Title
Providing In-Vehicle Soft Safety Alerts Using Mobile Millennium Data and Vehicle Event Information

Authors
Nowakowski, Christopher
Gupta, Somak Datta
Myers, Scott
et al.

Publication Date
2012-09-12

Providing In-Vehicle Soft Safety Alerts
Using Mobile Millennium Data and Vehicle Event Information

Final Report For Renault

September 21, 2012

Authors

Christopher Nowakowski
Somak Datta Gupta
Scott Myers
Steve Shladover
Joe Butler
Alex Bayen

ABSTRACT

The U.C. Berkeley, California PATH-Renault collaboration, detailed in this report, built upon the Mobile Millennium and Networked Traveler projects that were conducted from 2007 to 2011 as part of the US DOT's SafeTrip-21 Initiative. The Mobile Millennium project provided a platform for aggregating traffic information across various sources, including infrastructure sensors, commercial data feeds, probe vehicles, and probe cell phones. The Networked Traveler project provided the California PATH instrumented research vehicle platform used to both deliver vehicle probe data back to the infrastructure and to generate Advanced Driver Assistance Systems (ADAS) alerts to the drivers of those vehicles. The main theme of this collaboration project was to demonstrate the potential to create Enhanced Probe Vehicles (EPVs) by merging vehicle CAN-bus data with the typical GPS data that is provided by normal probe vehicles. EPVs could then provide more accurate speed information to the traffic servers along with additional information that only the vehicle knows, such as current gear, hazard light warning activation, or even airbag deployment events.

During SafeTrip-21, the two California PATH research platforms were built independently, so the first goal of this project was integrate the two systems, allowing the instrumented research vehicles to communicate with the Mobile Millennium traffic aggregation servers. The second goal was to enhance the data gathered and stored by the Mobile Millennium traffic aggregation server to include the enhanced information provided by the EPVs, specifically, to include hazard warning light activation events. The third and final goal of this project was to integrate the information provided by the EPVs to the traffic server into soft-safety alerts that could be provided to the drivers of the EPVs and demonstrate the prototype system. This report documents the systems that were built for the demonstration, and provides a short analysis of the system performance from data gathered through both a simulation and an on-the-road vehicle test.

Key Words: ITS, Intelligent Transportation Systems, ATIS, Advanced Traveler Information Systems, Traffic Information, Human Factors, Driving Behavior, Situational Awareness, Soft Safety Alerts, End-of-Queue Warnings

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Background

The U.C. Berkeley, California PATH-Renault collaboration, detailed in this report, builds upon two projects that were conducted as part of the US DOT's SafeTrip-21 initiative. The SafeTrip-21 initiative started in 2007 and concluded in 2011. It was launched by the US DOT's Research and Innovative Technology Administration (RITA), in partnership with the California Department of Transportation (Caltrans) and other state DOTs, to demonstrate how current ITS (Intelligent Transportation Systems) solutions could improve transportation safety and reduce congestion. Although there were a number of different and diverse projects conducted under the SafeTrip-21 initiative, the two projects relevant to the current collaboration included the Mobile Millennium Project and the Networked Traveler Foresighted Driving Field Experiment, both conducted at California PATH.

In the Mobile Millennium Project (Bayen, et al., 2010, 2011), California PATH demonstrated the potential of GPS in cell phones to alter the way traffic data is collected. The existing cell phone infrastructure was leveraged to collect vehicle speed and travel time data and merge it with data gathered from the infrastructure through the California Performance Measurement System (PeMS) and through traffic speed and travel time estimates provide by private companies such as Navteq (http://www.navteq.com) and SpeedInfo (http://www.speedinfo.com).

In the Networked Traveler Foresighted Driving Field Experiment (Nowakowski, et al., 2011, 2012), California PATH demonstrated how Advanced Driver Assistance Systems (ADAS) could provide real-time soft-safety alerts to drivers, and this project demonstrated how these alerts could benefit drivers approaching end-of-queue traffic scenarios. In this project, commercial data sources such as Navteq and SpeedInfo were leveraged, along with the existing cell phone infrastructure, to provide the drivers with a "Slow Traffic Ahead" alert about 60 seconds before reaching traffic that had significantly slowed.

It was hoped that the soft-safety alerts might reduce end-of-queue, rear-end crashes through one of several means. By increasing the driver's expectation of an impending speed change with an alert, we might decrease the probability of an inopportune driver distraction, influence smoother speed changes, and minimize the speed differentials among the vehicles in the traffic flow. Each of these reactions should help to reduce the probability of an end-of-queue crash.

To test this hypothesis, four California PATH research vehicles were instrumented to collect driving behavior data and provide soft-safety alerts to the drivers when travelling around the San Francisco Bay Area. A total of 24 drivers participated in the experiment for two weeks over the course of 5 months. During the first week, the system simply collected data, but during the second week, the system provided drivers with an audible "Slow Traffic Ahead" alert, also specifying the detected speed of the traffic ahead. Among the test results, there were clear reductions in peak deceleration rates when the soft-safety alerts were provided, suggesting that these types of alerts might provide some safety benefits.

However, the raw traffic data provided by the commercial vendors was not without issue, and over 40 percent of the alerts received during the field experiment ended up being false alarms related to incomplete or inaccurate data. Since the Mobile Millennium project provided a robust data aggregation and traffic speed prediction platform, it was a natural extension of the two projects to attempt to merge the functionality provided by each of the two systems.

1.2 Project Overview

1.2.1 Project Goals

The ultimate goal of the PATH-Renault collaboration is to build and demonstrate a collaborative system utilizing V2I communication combining the enhanced information that can be generated from probe vehicles, with estimates of traffic density and the road network conditions, to provide contextual safety-related information to drivers. To this end, the Mobile Millennium project provides a platform for aggregating traffic information across various sources, including infrastructure sensors, commercial data feeds, probe vehicles, and probe cell phones. The California PATH instrumented research vehicles, built for the Networked Traveler Foresighted Driving Experiment, provide a platform to both deliver enhanced vehicle probe data back to the infrastructure and to generate ADAS alerts to the drivers of those vehicles.

The first goal of the project was to integrate the Networked Traveler Foresighted Driving Experiment vehicle platform with the data that could be provided by the highway traffic models built during the Mobile Millennium project. While this integration was relatively straightforward, there were a number of implementation challenges to be discussed later. Once this goal was completed, the two California PATH research vehicles would be able to receive traffic speed data that had been gathered and processed through the Mobile Millennium architecture, allowing the vehicle to provide the drivers with soft-safety alerts regarding "Slow Traffic Ahead."

The second goal of this project was to introduce the concept of an enhanced probe vehicle. Currently, probe vehicles or probe cell phones provide basic GPS information (latitude, longitude, and a GPS speed estimate) to the data aggregation servers which feed into the highway traffic models that were built during the Mobile Millennium project. In contrast, an enhanced probe vehicle would gather and transmit data both from the GPS and from the vehicle's own CAN bus. Using the additional information gathered from the vehicle's CAN bus, the probe data can be enhanced, first, to provide a more accurate vehicle speed and, second, to include data known only to the vehicle, such as current gear, hazard light status, or even an airbag deployment.

Furthermore, adverse events involving the enhanced probe vehicle could be defined, recorded, and transmitted to the traffic data aggregation servers, and then this information could be fed back to other approaching vehicles in the form of additional soft-safety alerts. As an example, if an enhanced probe vehicle was involved in a collision, it might be able to transmit this event to the traffic aggregation servers, and once this propagated through the system, approaching vehicles would not only be warned about slow traffic ahead, but they might also be warned about an accident or disabled vehicle ahead. For the purposes of this collaborative demonstration, the initiation of the hazard warning lights would serve as a surrogate for the types of events that could be generated by an enhanced probe vehicle. The California PATH research vehicle platform was already set up to read and send CAN bus information, but there was a fair amount of work to be done on the Mobile Millennium side to aggregate, store, process, and serve up the enhanced probe data to approaching vehicles.

The third and final goal of the project was to integrate the information provided about traffic events from the enhanced probe vehicles into the soft-safety alerts provided to the drivers. During the Networked Traveler project, the only alerts provided to the drivers were based on traffic speeds, warning the drivers about slow traffic ahead. With the introduction of enhanced probe vehicles, additional soft-safety alerts could be provided to the drivers, and alterations to the alert system in the California PATH research vehicle that was left over from the Networked Traveler project would be required.

1.2.2 Demonstration Use-Case Scenarios

Three use-case scenarios were proposed for demonstration in this project. The first use-case scenario involved a major freeway traffic slowdown. The demonstration scenario would use one of the instrumented Nissan Altimas acting as enhanced probe vehicles (EPV) and receiving traffic information from the infrastructure. As the vehicle approaches a real traffic jam, it receives the speeds of the traffic ahead from the Mobile Millennium server, and determines when to give the driver an alert regarding slowed traffic ahead. Basically, if the vehicle is travelling more than 15 mph above the mean speed of traffic that is about a mile ahead, then an alert would be given. As an example, the EPV is travelling at 65 mph and the traffic ahead is 35 mph, then the EPV would issue an auditory alert saying, “Slow Traffic Ahead, 35 Miles Per Hour,” when the EPV was about 60 seconds from the end of the slowed traffic queue.

In the second use-case scenario, a disabled EPV would be simulated using one of the two Nissan Altimas, and as the second vehicle approached, it would get a warning about the presence of the disabled vehicle. In this scenario, both Nissan Altimas would be acting as EPVs. The first vehicle enters the highway, travels for a few miles, and then pulls off to the side of the road while engaging its hazard warning lights. The second Nissan Altima would follow the same route, getting information about both traffic speeds and EPV events from the Mobile Millennium server. As the second vehicle approaches the simulated disabled vehicle, an auditory alert would be given to the driver, such as, “Caution, Disabled Vehicle Ahead.”

The third use-case scenario is simply a combination of the first two scenarios, and it demonstrates the possibility of providing context aware soft-safety alerts. In this scenario, both Nissan Altimas will be acting as EPVs. The first vehicle enters the highway, travels for a few miles, locates an area with a queue of slowed or stopped traffic, and then pulls off to the side of the road while engaging its hazard warning lights. The second Nissan Altima would follow the same route, getting information about both traffic speeds and EPV events from the Mobile Millennium server. As the second vehicle approaches both the slowed traffic and the simulated disabled vehicle, an auditory alert would be given to the driver, such as, “Caution, Disabled Vehicle Ahead, 35 Miles Per Hour.”

1.3 System Overview

1.3.1 Architecture Overview

The proposed system overview architecture is shown in Figure 1.1, consisting of two major components. The first major component is the traffic data server, which was built upon the Mobile Millennium architecture. The second major component is the enhanced probe vehicles, which were built upon the existing architecture provided by the two California PATH research vehicles that were developed during the Networked Traveler project.

