CarTel: A Distributed Mobile Sensor Computing System

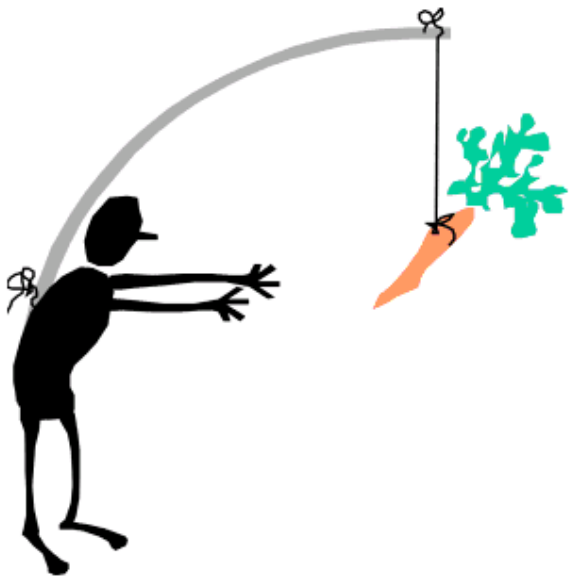
by Bret Hull, Vladimir Bychkovsky, Yang Zhang, Kevin Chen, Michel Goraczko, Allen Miu, Eugene Shih, Hari Balakrishnan and Samuel Madden

Presenter: Ionut Trestian

CarTel - Introduction

- Mobile Sensor Computing System
- Uses CARTEL nodes –mobile embedded computer coupled to sensors.
- Functioning:
  - Nodes gather sensor readings
  - Local data processing (prioritization, summarization …)
  - Deliver to a central portal
  - At portal, data can be analyzed and visualized

CarTel - Motivation



- Emerging of mobile sensor networks
- "Technology push" thousands of sensors available embedded in computers and phones.
- "Application pull" – the need for cheap monitoring applications.
- Mobile solutions have advantages over static ones
  - Bigger geographical areas covered
  - No need for static infrastructure put in place

