Data Management in the CarTel Mobile Sensor Computing System

V. Bychkovsky, K. Chen, M. Goraczko, H. Hu, B. Hull, A. Miu,
E. Shih, Y. Zhang, H. Balakrishnan, and S. Madden
MIT Computer Science and Artificial Intelligence Laboratory
The Stata Center, 32 Vassar St., Cambridge, MA 02139
http://cartel.csail.mit.edu/

1. INTRODUCTION

Worldwide, there are over 600 million automobiles on the road. Each automobile is a potentially rich source of sensor data, with the current generation of cars having over 100 sensors. Unlike many other mobile platforms, an automobile is a resource-rich environment that can support relatively robust computation and communication systems. More importantly, because automobiles interface with a vast amount of the physical world and are well-integrated into our daily lives, they are uniquely positioned to enable a broad range of sensing applications.

What can we do with 600 million mobile computing units (cars), each with tens of sensors, and on which we can place large amounts of computation? Here are some classes of applications that would arise if we expanded the reach of today's Internet-based computing substrate to include automobiles. In all these applications, cars are information sources.

1. *Traffic monitoring and route planning.* Suppose we tracked the location of every car (suitably anonymized) once per second using GPS. We can use this information to develop statistical models of traffic delays at various times of day on different road segments. Suppose you want to leave your home for the airport to catch a flight at 8:00 am. Which of the four different routes to the airport should you take?

2. *Preventive maintenance and diagnostics of cars.* By tapping into the on-board sensors using the standard CAN (controller area network) interface, and by attaching a variety of external sensors, we can monitor and report internal performance characteristics such as emissions, gas mileage, tire pressure, suspension health, etc. These reports can be combined with historical data, highlighting long-term changes in a car's internals, and correlating a given car's information with other cars of the same vintage to detect anomalies in a car's performance.

3. *Civil infrastructure monitoring.* When equipped with additional sensors to sample vibration and other conditions, cars can act as excellent "probes" to sense road conditions. Assessing and reporting road surface conditions such as potholes, oil spills, flooding, and ice can help cities and towns identify roads that need repair at relatively low cost, and help drivers to learn about hazardous driving conditions.

To enable these types of applications and others, we propose a reusable data management system, called *CarTel*, for querying and collecting data from intermittently connected devices. Our platform provides a dynamic query system that allows for both continuous and snapshot geo-spatial queries over car position, speed, and sensory data as well as both a low-cost and high-bandwidth substrate for communicating with a large network of mobile devices. What follows is a more detailed treatment of the key components of the CarTel mobile sensor computing system.

2. SYSTEM OVERVIEW

Figure 1 illustrates the basic architecture of the CarTel system. Each car is equipped with a *CarTel node*, which is an embedded PC outfitted with a variety of sensors and software for data collection. As cars drive around, they collect data, such as the car's current GPS location, WiFi availability, and engine performance anomalies. Sensor data is uploaded and control messages are downloaded using intermittent wireless connections (e.g., 802.11 hotspots, Bluetooth cellphones, or other CarTel-enabled cars). This information eventually reaches the Internet, where it is delivered to our data management interface, the AutoPortal, for visualization, analysis, and browsing.

CarTel relies on three key underlying technologies:

1. *AutoPortal:* Server software that provides data management, visualization, and web-based querying. This software requests data from remote nodes, aggregates reports arriving from those nodes into a coherent picture of current conditions, and visualizes that data.

2. *CafNet:* A networking infrastructure for carry-and-forward networks that leverages variable and intermittent network connectivity. CafNet is designed to work

SIGMOD 2006, June 27–29, 2006, Chicago, Illinois, USA.

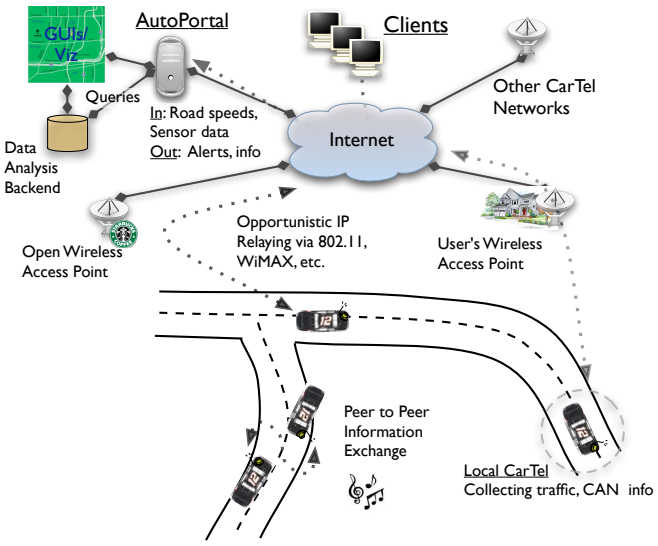

Figure 1: CarTel Architecture.