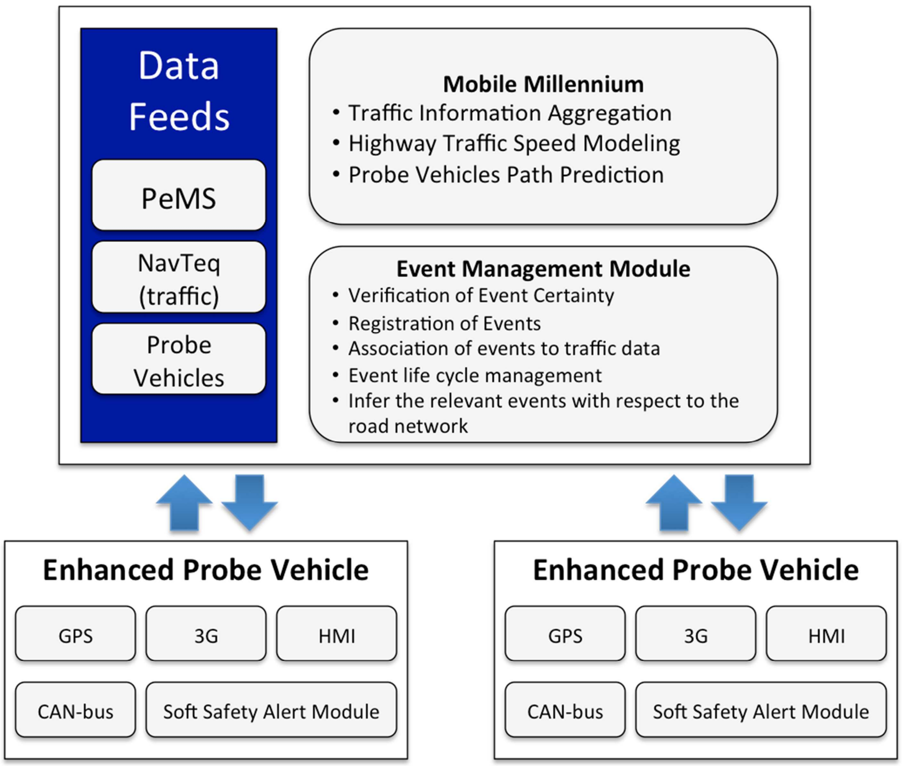


Figure 1.1: System Overview.

The Mobile Millennium server receives data from multiple sources including PeMS (the Caltrans system of freeway loop detectors), Navteq (commercial traffic data), and from various probe vehicles, including the two Enhanced Probe Vehicles (EPVs) developed for this project. When an EPV is detected, the Mobile Millennium server will match the vehicle location to a highway segment and keep a record of the events triggered by that vehicle, such as the triggering of the vehicle's hazard warning lights. The server then returns relevant traffic and event information to each EPV with which it is communicating.

The EPVs for this project included the two California PATH Nissan Altima research vehicles. Vehicle position information is read from the aftermarket GPS receiver, and a connection with the vehicle CAN-bus enables the system to observe the status of the vehicle, including speed and the activation of the hazard warning lights. Communication to the Mobile Millennium server is achieved through a 3G cell modem data connection. Each vehicle continually updates its position, speed, and status, and in return, receives relevant traffic and event information. Onboard the vehicle, the soft-safety alert module processes the vehicle's current status, the traffic information, and the event information, and then it decides whether or not to issue an alert to the driver through the Human-Machine Interface (HMI). Both audible alerts and a visualization of the system status were to be provided to the driver and passengers.

1.3.2 Alert Algorithm Overview

The concept for the soft-safety alert algorithm is fairly straightforward and described in Figure 1.2. The EPV is travelling along the freeway communicating with the Mobile Millennium traffic server. When a traffic slowdown or disabled vehicle is encountered ahead, the alert algorithm calculates the Estimated Time of Arrival (ETA) to the slowed traffic or disabled vehicle. When the vehicle's current speed exceeds the speed of the traffic ahead by more than a preset speed (such as 15 mph) and the ETA to the slowed traffic is less than a preset value (such as 60 seconds), then an audible alert is issued to the driver.

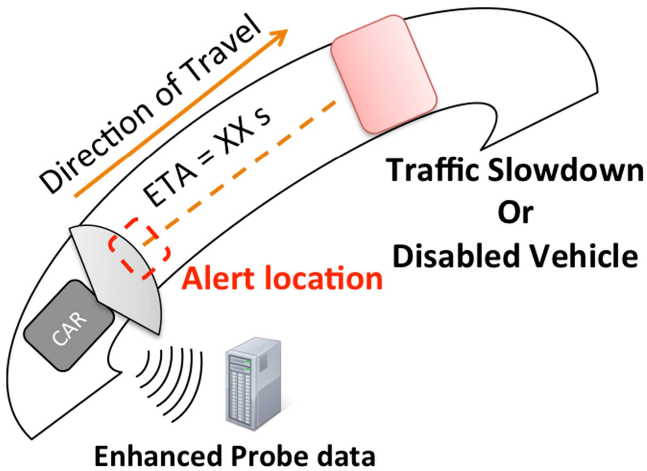


Figure 1.2: Soft-Safety Alert Algorithm Concept.

Three possible audible alerts could be given to the driver. First, if the driver is encountering slowed traffic, an alert would be provided saying, "Slow Traffic Ahead, 35 Miles Per Hour," where 35 mph would indicate the average speed of the traffic ahead. Second, if the driver encounters a disabled vehicle ahead (without a corresponding traffic slowdown), then an alert would be provided saying, "Caution, Disabled Vehicle Ahead." Finally, if the driver encounters both slowed traffic and a disabled vehicle ahead, then the alert would say, "Caution, Disabled Vehicle Ahead, 35 Miles Per Hour," where again, 35 mph would indicate the average speed of the traffic ahead.

2 PATH RESEARCH VEHICLE ARCHITECTURE

2.1 Vehicle Architecture Overview

The California PATH instrumented vehicle Data Acquisition System (DAS) used in the two Nissan Altima EPVs (Enhanced Probe Vehicles) was built using almost entirely Commercial Off-The-Shelf (COTS) components. The system contained two computers which were based on the Mini ITX platform. The first computer was used to gather and record vehicle CAN and external sensor data, communicate with the Mobile Millennium traffic data server, and to process and generate relevant soft-safety alerts for the driver. The second computer was dedicated to video data acquisition. Both of the computers were located in the trunk of the vehicle.

As shown in Figure 2.1, the DAS computer gathered data from both the vehicle manufacturer's CAN and from several other sensors that were added to the vehicles. The CAN data was gathered through a CAN to USB converter. Additional sensors and displays included the following:

- A forward looking Eaton-Vorad EVT-300 Radar
- A 3-axis, MEMSense, Combination Accelerometer and Gyroscope
- A Garmin GPS18x, 5 Hz, D-GPS
- A LandCell-882 3G Wireless Router
- A 7" LCD Display and a PC Speaker for the Driver Vehicle Interface (DVI)

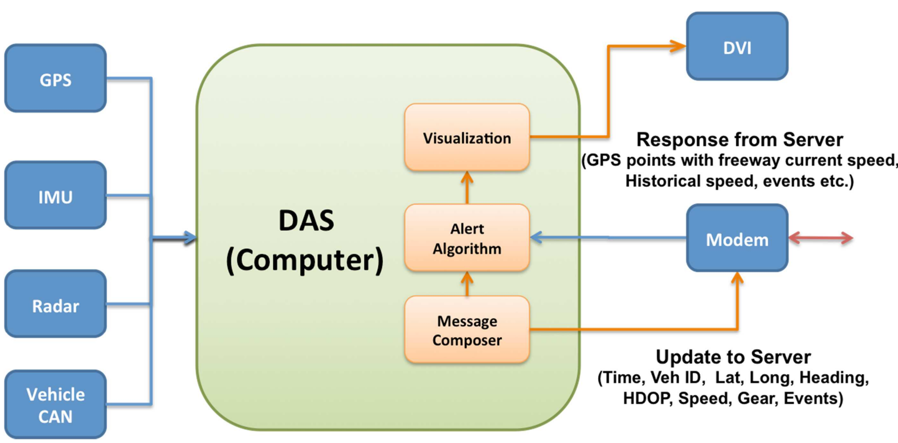


Figure 2.1: Vehicle Architecture Overview.

Other incidental equipment required by the system included a standard Ethernet switch and a USB Digital Input/Output (DIO) device that was used to mute the vehicle's audio system whenever audible alerts were being issued. The audible alerts were issued through a standard PC speaker mounted on the dashboard. A more detailed system diagram is shown in Figure 2.2.