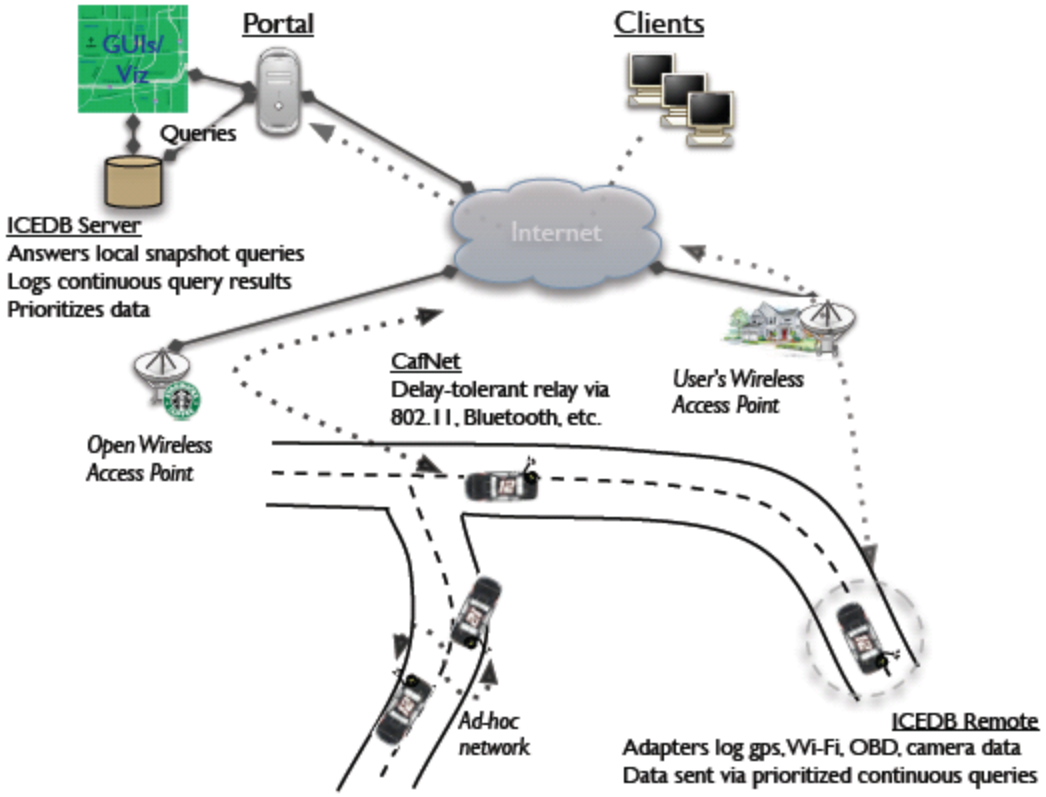
CarTel - Applications

- Environmental monitoring (chemical and pollution)
- Civil infrastructure monitoring (vibrations and other sensors to monitor state of roads, bridges)
- Automotive diagnostics (obtaining information from a vehicle's on-board sensors)
- Geo-imaging (capturing images from locations)
- Data muling (using cars and people as delivery networks)

CarTel – Overview & Contributions

- Reusable software platform that can be used to build many mobile sensing applications.
- High-level goals:
  - Provide a simple programming interface
    - Apps can be written as web apps – distribution and mobility are hidden)
  - Handle large amounts of heterogeneous sensor data
    - Any kind of sensors could be plugged,
    - Large amounts of data can be buffered and processed on nodes
  - Handle intermittent connectivity
    - Opportunistic wireless, eg WiFi passing by and therefore intermittent
    - Other storage devices (USB keys or flash drives) used as data mules

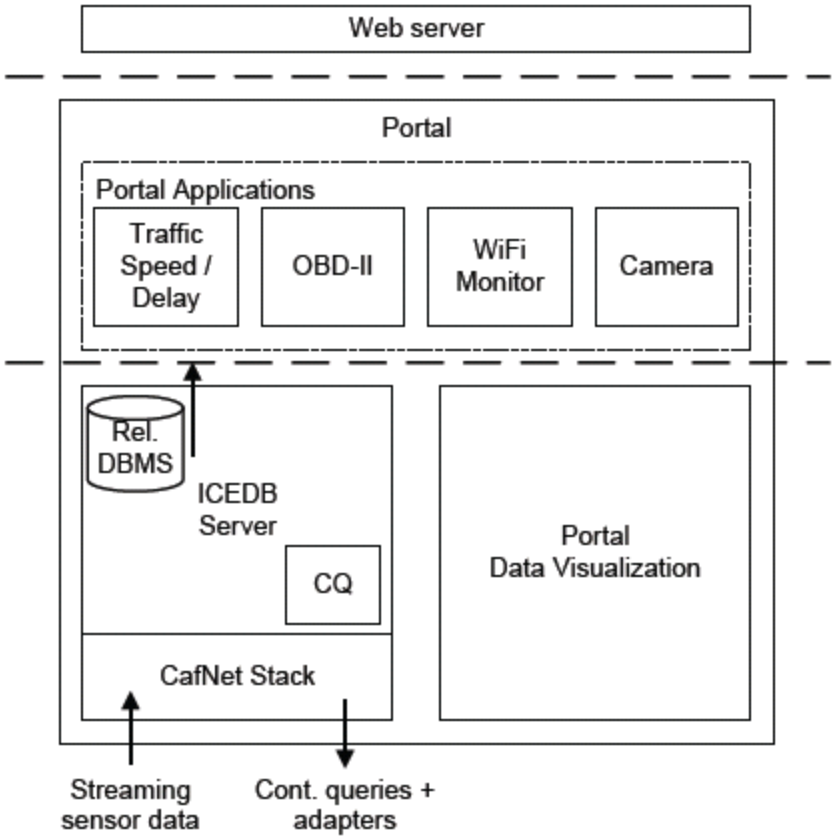
CarTel - Components



- ICEDB (intermittently connected database)
  - A delay-tolerant continuous query processor.
- CafNet (carry-and-forward network)
  - a delay-tolerant network stack
- Portal
  - Sink of the data
  - Data visualization