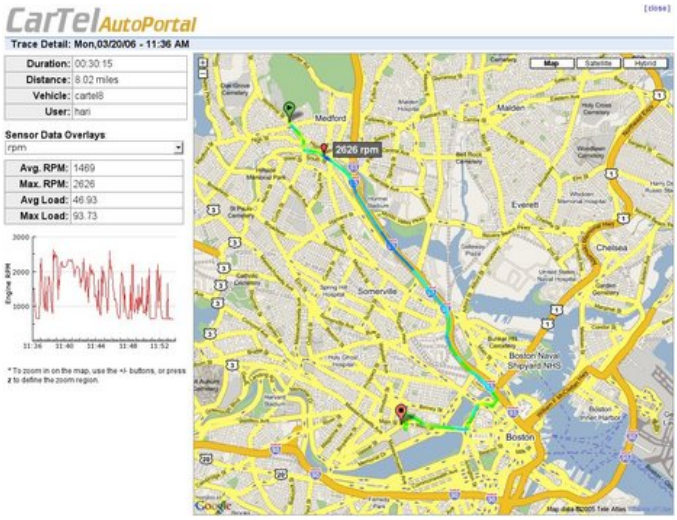

Figure 2: Screenshot of the AutoPortal prototype. Here, the user is visualizing a recently driven route.

with a heterogenous set of network technologies and manages the routing of data across many unreliable, high-latency links. CafNet treats the mobility of its network medium (e.g., USB keys, PDAs, cell phones) as an asset that helps it extend the reach of traditional networks.

3. *CarTelDB:* A device-level data management infrastructure that collects, pre-processes, and prioritizes information on remote nodes running CarTel software. Each device is able to automatically adjust its data-collection schema depending on the sensors present in the car. Data is aggregated and queries are processed using a simple stream-processing engine.

2.1 AutoPortal

The AutoPortal is the primary interface by which users interact with the CarTel network. It provides several classes of functionality, including data sharing, query processing, data visualization, and network management.

Figure 2 shows a prototype of an AutoPortal application that logs a user's travels. For each trip, the portal shows various statistics about the journey, as well as a graphical illustration of the trip, with the speed at each point along the route illustrated by different colors. Trips are grouped by start and/or end destination, or can be visually queried by highlighting regions of the map. Users have found even this simple application to be useful because it allows them to see the points of congestion on their routes between home and work. The AutoPortal provides an easy way for users to compare routes and determine which ones minimize their expected travel time.

The AutoPortal will include a query interface that allows users to visually filter and process sensor data without extensive knowledge of SQL. Using our system, users can graphically define *interest regions* on a map showing the collection area. Users then define connections between interest regions using boolean operators. Additionally, domain-specific operators can be dragged to a region to apply filtering and processing to any matching sensor data.

For example, imagine that you want to compare all of your routes between home and work. You would first highlight two regions, one around your home and one around your work, and set the return type for each to be a GPS trace. You would then drag a boolean 'and' operator onto the map to link the two highlighted regions, ensuring that only those traces that go from your home to work are returned. Finally, to make the result set more manageable, you can apply an aggregate operator over the result set that groups traces based on similarity. Traces that follow the same route would be grouped and only statistics for unique routes would be reported.

2.2 CafNet

The design of our network infrastructure, CafNet, is motivated by two technology trends: (1) the increasing prevalence of pockets of inexpensive high-bandwidth wireless access using WiFi, and (2) the rapidly growing amounts of inexpensive storage on small devices. Sensing applications can generate hundreds of megabytes of data every day. Cars do not usually have continuous access to high-bandwidth connectivity. However, during the course of a day, they will have access to islands of high wireless bandwidth. Hence, our approach to networking integrates routing and storage by relying on the movement of cars and users to move data from source to destination.

The fundamental data-carrying unit in CarTel is a *data mule*. Any storage device can act as data mule, including USB keys and cell phones. As cars traverse islands of connectivity (e.g., WiFi), they opportunistically relay data via the Internet. Mules also opportunistically forward data to each other when they are in the same wireless neighborhood.