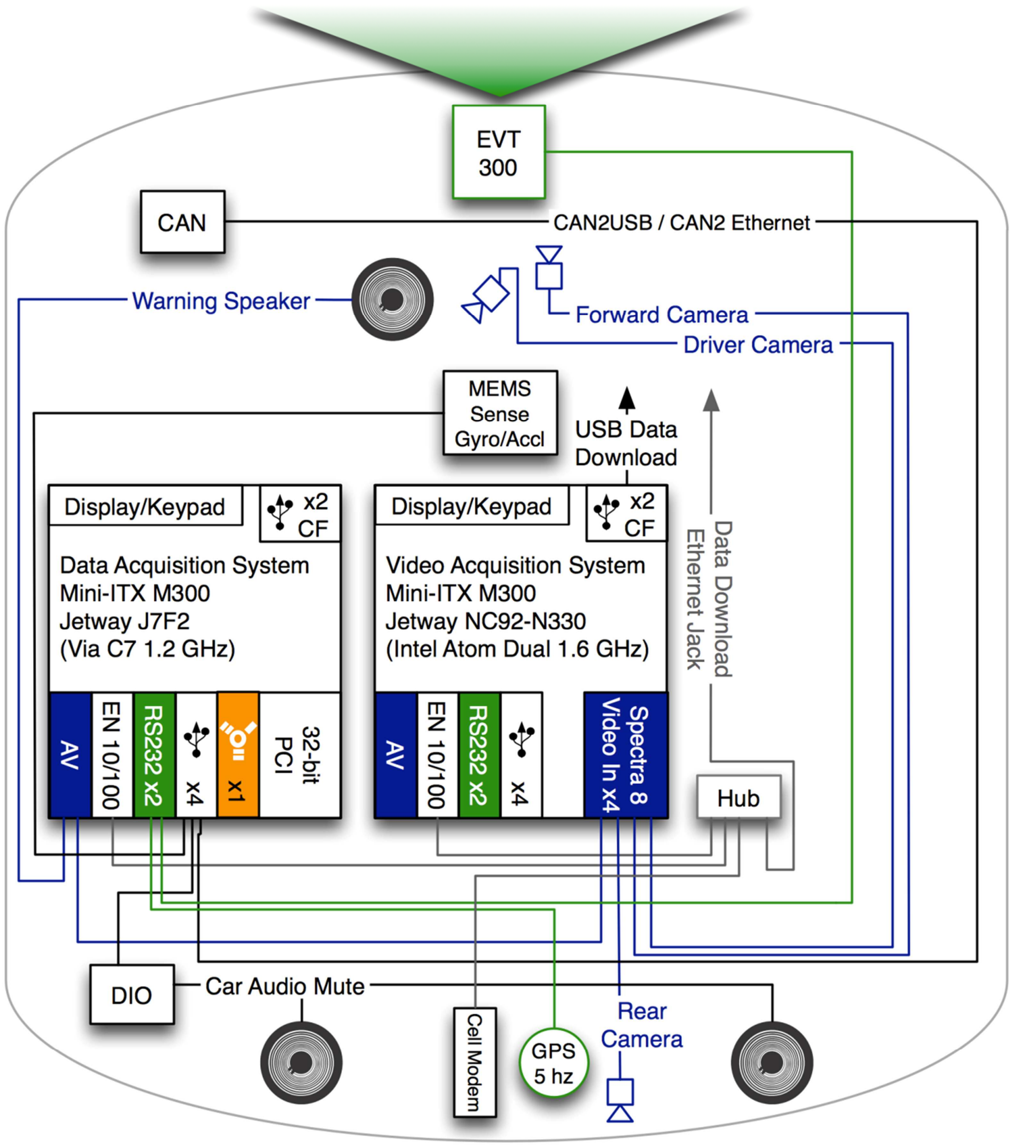
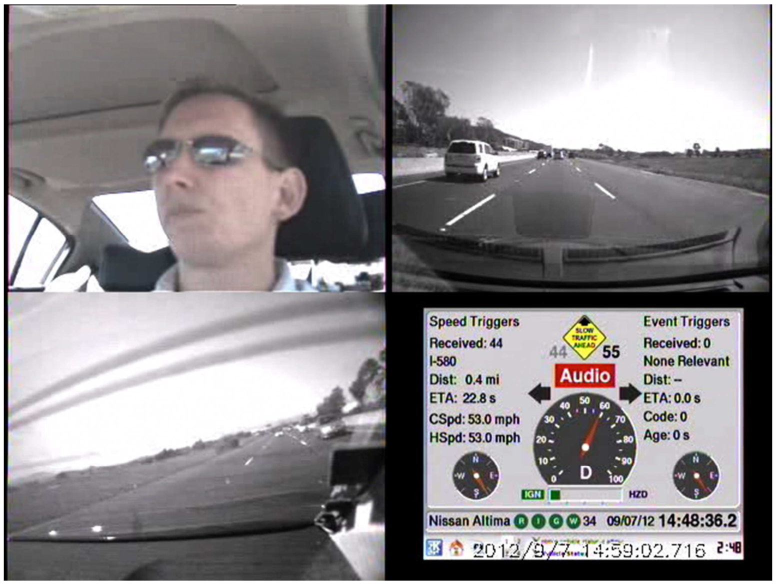


Figure 2.2: California PATH Instrumented Vehicle Systems Diagram.

The VAS (Video Acquisition System) computer could record up to 4 channels of video, each at 320x240 pixels at up to 30 frames per second. Three cameras were placed to capture the forward and rear driving scenes and the driver's face. Since the alert that was being tested was only given using audio and audio was not recorded during the experiment, the fourth video channel was a system status visualization display that was generated by the data acquisition computer. Figure 2.3 depicts an output image as recorded by the video acquisition computer. As shown in this image, a "Slow Traffic Ahead" alert was currently being given to the driver to warn the driver that the traffic speed ahead was expected to be 55 mph.

Driver Face

Forward Scene



Rear Scene

System Visualization

Figure 2.3: Video Acquisition System Output Image.

2.2 Vehicle Software Architecture

The vehicle DAS and VAS software was written in C and C++ and compiled for the Linux operating system. All of the software was custom written by California PATH, but many parts of the software were based on open source drivers and libraries. Due to the availability of various required hardware drivers, the data acquisition computer ran a version of Slackware Linux and the video acquisition computer ran a version of Debian Linux. The software architecture consists of a set of processes running on the each of the data acquisition computers, communicating through the Publish/Subscribe database. (See Figure 2.4.)

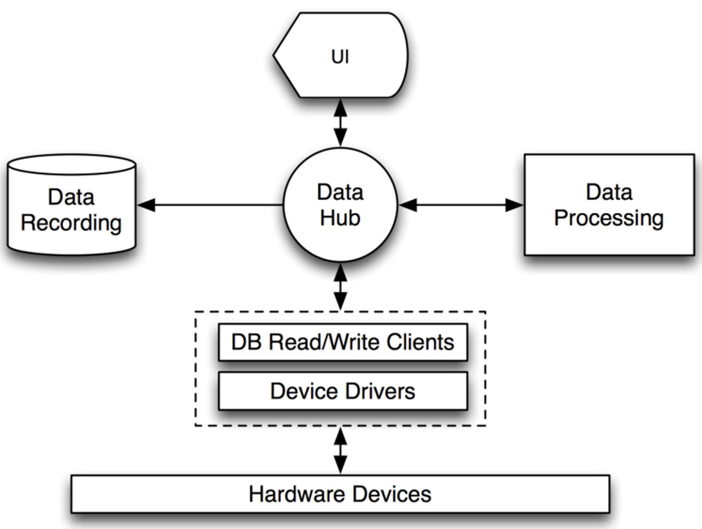


Figure 2.4: California PATH Publish/Subscribe Software Architecture.

The California PATH data hub lies at the center of the software architecture. The data hub is simply a memory space for storing data that can be shared among different processes. At the lowest level of the software architecture, device drivers and data hub client processes were written for each sensor or device to read the data from that device and place it into the data hub.

At the data processing level, various processes were written to use the data that was gathered through the sensors. For example, a communications process would read the necessary data from the data hub and send vehicle position and speed to the Mobile Millennium server over the 3G data modem using the Linux system "curl" command. The response from the server was written to a file. At the same data processing level, an alert process would read and parse the traffic information received from the Mobile Millennium server, read the data gathered by the sensors from the data hub, determine whether or not an alert condition existed, issue an auditory alert if necessary, and write the alert information back to the data hub.

At the UI level, a display process would then read the sensor and alert data from the data hub and use it to draw the system status visualization screen. Finally, at the data recording level, a data recording process would write selected data from the data hub to files stored on the DAS computer. These files could then be downloaded from the vehicle and analyzed later in Matlab.

2.3 Alert Algorithm

The actual vehicle-server communication message formats are discussed, but essentially, the Mobile Millennium server provided the vehicle with two lists of data at each update interval (a configurable parameter): (1) a list of freeway speed locations and (2) a list of EPV event locations. These speed and event locations could each potentially be a trigger point for the alert algorithm. Each trigger contained the following data elements:

Speed Trigger Data Elements

- GPS Latitude
- GPS Longitude
- Heading (compass)
- Current Speed (mph)
- Historical Speed (mph)
- Road Segment Name (character string)

Event Trigger Data Elements

- GPS Latitude
- GPS Longitude
- Heading (compass)
- Event Code (integer)
- Event Age (s)
- Road Segment Name (character string)

During each processing update cycle (a configurable parameter), the alert algorithm used the speed and event trigger data to calculate the distance, estimated time of arrival (ETA), and relative bearing to the trigger point based on the vehicle's current GPS location and speed. Relative bearing was used to discard trigger points that the vehicle already passed. The alert algorithm then simply cycled through the speed and event trigger point lists to determine if any alerts were triggered.

A "Slow Traffic Ahead" alert was triggered if the following four conditions were satisfied:

1. $ETA < ETA_{max}$
2. Current Speed – Trigger Speed > Speed Difference $Threshold_{max}$
3. Trigger Speed < Alert $Speed_{max}$

4. Elapsed Time Since Last Alert > Audible Alert Refractory Period

A "Caution, Disabled Vehicle Ahead" alert was triggered if the following two conditions were satisfied:

1. ETA < ETA_{max}
2. Elapsed Time Since Last Alert > Audible Alert Refractory Period

The lists of speed and event triggers were processed separately; however, when an alert was encountered using one list, the other list was reprocessed using a relaxed setting for the maximum ETA. Thus, if slow traffic was reported 60 seconds ahead, triggering an alert, the event list was reprocessed looking for disabled vehicles up to 120 seconds ahead. If an alert was triggered on both lists, then a combined alert message was provided to the driver.

Most of the alert parameters discussed above were configurable and were stored in a configuration file read at the start of the application. The parameters and their purposed and used values are listed below in Table 2.1, and a few comments on these values follow. The update interval set for this project was set to 10 seconds, but this was primarily based on the fact that the matching of the vehicle location to a freeway segment was being performed by the server and a minimum of three vehicle locations were required to match the vehicle to a roadway segment. Alternative architectures, such as providing the server with a GPS history of the vehicle, rather than a single point, could easily allow the update interval to be raised to 60 seconds. Typically, the Mobile Millennium server tried to provide the vehicle with about 5 minutes of data during each update.

Table 2.1: Configurable Parameters Related to the Alert Algorithm.

Parameter	**Recommended Settings**	**Demo Settings**
Update Interval (seconds)	60 s	10 s
Processing Update Rate (milliseconds)	Depends on GPS Update Rate	200 (ms)
Event Age (seconds)	Depends on Event Type	180 s
Relative Bearing to Trigger/Event	Roadway Geometry Dependent	± 90 Degrees
Max ETA to Slowed Traffic or Event	60 s	30 s
Maximum Alert Speed	50 mph	60 mph
Max Speed Difference Threshold	15 mph	5 mph
Audible Alert Refractory Period	120 – 180 s	45 s

The alert algorithm processing rate was set based on the update rate of the vehicle's GPS (200 ms) since the algorithm can only recalculate the distances and ETAs to each of the trigger points when there is new GPS information available. The 60 second recommendation for the maximum ETA at which to alert the driver was based on the results of the Networked Traveler project (Nowakowski, et al., 2011, 2012). In that study, alerts given 90+ seconds from the traffic slowdown were often rated as early, while alerts coming less than 60 seconds from the traffic slowdown were often rated as late. Similarly, the recommended settings for the maximum alert speed, maximum speed difference threshold, and audible alert refractory period came from the Networked Traveler project results.

2.4 System Status Visualization

Two screen captures of the system status screen are shown in Figure 2.5 and Figure 2.6. During the demonstration the system status screen was shown on a 7" LCD display mounted in the vehicle for the driver and passengers to observe. The information on the screen was dynamically updated by the DAS computer, and the purpose of the screen was to help visualize the system status and the flow of information from the Mobile Millennium server to the vehicle.

