode and send a Hyper-Text Transfer Protocol (HTTP) request to the server and then receive and decode the response. This round trip time will be measured in milliseconds on the measuring device’s clock.

**Location Accuracy:** Location accuracy data is taken directly from the Android device’s location service and is defined in the [Android documentation](https://developer.android.com/reference/android/location/Location.html) as follows:

Get the estimated horizontal accuracy of this location, radial, in meters.

We define horizontal accuracy as the radius of 68% confidence. In other words, if you draw a circle centered at this location's latitude and longitude, and with a radius equal to the accuracy, then there is a 68% probability that the true location is inside the circle.

In statistical terms, it is assumed that location errors are random with a normal distribution, so the 68% confidence circle represents one standard deviation. Note that in practice, location errors do not always follow such a simple distribution.

This accuracy estimation is only concerned with horizontal accuracy, and does not indicate the accuracy of bearing, velocity or altitude if those are included in this Location.

**Log Frequency:** Defined as the delay, in milliseconds, between consecutive log entries in the data file for a given device. This represents the minimum frequency (i.e. not counting event entries or other more spontaneous communications) with which the device is communicating with the server.

Each of these metrics will be computed for each device per each run.

# First Performance Test Results

|  |  |
| --- | --- |
| Test Run # | 1 |
| File Name | 21-07-2017-17-30-675DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 83.208 |
| StdDev Pedestrian Latency (ms) | 10.243 |
| Average Driver Latency (ms) | 130.947 |
| StdDev Driver Latency (ms) | 11.623 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.551 |
| StdDev Pedestrian Location Accuracy (m) | 0.000 |
| Average Driver Location Accuracy (m) | 9.285 |
| StdDev Driver Location Accuracy (m) | 0.481 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 27.964 |
| Average Driver Log Frequency (ms) | 201 |
| StdDev Driver Log Frequency (ms) | 60.992 |

|  |  |
| --- | --- |
| Test Run # | 2 |
| File Name | 21-07-2017-17-33-578DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 78.379 |
| StdDev Pedestrian Latency (ms) | 10.564 |
| Average Driver Latency (ms) | 121.036 |
| StdDev Driver Latency (ms) | 9.295 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.682 |
| StdDev Pedestrian Location Accuracy (m) | 0.427 |
| Average Driver Location Accuracy (m) | 10.432 |
| StdDev Driver Location Accuracy (m) | 1.001 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 19.053 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 37.497 |

|  |  |
| --- | --- |
| Test Run # | 3 |
| File Name | 21-07-2017-17-37-657DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 77.275 |
| StdDev Pedestrian Latency (ms) | 12.110 |
| Average Driver Latency (ms) | 125.454 |
| StdDev Driver Latency (ms) | 15.342 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.551 |
| StdDev Pedestrian Location Accuracy (m) | 3.553e-15 |
| Average Driver Location Accuracy (m) | 9.2 |
| StdDev Driver Location Accuracy (m) | 0.400 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 22.627 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 58.078 |

|  |  |
| --- | --- |
| Test Run # | 4 |
| File Name | 21-07-2017-17-40-804DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 107.431 |
| StdDev Pedestrian Latency (ms) | 7.643 |
| Average Driver Latency (ms) | 91.692 |
| StdDev Driver Latency (ms) | 11.615 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.744 |
| StdDev Pedestrian Location Accuracy (m) | 2.617 |
| Average Driver Location Accuracy (m) | 6.066 |
| StdDev Driver Location Accuracy (m) | 2.351 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 22.627 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 58.078 |

|  |  |
| --- | --- |
| Test Run # | 5 |
| File Name | 21-07-2017-17-43-195DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 105.425 |
| StdDev Pedestrian Latency (ms) | 12.9523 |
| Average Driver Latency (ms) | 91.037 |
| StdDev Driver Latency (ms) | 7.564 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 7.083 |
| StdDev Pedestrian Location Accuracy (m) | 3.399 |
| Average Driver Location Accuracy (m) | 4.551 |
| StdDev Driver Location Accuracy (m) | 0.0 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 45.978 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 16.217 |

|  |  |
| --- | --- |
| Test Run # | 6 |
| File Name | 21-07-2017-17-47DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 108.628 |
| StdDev Pedestrian Latency (ms) | 12.240 |
| Average Driver Latency (ms) | 74.962 |
| StdDev Driver Latency (ms) | 11.819 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 3.061 |
| StdDev Pedestrian Location Accuracy (m) | 0.240 |
| Average Driver Location Accuracy (m) | 4.551 |
| StdDev Driver Location Accuracy (m) | 0.0 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 201 |
| StdDev Pedestrian Log Frequency (ms) | 46.347 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 18.276 |