Future forwarding opportunities are hard to estimate reliably in a disconnected mobile environment, such as CarTel. Consequently, applications must gracefully adapt to varying network bandwidth and delay. Our CafNet implementation allows applications to express the importance of data through priorities. When bandwidth is insufficient, priorities are used to drop data according to its importance.

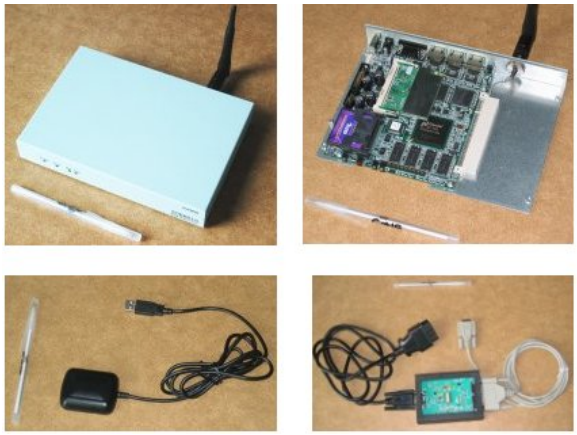

Figure 3: Prototype CarTel hardware placed in cars. Clockwise: the WiFi card in the box, the OBD interface to connect to the car's CAN, the GPS receiver, and the car gateway box.

We designed CafNet to decouple the communication challenges of managing an intermittently connected environment from those of querying and collecting data. Our communication stack consists of three layers: transport, network, and mule adaptation layer. The purpose of the transport layer is to provide applications with an interface for sending and receiving data. The network layer maintains topology information and chooses the appropriate media for the next hop. The mule adaptation layer presents heterogeneous types of media to the network layer in a uniform fashion.

2.3 CarTelDB

Figure 3 shows a prototype of the CarTel data management and collection hardware deployed in each car. Inside each CarTel node is an embedded PC running Linux attached to a variety of sensors. This hardware runs a simple data management system that makes it easy to build applications that process and collect different types of information from the vehicle. This system provides the following features:

1. A way to add support for collecting data from new types of sensor hardware. Sensors are self-describing and expose a set of *attributes* that can be referred to in queries. For example, if an engineer wants to collect data about engine RPMs in the car, he can write a small amount of code that retrieves RPMs at a particular rate from the on-board vehicle bus and passes that data onto the query system as the `rpm` attribute. The adapters and attribute metadata for these sensors (*data sources*) can be propagated to the devices, or can be provided by the sensors themselves. By advertising their attributes and source properties, sensors allow the CarTel node to recognize and collect data immediately upon attachment, without manual configuration or any other user intervention.

2. Continuous queries that read in low-level attributes and process them. For example, if an engineer wants to compare engine RPMs to speed, he can run a query that samples both attributes and compares their relative changes every few seconds. These queries are similar to regular SQL queries, but with an additional parameter to specify the rate at which the query is performed over the appropriate tables and output to CafNet. The query manager on the device is responsible for locally adding and removing these queries, and executing them at the correct intervals. The query results are stored in a local database.

3. An interface to CafNet that delivers query results over intermittently available network resources. Data source updates, query updates, and query results are all transmitted via CafNet. In addition, we make several small extensions to SQL that allow users to specify prioritization functions. These functions dictate the transmission order of query results and are particularly useful in variable bandwidth environments where it is unlikely that all data will be delivered (and a simple FIFO ordering of tuples will results in extremely spotty coverage).

3. DEMONSTRATION HIGHLIGHTS

In this demo we show a prototype of the CarTel sensor computing system, including the following highlights:

1. *Visual query system:* We demonstrate our graphical query system using historical data collected by our CarTel deployment in Boston and Seattle. This demo shows how our system can be used to compare travel times for a user's common routes. For example, one of our users commonly travels from Winchester to Boston. Each day he must select one of four routes to take into work. Using our collected data, we show how such a user can sift through the thousands of traces he has collected to meaningfully compare these four routes.

2. *Opportunistic data transfers:* We rent three cars and install a CarTel node in each one. During the demo, these cars travel around downtown Chicago, collecting sensor data and uploading the data to our servers over wireless connections. At the conference site, we use the AutoPortal to show the access points being used for uploads and traces of where the cars have been traveling in the last few minutes.

3. *Real-time sensing and data muling:* We connect a digital camera to several stationary CarTel nodes. The images that these cameras collect are relayed through a Bluetooth equipped cellphone to a laptop, demonstrating data muling. The cellphone displays networking statistics such as current throughput and amount of queued data. The laptop displays video frames received from the laptop.