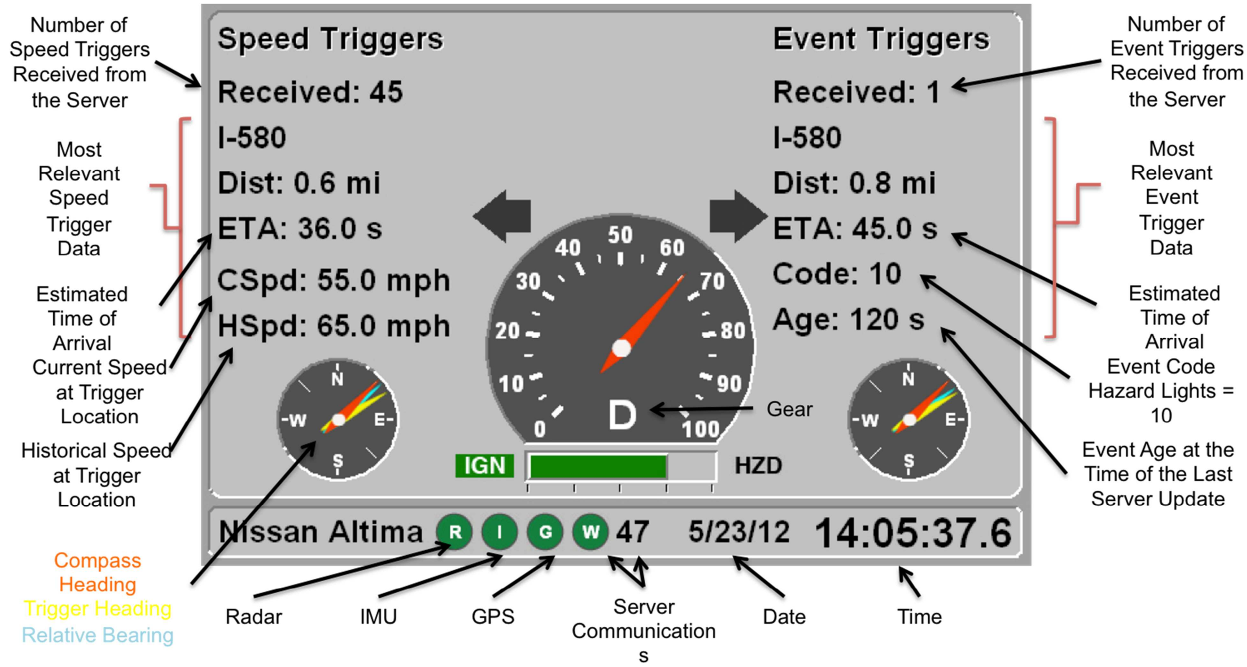


Figure 2.5: DVI Pre-Alert System Status Visualization Screen.

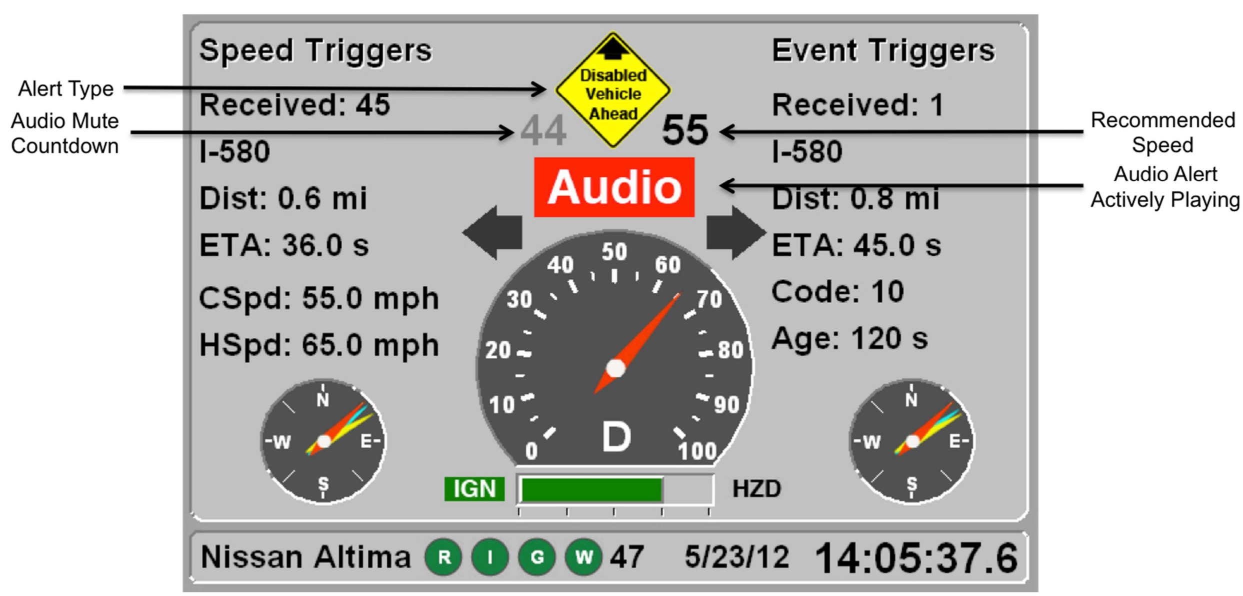


Figure 2.6: DVI Post-Alert System Status Visualization Screen.

The first screen capture shows the system in a pre-alert state. In this figure, the vehicle is travelling at 65 mph, and since the start of the trip, 47 communication attempts have been made with the Mobile Millennium server. Currently, 45 active speed triggers had been received, and the closest speed trigger was 0.6 miles away with an ETA of 36 seconds. Additionally, one event trigger had been received, and it was 0.8 miles away with an ETA of 45 seconds. The second screen capture shows the system in the same state after an alert had been triggered. The triggering of the alert adds a few more data elements to the screen such as the alert type, the recommended speed, and an indication that an audible alert is currently playing. Based on this screen, an auditory alert saying, "Caution, Disabled Vehicle Ahead, 55 Miles Per Hour," would currently be playing for the driver. Again, the visual screen was designed with the purpose of providing a visualization of the demonstration, and not as a prototype of an interface that would be provided by a vehicle manufacturer to a typical driver.

3 MOBILE MILLENNIUM TRAFFIC DATA ARCHITECTURE

3.1 Server Architecture

The Mobile Millennium server architecture, as depicted in Figure 3.1, can be broken into two main processes: (1) the updating of the current traffic data through the Mobile Millennium Highway Model, and (2) the processing of the Enhanced Probe Vehicle (EPV) information requests and events. Highway speed estimation operates on a perpetual basis through the Mobile Millennium Highway Model, beginning with data from the Caltrans PeMS (Performance Measurement System) inductive loop sensors near the road surface. These sensors provide flow and occupancy data, which then get filtered and processed to estimate current highway speeds. Other data sources, such as probe vehicles and commercial traffic data (from Navteq) can then be fused with the PeMS data to provide refined freeway speed estimates in the highway model.

The processing of the EPV information requests can be broken into three parts: vehicle path inference, vehicle event register, and return message composition. When an EPV message is received by the server, it is first parsed for vehicle location and speed. The GPS location provided by the vehicle, along with several past locations provided by the same vehicle, is processed using a path inference algorithm to basically match the current vehicle location to a specific roadway segment in the Mobile Millennium Highway Model. In parallel, the EPV message is also processed by the event register, which is responsible for storing, cancelling, and decaying EPV events. Finally, the message composer process composes the return message to the EPV, which includes both speeds and events along the vehicle's predicted path.

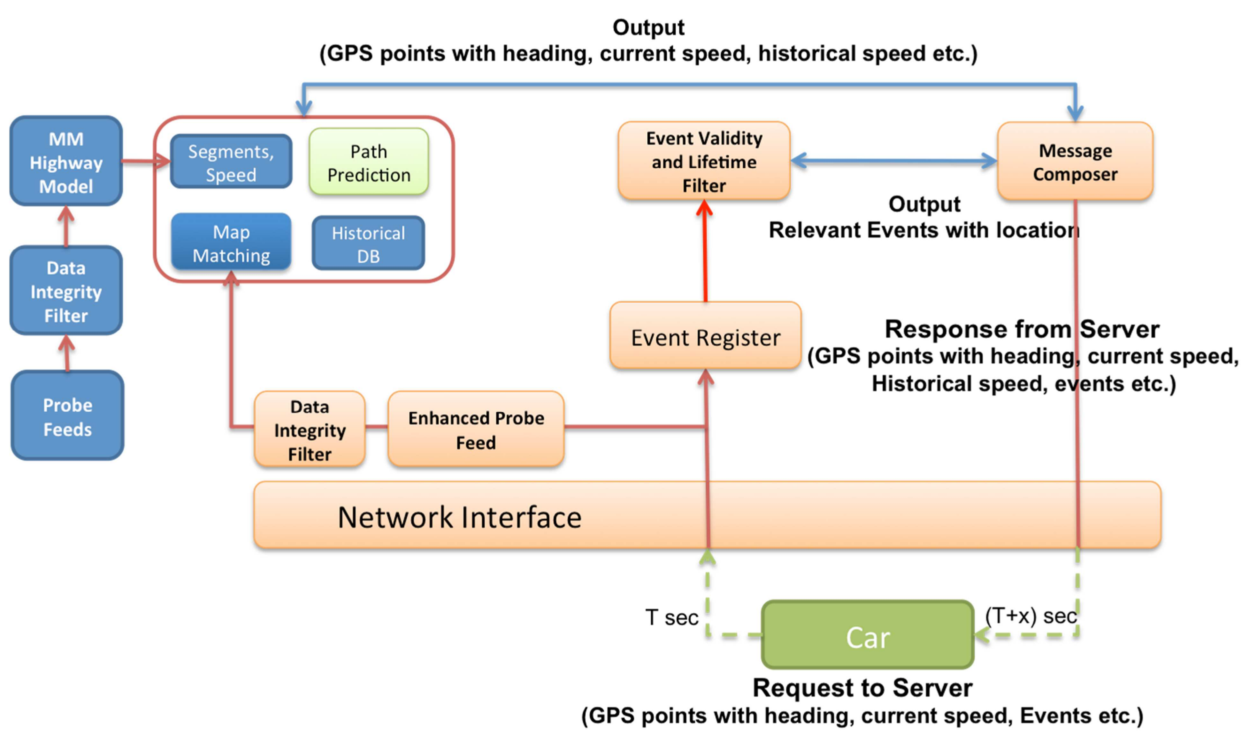


Figure 3.1: Server Architecture.

3.2 Vehicle-Server Message Exchange

The web-based communication between the EPVs and the server utilized the cURL command. Basically, the EPV posted a web request with its current data, and the server responded with a formatted text response. For expediency in prototyping at this stage in the development process, all communications were simply plain text encoded with individual parameters being comma delimited. The format of the vehicle information request message to the server is detailed in Table 3.1, and the format for the server response message is detailed in Table 3.2.

Table 3.1: Vehicle Message to Server.