CarTel – Portal programming model

- Applications running on the portal issue continuous queries using an API exported by ICEDB
- Queries specify:
  - Sensor data to be acquired (rate, sampling, filtering ..)
  - Query results stream across an intermittently connected network and populate a relational database at the portal
  - Applications issue snapshot queries that run on whatever data is currently available.



CarTel - Contributions

- CarTel builds on previous work on mobile systems, sensor data management and DTNs.
- Therefore the main contribution is the synthesis of ideas.
- Other contributions:
  - Expanding the notion of continuous queries to handle intermittent connectivity.
  - Enabling modular upgrades to integrate new sensors and data types using adapters.
  - The CafNet "carry-and-forward" DTN stack that delivers data in intermittently connected environments (using callbacks across all layers – allows the sender to dynamically prioritize data).
  - Design of portal and visualization interface

ICEDB

- ICEDB distributes query execution and result delivery between the ICEDB server running on the portal and the remote nodes.
- Server
  - Maintains a list of continuous queries submitted by the applications and pushes them to the remote nodes using CafNet
- Nodes
  - Run ICEDB remote to process the sensor data and return the query results using CafNet

ICEDB – Data types

- Supports heterogeneous data types and makes the addition and removal of sensors easy
- Uses a meta-data package called an adapter for handling new sensor types (basically a script)
  - Automatically create local tables to store sensor readings
  - Acquire tuples from the sensor
  - Parse sensor readings to store them in the database and process them by subsequent queries.
- Adapter attributes:
  - ID, name, Type (push or pull data), Rate (for pull), Forwarding flag (raw data delivered to the portal), Schema (name-type pairs, currently PostgreSQL data types), Priority.
- Currently, adapters for node diagnostics, GPS receiver, OBD-II interface, WiFi interface and digital camera.

ICEDB – Continuous Query Model

- usage of RATE clause
- *SELECT carid, traceid, time, loc FROM gpsWHERE gps.time BETWEEN now()-1 mins and now()* ***RATE*** *5 mins*
- clock synchronization via NTP
- Prioritized data delivery
- Prioritization Schemes
  - *Local Prioritization*
    - *Cannot receive feedback from portal*
  - *Global Prioritization*

ICEDB – Prioritization Schemes

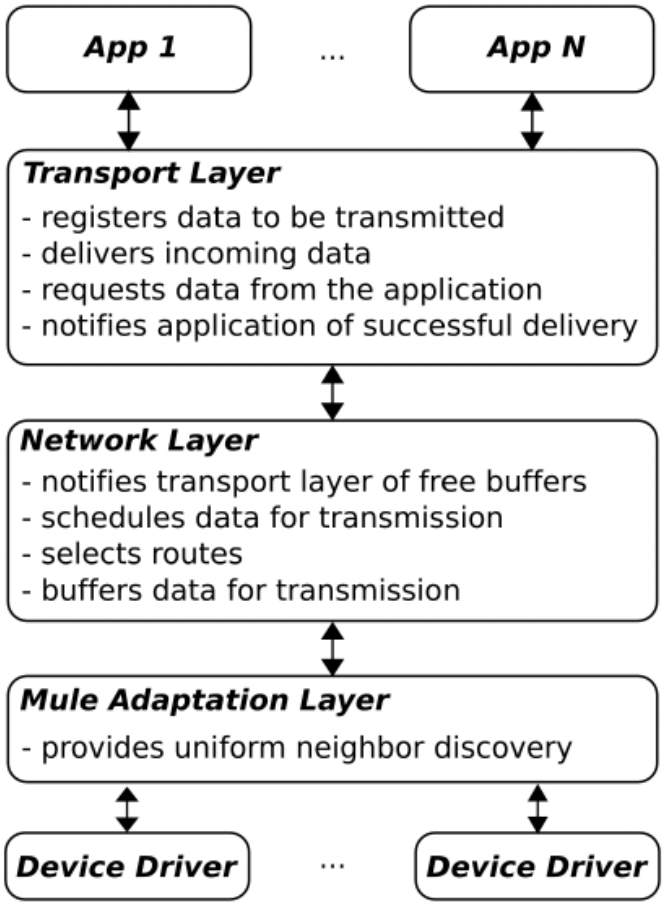
- Local Prioritization
  - Uses two language extensions for specifying the transmission order of results: PRIORITY and DELIVERY ORDER.
  - PRIORITY assigns a numeric priority to the query's result buffer
  - DELIVERY ORDER is the further level of granularity which goes into the query and specifies the order of the attributes.
- Global Prioritization
  - Expressed using the SUMMARIZE AS clause which specifies a query that will compute a summary.
  - Mainly based on aggregates