# First Performance Test Conclusions

From the above data it can be seen that on average the devices are capable of meeting their current performance configuration. The latency for each device is less than the total duration of a device timestep such that the devices do not spend much time waiting for network activity when doing their computations. The average log frequencies for each device are very close to the expected value for a 5hz update frequency but the relatively large standard deviations suggest that - while the variance averages out over a the whole data set – any given timestep is not guaranteed to land within the specified frequency, suggesting jitter in the frequency at which the devices communicate on the device. The location accuracy figures appear to be quite reliable given their very low (or even 0) standard deviation values. While the confidence circle for the devices appears to be relatively large it still seems largely suitable for an application such as this where the primary purpose is awareness of the pedestrian rather than any sort of automated control behavior to avoid them.

As a comparison between the devices, it would appear that the Samsung Galaxy S7 is an overall more performant device than the Samsung Galaxy Tab E. Without speculating as to cause, it appears to experience overall lower network latency and lower variation in location accuracy values. Many of the other differences appear more related to the selected role than to the device perhaps pointing to the software itself (server-side or client-side) causing some of the differences.

# First Performance Test Potential Improvements

From the data set there appear to be hints towards a few possible paths forward. Further investigation of the differences in network and computational behavior of the two modes of the Android application and server may yield improvements in terms of reduced variation of logging frequency.

Few gains are likely to be had in any way related to location accuracy. Location accuracy appears to be a characteristic of the device and of the GPS solution available to the device at run-time. Possible routes for improvement in this metric would likely involve changing at least one if not both of the chosen Android devices.

Log frequency may be able to be improved in reliability (in terms of reducing standard deviation) through a more thorough analysis of the software components. Since log frequency appears to differ between the two roles rather than between the two devices an analysis of the differences in network usage and application profiles between the two roles might lead to bringing the driver mode performance more in line with the pedestrian mode and/or improvements across the board.

# Second Performance Test Results

A second round of performance testing was conducted after modifications to the Android application and the Server application to verify the continued performance of the software system. The modifications added support for multiple approaches to the same mid-block crossing, heading-based alert delivery, improved interface feedback to the pedestrian, and fixes to various crashes and bugs. This test was conducted using the same methodology, hardware, and environment as the original performance testing in order to ensure that the data is properly comparable. The following data is derived using the same analysis script as the first performance test results:

|  |  |
| --- | --- |
| Test Run # | 1 |
| File Name | 03-11-2017-17-15-255DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 81.096 |
| StdDev Pedestrian Latency (ms) | 4.080 |
| Average Driver Latency (ms) | 74.973 |
| StdDev Driver Latency (ms) | 9.934 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.861 |
| StdDev Pedestrian Location Accuracy (m) | 0.612 |
| Average Driver Location Accuracy (m) | 16.155 |
| StdDev Driver Location Accuracy (m) | 3.183 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 13.000 |
| Average Driver Log Frequency (ms) | 199 |
| StdDev Driver Log Frequency (ms) | 17.146 |

|  |  |
| --- | --- |
| Test Run # | 2 |
| File Name | 03-11-2017-17-18-24DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 75.903 |
| StdDev Pedestrian Latency (ms) | 10.104 |
| Average Driver Latency (ms) | 48.987 |
| StdDev Driver Latency (ms) | 11.438 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.790 |
| StdDev Pedestrian Location Accuracy (m) | 0.552 |
| Average Driver Location Accuracy (m) | 18.320 |
| StdDev Driver Location Accuracy (m) | 5.867 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 13.000 |
| Average Driver Log Frequency (ms) | 217 |
| StdDev Driver Log Frequency (ms) | 347.127 |

|  |  |
| --- | --- |
| Test Run # | 3 |
| File Name | 03-11-2017-17-20-876DATA.csv |
| Pedestrian Device | Samsung Galaxy S7 |
| Driver Device | Samsung Galaxy Galaxy Tab E |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 80.471 |
| StdDev Pedestrian Latency (ms) | 14.982 |
| Average Driver Latency (ms) | 90.074 |
| StdDev Driver Latency (ms) | 9.363 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 4.551 |
| StdDev Pedestrian Location Accuracy (m) | 0.0 |
| Average Driver Location Accuracy (m) | 22.241 |
| StdDev Driver Location Accuracy (m) | 7.681 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 13.000 |
| Average Driver Log Frequency (ms) | 218 |
| StdDev Driver Log Frequency (ms) | 338.077 |

|  |  |
| --- | --- |
| Test Run # | 4 |
| File Name | 03-11-2017-17-20-876DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 67.801 |
| StdDev Pedestrian Latency (ms) | 6.890 |
| Average Driver Latency (ms) | 76.517 |
| StdDev Driver Latency (ms) | 8.074 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 13.345 |
| StdDev Pedestrian Location Accuracy (m) | 1.778 |
| Average Driver Location Accuracy (m) | 5.956 |
| StdDev Driver Location Accuracy (m) | 2.929 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 16.401 |
| Average Driver Log Frequency (ms) | 200 |
| StdDev Driver Log Frequency (ms) | 11.790 |