Parameter	Description
Current Time	Unix time in seconds since epoch (January 1, 1970)
Vehicle ID	Character string
Latitude	GPS position latitude
Longitude	GPS position longitude
Heading	GPS compass heading (0 – 360 degrees)
HDOP	GPS measure of accuracy
Vehicle Speed	Obtained from CAN-bus (m/s)
Vehicle Gear	Obtained from CAN-bus and converted to an integer from 0 – 4 0 Unknown | 1 Park | 2 Reverse | 3 Neutral | 4 Drive
Vehicle Event Code	Events obtained from CAN-bus 0 No Event | 10 Hazard Warning Lights Active

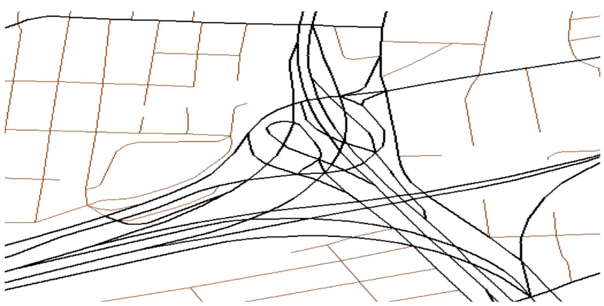
Table 3.2: Server Response Message to Vehicle.

Parameter	Description
Current Time	Unix time in seconds since epoch (January 1, 1970)
Number of Speed Triggers	Integer
Latitude	GPS latitude
Longitude	GPS longitude
Heading	Freeway direction compass heading (0 – 360 degrees)
Current Speed	Current freeway speed for this location
Historical Speed	Historical freeway speed for this location/time (from Navteq)
Freeway Segment Name	Character String
Number of Event Triggers	Integer
Latitude	GPS latitude
Longitude	GPS longitude
Event Code	Integer (10 for Hazard Warning Lights Active)
Event Age	Seconds
Heading	Last known vehicle heading (0 – 360 degrees)
Freeway Segment Name	Character String

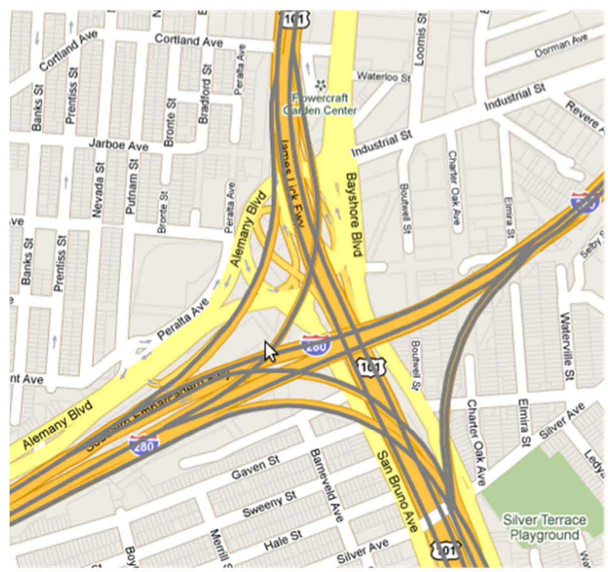
The server response messages were variable in length, dependent upon the number of speed and event triggers returned to the vehicle. Typically, the server tried to predict the vehicle path out for at least 5 minutes of travel, and this resulted in anywhere from 50 to 100 speed trigger points being transmitted during each message. If the vehicle's current location could not be matched to a highway model freeway segment, then zero speed triggers were returned.

3.3 Mobile Millennium Highway Model Graph

The Mobile Millennium Highway Model Graph is essentially a simplified version of the highway maps provided by Navteq. As shown in Figure 3.2, the Navteq maps include links and nodes for on-ramps, off-ramps, and frontage roads that are all considered as part of the freeway system. Additionally, the intersection of multiple links can result in the generation of many additional short links to handle the various travel path connection options. The Mobile Millennium Highway Model Graph simplifies the list of links and nodes to include only the freeway segments, and the intersection of multiple links becomes a single node. However, longer Navteq links may be subdivided in the Mobile Millennium Model Graph so that freeway speeds can be interpolated between known points using the cell transmission model.



Navteq Map



Mobile Millennium Highway Model

Figure 3.2: Comparison of Navteq and Mobile Millennium Maps.

Due to computer processing power constraints and the prototype nature of this project, the actual server used in the demonstration contained only a subset of the freeways in the San Francisco Bay Area near and around the Richmond Field Station in Richmond, CA. As shown in Figure 3.3, the coverage area included I-580 and I-80, which were the primary roads used in the demonstration.

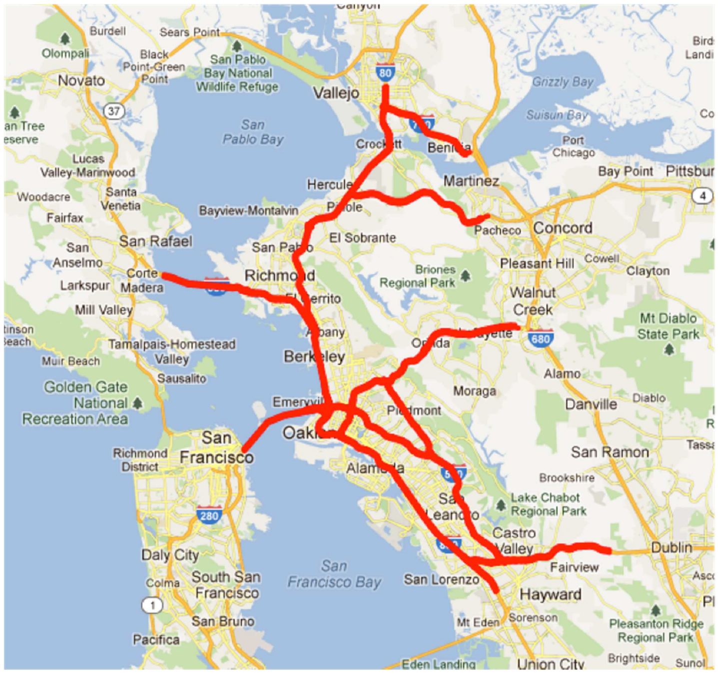


Figure 3.3: Mobile Millennium Highway Model Used During the Demonstration.

3.4 Vehicle Path Inference and Future Path Prediction

After receiving a location update from an EPV, the server used a path inference filter (PIF) algorithm to match the probe vehicle to a specific link of the Mobile Millennium Highway Model Graph. The PIF was not simply a direct map matching algorithm because of the inaccuracies in the GPS location provided by the probe vehicle. The raw GPS coordinates tend to not clearly follow a highway or arterial road, rather the GPS coordinates "bounce" from side to side and forward or backward. This is especially problematic for an arterial or city map due to the frequent intersections and traffic signals; however, even on a highway, the GPS coordinates can frequently provide only a probable location, especially when frontage roads and overpasses are considered.

To counter the issues with simply map matching a single GPS location, the PIF kept track of three previous GPS coordinates for each vehicle, along with a record containing the last known Navteq map link that was matched to the vehicle. The accuracy of the PIF increases both with the number of points per unit time (frequency) and with the number of points available to look back to; however, there was also a trade-off between accuracy and processing speed. Thus, for this demonstration, the probe vehicle needed to provide at least three updates to the server before a match could be made. Furthermore, when a location match is made, the PIF also returns a probability of accuracy. If the probability was less than 65 percent, then the match was discarded and no freeway speeds were returned to the EPV.

Once the probe vehicle was matched to a link on the Mobile Millennium Highway Model Graph, the vehicle's possible future path was calculated by "walking" the model graph links in the forward direction to return a set of all model graph link IDs within a five-minute driving time. Once the list of model graph link IDs was gathered, a SQL query was performed to get the associated highway model current and historical speeds for each link. The message composer process then formatted the list of links and speeds to return to the EPV.

3.5 Enhanced Probe Vehicle Event Register

Along with GPS location and speed, EPVs also provided an event code, either a 0 for no event, or a 10 for the activation of the hazard warning lights. The event register was used to both store and decay events supplied by the EPVs. The event register was basically a database of vehicle IDs, locations, and events. The time was tracked for each event in the register and events could be cancelled in one of two ways. First, the EPV could provide a subsequent update that cancels the event. Second, each event was given a finite decay time, for example, 3 minutes for hazard light activation.

If a vehicle was sitting on the side of the road with its hazard lights on, it would broadcast this event state on each update to the server. Each new update from a vehicle was checked by the event register. If the vehicle cancelled the event, then the event status was removed from active in the event register. If an event was verified as continuing, then the event was reaffirmed as active and the decay time was reset to the default for that event type. If a vehicle stopped broadcasting while an event was still active, the server would keep the event active until the decay time for that event was exceeded, after which the event would be labelled as inactive.

The event register used the PostgresQL database with post-GIS extensions. On each EPV update, the message composer process executed a SQL (structured query language) command to return all events within five minutes driving distance as the crow flies and not older than 3 minutes. The message composer also filtered based on the vehicle ID to make sure that it only returned events from other cars. This query could easily be filtered in the future to return only events within a forward “cone” of travel by discarding events for which the disabled vehicle’s Navteq link ID does not match any of the speed trigger link IDs that are being returned to the EPV.

4 SIMULATION RESULTS

4.1 Simulation Overview

As part of the development and testing process, a simulation tool was written using the Matlab programming and analysis language and environment. Primary purpose of the simulation was twofold: (1) to playback prerecorded vehicle data, thus, simulating a vehicle moving along the freeway network, and (2) to simulate disabled vehicles on the freeway for the two PATH research vehicles acting as EPVs to encounter. One of the advantages of the simulation tool was that the vehicle-server communication was handled in exactly the same way as it was done on the vehicle, so the simulation tool was invaluable in debugging the vehicle-server communications protocols. Furthermore, the simulation tool was able to record and save the entire output of the communications received from the server, which then provided for a more detailed analysis of the system performance than what was available from the data set that was saved on the vehicle.

The simulation tool loaded pre-recorded vehicle data from any of the California PATH research vehicles, and played back the vehicle route (GPS location and vehicle speed) in real time, substituting the current clock time for the time at which the data was actually recorded. The simulated vehicle trip that was analyzed was recorded on July 15, 2010, at about 11:00 PM, by one of the California PATH Nissan Altima research vehicles. The route taken by the vehicle is shown in Figure 4.1, and this particular route was often used in simulation because the research vehicle both started and ended the trip at the Richmond Field Station, allowing the simulated trip to be looped without an apparent jump in the vehicle position.

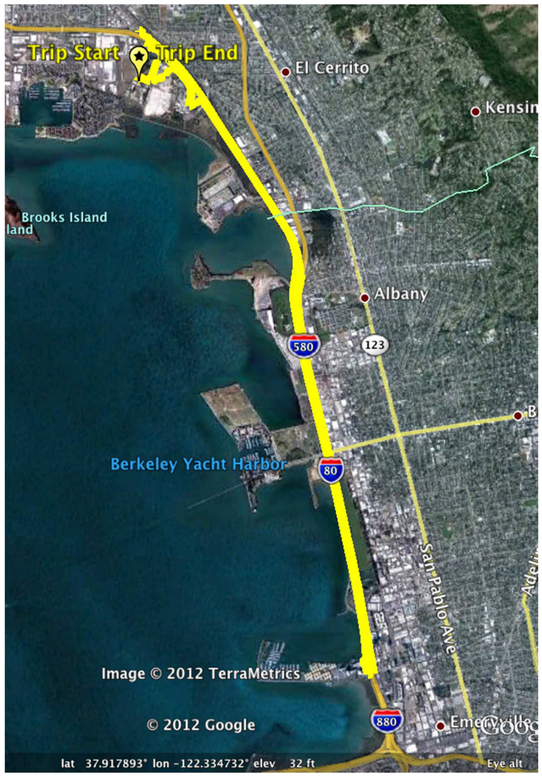
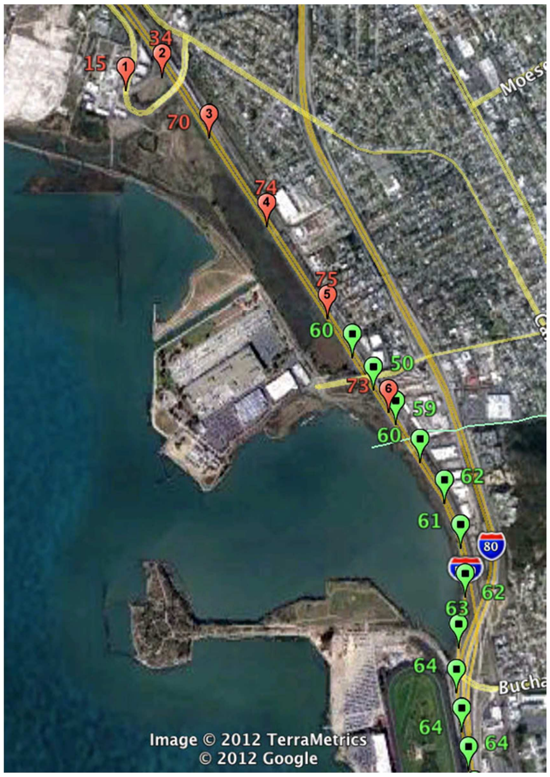


Figure 4.1: Simulated Trip Vehicle Route.

4.2 Simulation Results

4.2.1 Entering A Freeway – Initial Server Data Update

The actual simulation discussed in this section was run at approximately 4:00 PM on a Wednesday to provide some evening traffic along the test route using the pre-recorded trip data previously discussed. The vehicle-server communication rate was set to update every 15 seconds, and over the 17-minute route, there were 70 vehicle-server communication transactions. Figure 4.2 provides an overview of the data provided to the vehicle during the first update from the server. Shown on the left side of Figure 4.2 are the vehicle positions during the first six communication attempts, plotted in red bubbles where the number in the bubble depicts the communication attempt. The green bubbles with a square depict freeway speed points, and the yellow bubbles with diamonds depict simulated disabled vehicle locations. The numbers next to the bubbles indicate either the vehicle speed (red) or the freeway traffic speed (green) in miles per hour (MPH).



Initial Vehicle Positions (Red)   Freeway Speed Data Returned by the Server (Green)

Figure 4.2: Geographic Plot of First Set of Data Received by the Simulated Vehicle.

On the first update to the server, the simulated vehicle had exited the Richmond Field Station and was on Bayview Avenue about to get onto the freeway entrance ramp headed east on I-580 towards Berkeley. With the vehicle updating its position every 15 seconds, it took 5 additional updates, during which time the vehicle traveled 1.3 miles or about 1 minute and 15 seconds, for the Mobile Millennium server path inference algorithm to correctly place the vehicle on I-580 and provide the vehicle with data on the freeway speeds ahead. By this time the vehicle had reached Central Avenue, the next exit on the freeway. Similar data delays were seen whenever the vehicle entered the freeway.

Another issue seen with the initial placing of the vehicle onto the highway was that the path inference algorithm would sometimes place the vehicle on the correct highway, but facing the wrong direction, i.e., all of the freeway speed locations sent to the vehicle were for the opposite direction traffic. However, when this occurred, it was usually corrected within the next update or two at the most.

4.2.2 Loss of Vehicle Match to a Highway Link

Looking at the right side of Figure 4.2, once the server provided the vehicle with data, it provided data along I-580 east and I-80 west (which are in fact the same direction, southbound) for about 5 miles, all the way down to Emeryville. Where the freeway branched off, points were provided along all the branches. This strategy appeared to work well. Also noted in Figure 4.2, the spacing between highway speed points was on the order of about 200 m or 650 feet. A full 5 minutes of driving time at roughly 60-65 mph resulted in 40 to 50 speed data points being transmitted to the vehicle on each update. As the vehicle approached a freeway junction, the number of data points that needed to be transmitted to the vehicle increased since data for each branch was required. During the simulation, the maximum number of data points received in any single update was 125.

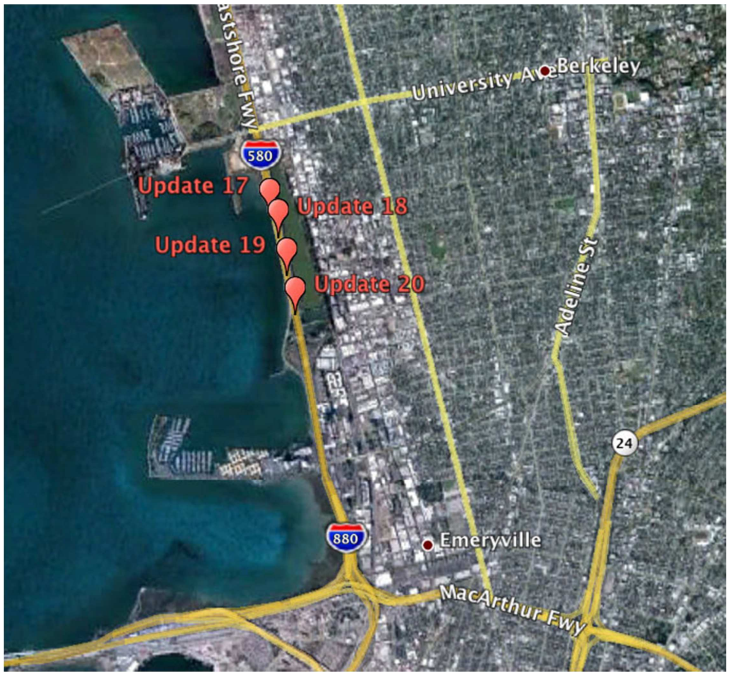


Figure 4.3: Location Where Simulated Vehicle Position Was Lost By Path Inference Algorithm.

Examining the output of the simulation data also revealed a number of locations where the path inference algorithm temporarily lost the position of the vehicle on the Mobile Millennium Model Graph. In Figure 4.3, the vehicle location on the model graph was lost during updates 18 and 19, and then regained on update 20. At this location, it is possible that the GPS might have placed the vehicle on the frontage road next to the freeway, and the path inference algorithm assumed that the vehicle had exited the freeway at University Avenue and got onto the frontage road. During the simulated trip, the vehicle location was lost three times while it was validly travelling on a highway. Each loss only lasted for one or two server updates (30 to 45 seconds).

Whenever the path inference algorithm failed to place the vehicle on a link in the Mobile Millennium Highway Model Graph, the server returned no freeway speed information (zero speed trigger points). To account for this behavior, the system in the vehicle had to maintain a copy of the last set of speed trigger points. This allowed the vehicle to keep processing previously received speed trigger data until the path inference algorithm on the server re-established the vehicle's position on the highway and provided it with new highway speed information.

4.2.3 *Soft-Safety Alert Algorithm Testing*

The purpose of the simulation was to test the communication protocols, to test data manipulation algorithms before putting them on the vehicle, and to characterize the behavior of the freeway speed updates. In the vehicle, the alert algorithm ran continually each time the GPS updated the vehicle position, even between communication updates from the server. However, in the simulation, the only data that was recorded occurred at the moment of the communication update from the server. Additionally, the simulated vehicle speeds and travel times were unrelated to the current traffic conditions on the freeway at the time of the simulation. Still, based on the simulation data, there would have been several "Slow Traffic Ahead" alerts issued by the vehicle, and one such scenario is shown in Figure 4.4.

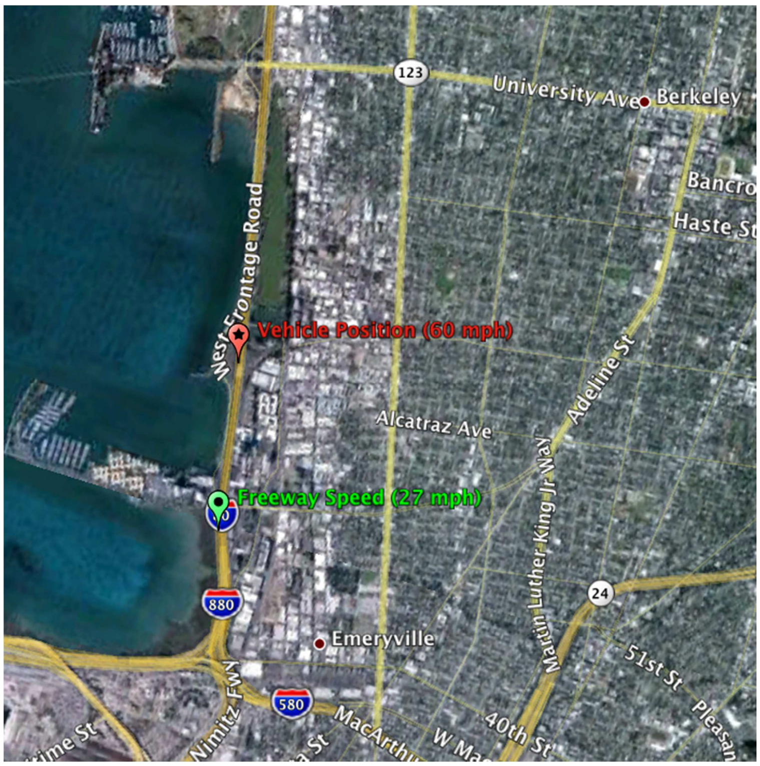


Figure 4.4: Alert Condition Encountered in Simulation.

In Figure 4.4, the vehicle position during the 21st communications update (shown in red) from the Mobile Millennium server provided a speed trigger (shown in green) with an ETA of 45.5 seconds from the current vehicle position. At the time in the simulation, the vehicle was travelling at 60 mph, while the speed at the trigger point near Powell Street in Emeryville, CA, was only 27 mph. Given the alert algorithm settings that were used in the Networked Traveler project (i.e., ETA less than 60 seconds and speed difference greater than 15 miles per hour), a "Slow Traffic Ahead, 30 Miles Per Hour" auditory alert would have been triggered.