|  |  |
| --- | --- |
| Test Run # | 5 |
| File Name | 03-11-2017-17-25-441DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 78.727 |
| StdDev Pedestrian Latency (ms) | 9.976 |
| Average Driver Latency (ms) | 69.360 |
| StdDev Driver Latency (ms) | 6.529 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 14.015 |
| StdDev Pedestrian Location Accuracy (m) | 0.647 |
| Average Driver Location Accuracy (m) | 4.551 |
| StdDev Driver Location Accuracy (m) | 0.0 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 32.047 |
| Average Driver Log Frequency (ms) | 200 |
| StdDev Driver Log Frequency (ms) | 12.490 |

|  |  |
| --- | --- |
| Test Run # | 6 |
| File Name | 03-11-2017-17-25-441DATA.csv |
| Pedestrian Device | Samsung Galaxy Tab E |
| Driver Device | Samsung Galaxy Galaxy S7 |
| **Latency (ms)** |  |
| Average Pedestrian Latency (ms) | 71.737 |
| StdDev Pedestrian Latency (ms) | 11.795 |
| Average Driver Latency (ms) | 78.546 |
| StdDev Driver Latency (ms) | 4.818 |
| **Location Accuracy (m)** |  |
| Average Pedestrian Location Accuracy (m) | 13.653 |
| StdDev Pedestrian Location Accuracy (m) | 0.0 |
| Average Driver Location Accuracy (m) | 4.581 |
| StdDev Driver Location Accuracy (m) | 0.211 |
| **Log Frequency (ms)** |  |
| Average Pedestrian Log Frequency (ms) | 199 |
| StdDev Pedestrian Log Frequency (ms) | 14.457 |
| Average Driver Log Frequency (ms) | 200 |
| StdDev Driver Log Frequency (ms) | 14.457 |

# Second Performance Test Conclusions

Based on the data and analysis from above it would appear that many of the conclusions drawn from the first performance test data set still hold after the software modifications were performed. Improvement in average latency overall seems to have taken place, though unpredictable cellular network conditions may play a role. This may also be due to changes in the implementation of the networking threads in the Android application. In either case, the latency still remains below the timestep size of the devices indicating that network latency is unlikely to cause significant issues for application functionality. This is borne out by the fact that the average log frequency for both devices in all tests is very close to 200 ms, indicating that data is being received by the server at the expected rate.

Test case 3 shows a concerning standard deviation in log frequency of over 300 ms. It would appear as though the vehicle device lost network connection (there was no duplication of “MOTORIST\_REGISTERED” events in the corresponding EVENTS log) between the timestamps of 1509729643040 and 1509729648905, causing this large deviation. Omitting this loss of connection from the sample causes the standard deviation in log delay to drop to 20.95 ms, comparable to other tests. Similar activity appears to have taken place in test 2, where similarly discarding the outlier results in a value of 30.315 ms.

The location accuracy of the tablet device appears to have degraded compared to the previous performance test. The average accuracy values for the Samsung Galaxy Tab E ranged from 13.345 m to 22.241 m compared to the same values for the previous test of 3.061 to 10.432. Since the location accuracy of the device is not controllable via application software it may be that environmental conditions such as cloud cover affected the GPS accuracy of the Samsung Galaxy Tab E in a negative fashion. Such marked differences in accuracy were not observed with the Samsung Galaxy S7, suggesting a potentially more sophisticated or robust GPS solution may be used in the Samsung Galaxy S7 compared to the Samsung Galaxy Tab E.

# Second Performance Test Potential Improvements

Potential improvements based on the data gathered in the second performance test are similar to those from the first performance test. Further evaluation of the GPS accuracy and quality of the Samsung Galaxy Tab E may be merited based on the observed fluctuation in average GPS accuracy between the two tests. If the device is affected by environmental conditions in such a way as to see large changes in GPS accuracy (relative to the minimal changes observed with the Samsung Galaxy S7) it may be advisable to replace the device with one that utilizes a higher quality GPS solution.

While log frequency remains similar in average and deviation to the first performance test data set, improvements may still be had in reducing the work each device must perform. In addition to the aforementioned analysis, profiling, and optimization of the Android applications themselves it seems likely that offloading some of the computational burden (geofence computation, alert status tracking) to the Server software may improve these metrics further. This would require the server to be sufficiently powerful to process the data for the clients, however, and may prove un-scalable in the long term, were this system to be deployed in a large environment.