5 DEMONSTRATION RESULTS

5.1 On-The-Road Test Overview

An on-the-road test of the system was conducted with the two California PATH research vehicles at approximately 2:45 PM on a Friday to provide some traffic along the test route. The Silver Nissan Altima drove from the RFS towards Albany and eventually pulled over to engage its hazard lights along I-80 W. The Gray Nissan Altima followed the same route about 1-2 minutes later. After passing the Silver Nissan Altima, both vehicles exited the freeway at University. The gray vehicle waited several minutes for the silver vehicle to turn around, and enter I-580 W headed back towards the RFS. The silver vehicle again pulled to the side of the road just past the RFS on the I-580 W and put on its hazard lights. The gray vehicle followed a few minutes later.

Figure 5.1 provides an overview of the test route, the locations where the first vehicle turned on its hazard lights to mimic a disabled vehicle (shown in yellow), and the locations where alerts were given to the second vehicle (shown in red). In addition to the two “Caution, Disabled Vehicle Ahead” alerts, the following vehicle also received two “Slow Traffic Ahead” alerts.

Figure 5.1: On-The-Road Test Route and Alert Locations.

For both vehicles, the vehicle-server communication was set to update every 10 seconds, and over the 30-minute test route, there were approximately 190 vehicle-server communication transactions. The mean time between server updates was 10.5 (SD 2.9) seconds, and the maximum time between successful server updates was 23.4 seconds. During the on-the-road test, the “Slow Traffic Ahead” alerts in the gray following vehicle were configured to be triggered with a speed differential greater than 5 mph, a maximum alert speed of up to 70 mph, and at an ETA to the location of the slowed traffic below 30 seconds. The gray following vehicle received an average of 25 (SD 19.2) speed triggers on each update from the server, with a maximum number of triggers received at 69. Since the silver car was the only disabled vehicle in the system, the gray following vehicle only ever received one disabled vehicle trigger.

5.2 Disabled Vehicle Ahead Alerts

During the on-the-road test, the silver vehicle turned on its hazard warning lights twice, simulating a disabled vehicle. There were two metrics of interest that could be gathered from this test. The first metric of interest was how long it took for the event to propagate through the system and reach the following vehicle. The propagation of an event through the system should be primarily related to the communication update rate, although there could also be additional system delays. Based on the 10-second communication update rate set on the vehicles for this test, it would be expected to take at least 10 to 20 seconds (or two asynchronous update cycles) for the silver vehicle to turn on its hazard warning lights and for the following gray vehicle to receive that information from the server in the form of an event trigger. Based on the data collected as shown in Table 5.1, it took between 20 and 40 seconds (two to three update cycles) for the hazard warning light activation to propagate through the system and be received by the following vehicle.

Table 5.1: Hazard Light Event Propagation Through the System.

Hazard Light Activation Event	Transmitted to Server	Received by Following Vehicle	Following Veh. Distance	Following Veh. ETA
1	9 (s)	31 (s)	0.3 mi	34.1 s
2	3 (s)	25 (s)	0.8 mi	47.0 s

The second metric of interest centered around the alert activation, and how closely the alerts activated to the parameter settings used for the test. For the “Caution, Disabled Vehicle Ahead” alert, the key parameter to trigger an alert was simply that the ETA be less than 30 seconds. The accuracy of the alert triggering would be expected to depend on the vehicle’s GPS update rate (set to 200 ms during this test) and overall accuracy. As shown in Figure 5.2, the audible alert was triggered almost exactly when the calculated ETA to the disabled vehicle reached 30 seconds. In the first alert case, the following vehicle was travelling between 18 and 25 mph in stop-and-go traffic, and in the second case, the following vehicle was travelling at a fairly constant 62 mph (SD 2.1 mph).

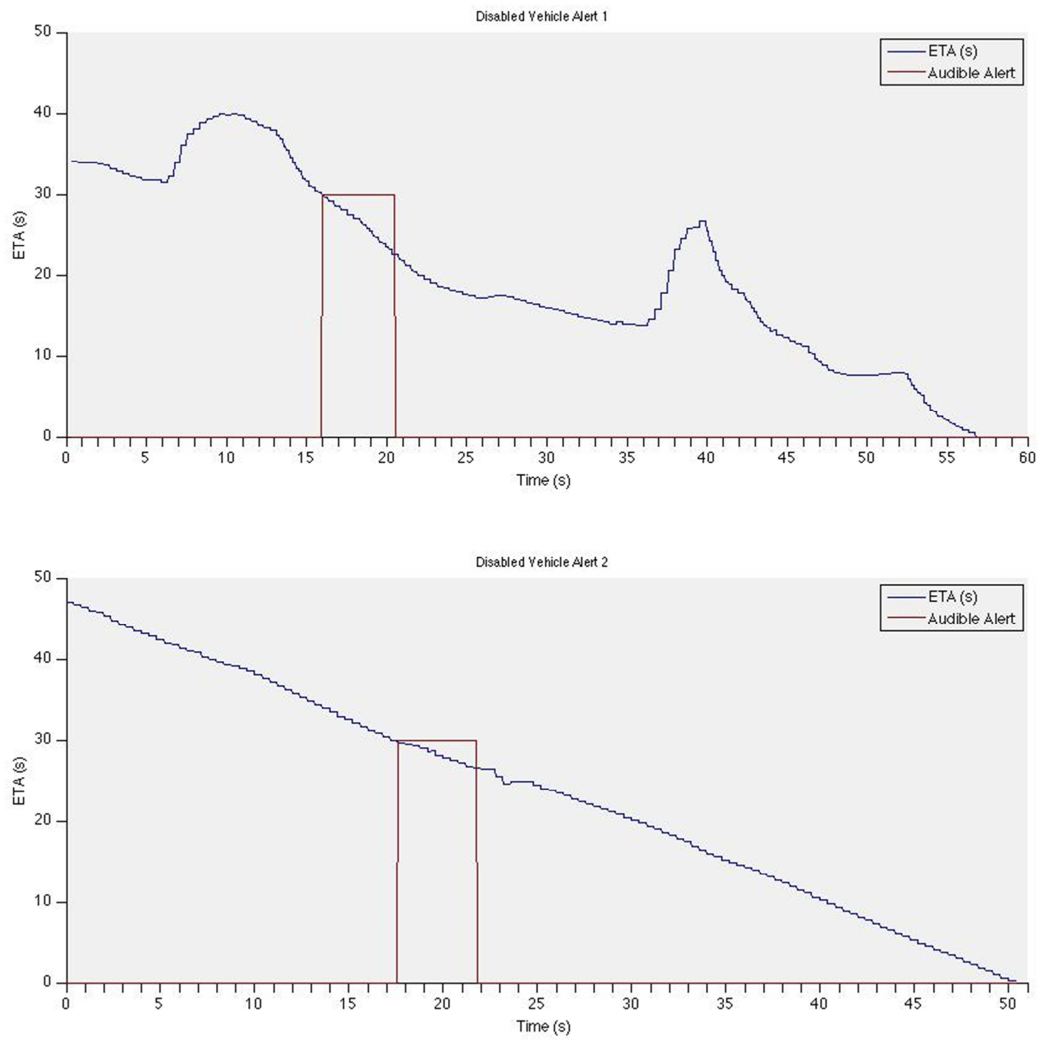


Figure 5.2: Disabled Vehicle Alert Activations as a Function of ETA.

5.3 Slow Traffic Ahead Alerts

During the on-the-road test, the gray following vehicle also received two “Slow Traffic Ahead” alerts, both unrelated to the disabled vehicle. At the first “Slow Traffic Ahead” alert, the gray vehicle was travelling at 59.8 mph when it received a set of speed triggers from the server. One of the speed triggers was 0.4 miles away (ETA 23.2 s) with a freeway speed of 53 mph. As shown in Figure 5.3, as soon as the speed trigger was received and processed by the alert algorithm, an audible alert was issued to the driver suggesting a speed of 55 mph. Although the alert algorithm functioned as designed, this case illustrates the problems that can occur when it takes too long for the vehicle to be matched to a freeway segment. The vehicle had been travelling on the freeway for almost 65 seconds (6 communications update cycles) before the path inference algorithm correctly placed the vehicle on the freeway and provided speed data. Thus, by the time that the speed data was provided to the vehicle, the ETA to the slowed traffic was below the 30-second ETA alert threshold.

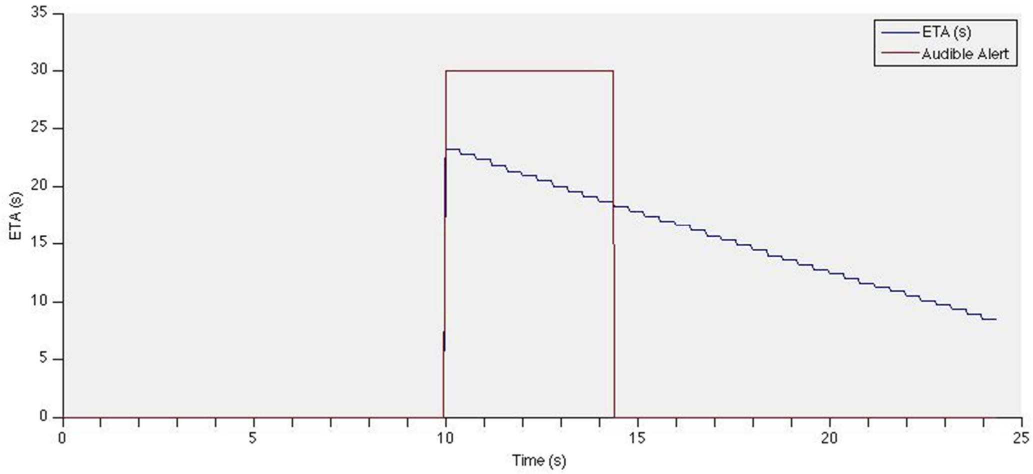


Figure 5.3: First "Slow Traffic Ahead" Alert as a Function of ETA.

At the second “Slow Traffic Ahead” alert, the gray vehicle was travelling at 67.8 mph and had been receiving about 55 speed triggers from the server on each update prior to the alert. As shown in Figure 5.4, the alert algorithm was processing the speed difference and ETA to each trigger until the vehicle passed the trigger, and the current freeway speed at each trigger was being reported around 63 mph. In the 10 seconds prior to the alert, the vehicle speed increased from 64 to almost 68 mph, and when the 5 mph speed difference threshold was exceeded, an alert was correctly triggered with an ETA of 16.7 seconds. This case illustrates how an alert could be triggered later than the desired ETA (30 seconds) when the vehicle is in the process of accelerating.

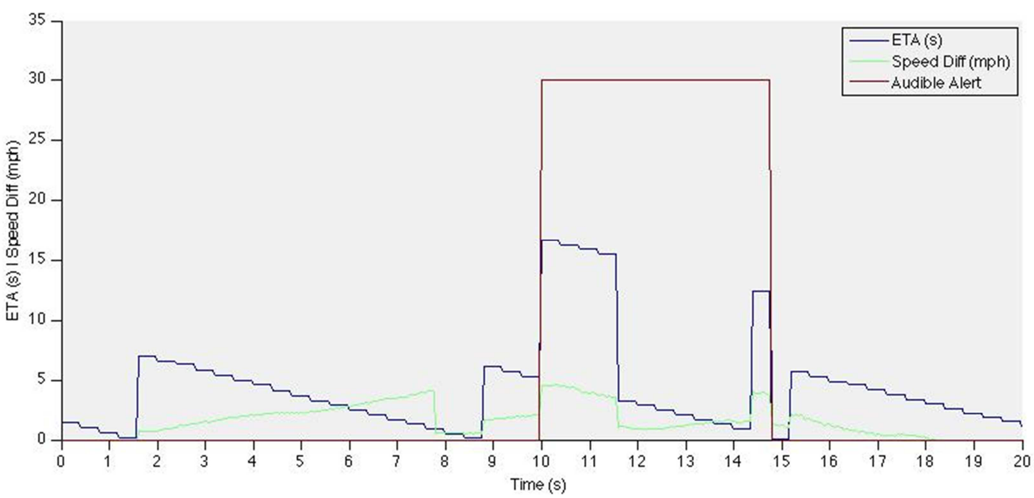


Figure 5.4: Second "Slow Traffic Ahead Alert as a Function of ETA.

6 SUMMARY AND CONCLUSIONS

6.1.1 Project Summary

The U.C. Berkeley, California PATH-Renault collaboration, detailed in this report, built upon the Mobile Millennium and Networked Traveler projects that were conducted from 2007 to 2011 as part of the US DOT's SafeTrip-21 Initiative. The Mobile Millennium project provided a platform for aggregating traffic information across various sources, including infrastructure sensors, commercial data feeds, probe vehicles, and probe cell phones. The Networked Traveler project provided the California PATH instrumented research vehicle platform to both deliver vehicle probe data back to the infrastructure and to generate ADAS alerts to the drivers of those vehicles.

The main theme of this collaboration project was to demonstrate the potential to create Enhanced Probe Vehicles (EPVs) by merging vehicle CAN-bus data with the typical GPS data that is provided by normal probe vehicles. EPVs could then provide more accurate speed information to the traffic servers along with additional information that only the vehicle knows, such as current gear, hazard light warning activation, or even airbag deployment events. During SafeTrip-21, the two California PATH research platforms were built independently, so the first goal of this project was integrate the two systems, allowing the instrumented research vehicles to communicate with the Mobile Millennium traffic aggregation servers. The second goal was to enhance the data gathered and stored by the Mobile Millennium traffic aggregation server to include the enhanced information provided by the EPVs, specifically, to include hazard warning light activation events. The third and final goal of this project was to integrate the information provided by the EPVs to the traffic server into soft-safety alerts that could be provided to the drivers of the EPVs and demonstrate the prototype system.

Overall, the construction of the system prototype and subsequent demonstration was a success. The two California PATH Nissan Altima research vehicles provided the Mobile Millennium server with EPV data including GPS, speed, gear, and hazard warning light activation events. On each communications update to the Mobile Millennium server, a path inference algorithm placed the vehicle on a link of its highway model, freeway traffic speeds were provided to the vehicle every 200 m up to a driving distance of about 5 minutes, and hazard warning light events were recorded, processed, and communicated back to the EPVs. The EPVs then used the freeway speed and event data to provide the drivers with soft-safety alerts such as "Slow Traffic Ahead" or "Caution, Disabled Vehicle Ahead."

6.1.2 System Architecture Conclusions

Based on the analysis of the data saved by both simulated and actual vehicles driving a route from the Richmond Field Station down to Powell Street in Emeryville, CA, there were several issues noted with the implementation of the system, and there was room for future improvement. The largest issues with the prototype system was related to the implementation of the path inference or matching algorithms. First, it often took too long for a vehicle to be placed on a freeway and to get freeway speed information flowing to the vehicle. Even with a 10-second update rate, it often took 60 to 90 seconds before the vehicle was correctly placed on a freeway segment. Second, there were frequent drop-outs when the vehicle was incorrectly shown to have exited the freeway.

While there are a number of ways to solve this issue, each comes with a series of trade-offs. In the implementation of this prototype, the knowledge of the Mobile Millennium Highway Model Graph was only contained on the server, and this dictated that the map matching of the current vehicle location to a link on the model graph be performed on the server. Currently, the path inference algorithm uses 3 points to compute a fix on the vehicle. More points could be used, but this strategy would require more computational time and power on the server as well as compounding the initial freeway entrance problem by requiring longer waiting times (more update cycles) for the first fix of the vehicle onto the highway. Some of this delay might be mitigated by changing the communication message architecture to include the vehicle transmitting a path history over some time frame on each update, rather than simply transmitting the current location at the time of the update. Transmitting a path history would increase the size of each update message, but it could reduce some of the burden currently on the server requiring it keep track of each vehicle's location history.

A second approach to solving this problem might come from a larger change in the system architecture. If the knowledge of the highway model graph was shared between the server and the vehicle, then map matching could take place on the vehicle. Since the vehicle already has higher fidelity GPS and IMU data, the vehicle already has its path history, and path inference would not be necessary. However, the tradeoff with this approach is that the model graph must be communicated and kept up to date between the server and the vehicle, but this strategy alone would not necessarily solve the problem. In essence, the vehicle doesn't need to just track where it has been and where it currently is; it needs to predict when it is getting onto a freeway so that freeway speed data can be downloaded before the vehicle is actually traveling on the freeway. There could be an end-of-queue disturbance near the end of the entrance ramp or before the next exit, so waiting for a minute or two to place the vehicle on the freeway and get the correct data from the server is likely to be unacceptable for this type of system. Strategies need to be explored on how to best provide the vehicle with relevant freeway information before it gets onto the freeway.

6.1.3 "Slow Traffic Ahead" Alert Conclusions

This collaboration effort was the second iteration on providing drivers with a soft-safety "Slow traffic Ahead" alert. The initial implementation used in the Networked Traveler project utilized fixed alert trigger points about 1 mile upstream from known traffic speed locations, and the spacing of these trigger points varied from maybe ¼ of a mile to several miles depending on the density of the traffic sensors, which led to many missed and false alerts. The current project utilized the traffic flow models built in the Mobile Millennium project to interpolate between known speed locations, providing estimates of traffic speeds every 200 m (650 ft). Rather than having fixed alert trigger points, the vehicles utilized the freeway speed estimates directly, and this strategy worked well.

However, the tradeoff with providing such closely spaced speed estimates was in the amount of data that needed to be transmitted to the vehicle on each update. Each update contained 50 to 100 speed points, and most of the points were redundant, containing similar speeds. One strategy to reduce the amount of data being transmitted between the server and the vehicle would be to filter out the redundant speed points, creating a speed topology map. Rather than having a fixed distance between speed estimates, the speed estimates could be placed far apart when the freeway speed is relatively stable, but closer together when the freeway speed is rapidly changing.

One final topic for further research relates to the algorithm used to trigger a "Slow Traffic Ahead" alert. In the Networked Traveler project, prior research discussed in the literature review suggested that drivers should be alerted to speed differentials greater than 15 mph, but other algorithms or conditions could be placed on the alert. For example, historical speeds could be used to raise or lower the speed differential threshold, and the alert algorithm could do a better job of looking at the speeds ahead on the freeway to determine the recommended speed for the driver.

6.1.4 "Disabled Vehicle Ahead" Alert Conclusions

This collaboration effort was the first attempt that the researchers in this project know of to create EPVs and provide drivers with soft-safety alerts regarding disabled vehicles. The prototype system worked as expected. With an EPV update interval of 10 seconds, there was minimal delay, between 25 and 31 seconds, in transmitting and receiving the locations of disabled vehicles on the freeway network. Currently, in the prototype system, EPV updates to the server occurred at fixed intervals, and if an event occurred, such as the activation of the hazard warning lights, this event was not transmitted until the next scheduled update request. During the demonstration, the update rate was set to 10 seconds, so the longest that it took for an event to reach the sever was 9 seconds, but if the update interval were raised to one or more minutes, much longer delays would be expected. Part of the solution would be for certain vehicle events to trigger an immediate update from the EPV to the server, but that only reduces the delay on the transmitting of the event from the EPV to the server. The delay would also then be dependent on the update rate of the vehicles receiving the alert information. Reducing the event propagation delay below the vehicle update rate would then require the communication system to utilize an information push architecture, rather than the current pull architecture.

Perhaps the largest topic that requires future research centers around what kind of soft-safety alert to provide to drivers and when to provide them. Once the traffic server knows that there is a disabled vehicle of some sort, the question still remains about what to do with that information. During the demonstration, drivers were warned about all disabled vehicles, whether they were related to a traffic backup or not. Whether or not this is the best policy is still unknown. As an example, the policy in many Traffic Management Centers (TMCs) is to confirm the event with further evidence before giving drivers an alert (either on the website or through a variable message sign). Thus, if someone calls in a disabled vehicle, operators in the TMC might be required to visually verify the vehicle using cameras or some other means before they take action, and this may or may not be the best policy for in-vehicle alerts.

Currently, there have been very few functional implementations of soft-safety driving alert systems, and the Networked Traveler project, along with this current collaboration between California PATH and Renault are among those few. Based on this limited research, this topic stills shows much promise, but the question still remains as to what kinds of soft-safety alerts driver will accept, desire, and be willing to purchase.

REFERENCES

Bayen, A. (2010). *Mobile Century Final Report for TO 1021 and TO 1029: A Traffic Sensing Field Experiment Using GPS Mobile Phones* (Technical Report UCB-ITS-CWP-2010-04). Berkeley, CA: California Center for Innovative Transportation, Institute of Transportation Studies, University of California, Berkeley.

Bayen, A., Butler, J., and Patire, A. (2011). *Mobile Millennium Final Report* (Technical Report UCB-ITS-CWP-2011-06). Berkeley, CA: California Center for Innovative Transportation, Institute of Transportation Studies, University of California, Berkeley.

Nowakowski, C., Datta Gupta, S., Vizzini, D., Sengupta, R., Mannasseh, C., Spring, J., VanderWerf, J., and Sharafsaleh, A. (2011). *SafeTrip 21 Initiative: Networked Traveler Foresighted Driving Field Experiment Final Report* (Technical Report UCB-ITS-PRR-2011-05). Berkeley, CA: California PATH, Institute of Transportation Studies, University of California, Berkeley.

Nowakowski, C., Vizzini, D., Datta Gupta, S., and Sengupta, R. (2012). Evaluation of a Real-Time, Freeway End-of-Queue Alerting System to Promote Driver Situational Awareness. *Proceedings of the 91st Annual TRB Meeting*. Washington, D.C.: Transportation Research Board, National Research Council.