CafNet - Overview

- Network stack for delay tolerant communication
- Uses a message-oriented data transmission and reception API not a stream-oriented connection such as TCP.
- All nodes are named using globally unique flat identifiers that don't embed any topological or organizational semantics (hashes).
- Network stack does no buffering
  - Applications buffer
- CafNet informs the application when connection is available
- Application can decide what data to send "at the last moment" instead of committing earlier to the data in advance by sending to the net layer.

CafNet – Basic Stack

- CTL can be implemented as a library or separate process which communicates with the application using RPC
- CTL provides optional delivery confirmation service by setting a flag (NONE or END2END - retransmissions).
- CNL provides a send function.
- Peer discovery happens at the MAL because these mechanisms are media-specific.



CafNet – Optimizations and Enhancements

- Design is "pure", has no network buffering but performs poorly when average duration of connectivity is not significantly larger than application package time.
- Solution: introduce a small amount of buffering in the stack (CNL).
- Setting CNL buffer large hinders application ability to prioritize data.
- Solution: Let application set the size of the CNL buffer.

The Portal

- Central repository
- Three main components
  - Portal framework
  - ICEDB server to retrieve sensor data
  - Data visualization library to display geo-coded attributes.
- Provides privacy to Cartel Users
- Applications issue continuous queries and view the results. Once issued the queries are pushed to the nodes. Intermediate results are displayed.
- Uses *Trace: data collected in a single trip*
- Provides two class of functions
  - interface to search traces using spatial queries
  - interface to overlay geographic attributes on a map, for any trace

The Portal

Trace Explorer

Query Options:

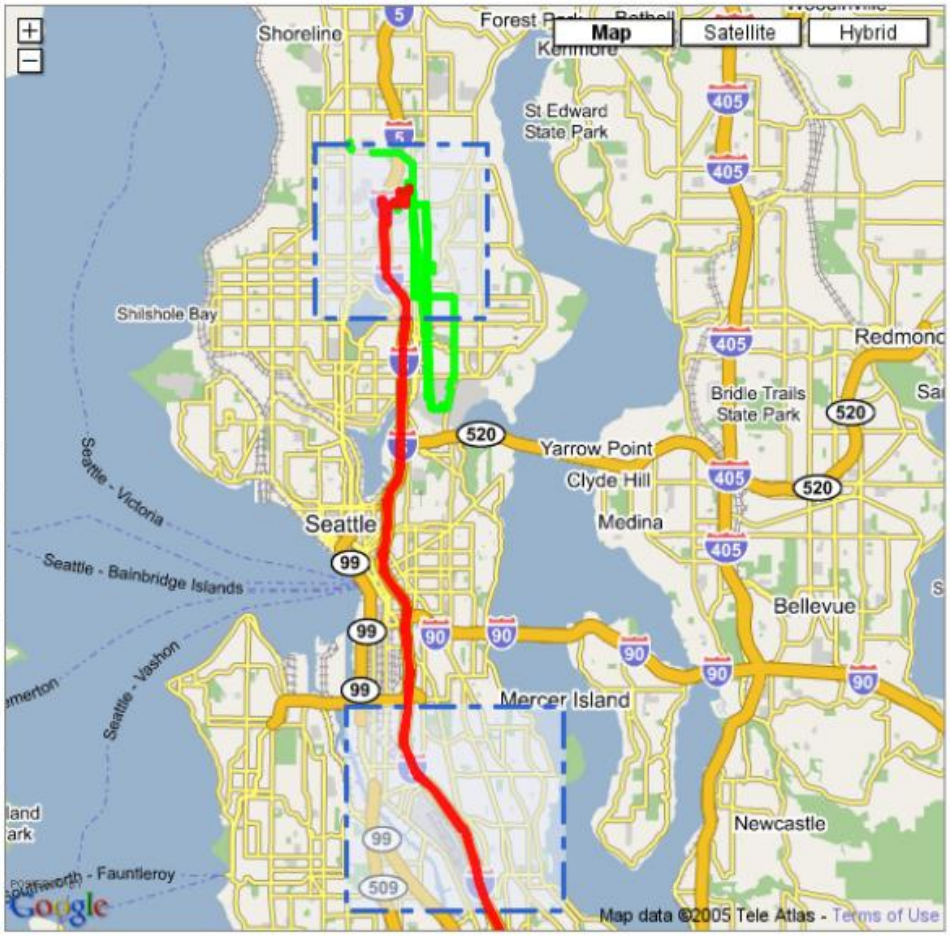
Date: All Dates

Tags: Anywhere

Operator: Intersects Refine Reset

	Date	Time	Dur.	Dist	Mark
[details]	Tue, Apr 4	12:11 AM	00:53:08	44.79 miles	☑
[details]	Mon, Apr 3	9:25 PM	00:32:32	9.25 miles	☐
[details]	Mon, Apr 3	12:00 PM	00:00:35	0.15 miles	☐
[details]	Mon, Apr 3	11:38 AM	00:02:33	0.39 miles	☐
[details]	Fri, Mar 31	8:55 PM	00:16:35	4.60 miles	☐
[details]	Fri, Mar 31	6:45 PM	00:18:39	5.03 miles	☐
[details]	Fri, Mar 31	1:52 PM	00:11:42	3.97 miles	☐
[details]	Thu, Mar 30	12:16 PM	00:27:45	7.58 miles	☐
[details]	Wed, Mar 29	12:07 PM	00:34:05	8.37 miles	☐
[details]	Tue, Mar 28	5:16 PM	01:01:38	0.95 miles	☐
[details]	Mon, Mar 27	12:04 PM	00:29:00	7.40 miles	☐
[details]	Sun, Mar 26	8:15 PM	00:00:41	0.23 miles	☐
[details]	Sun, Mar 26	8:08 PM	00:00:23	0.09 miles	☐
[details]	Sun, Mar 26	7:06 PM	00:03:27	1.37 miles	☐
[details]	Sun, Mar 26	6:58 PM	00:01:07	0.18 miles	☐

Older >



The Portal

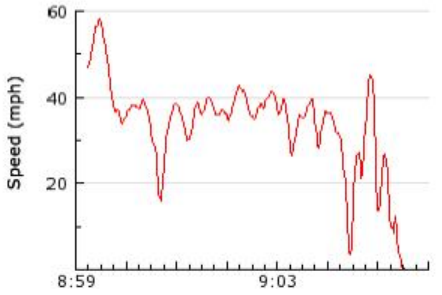
Trace Detail: Fri,03/10/06 - 8:59 PM

Duration:	00:06:17
Distance:	3.52 miles
Vehicle:	[redacted]

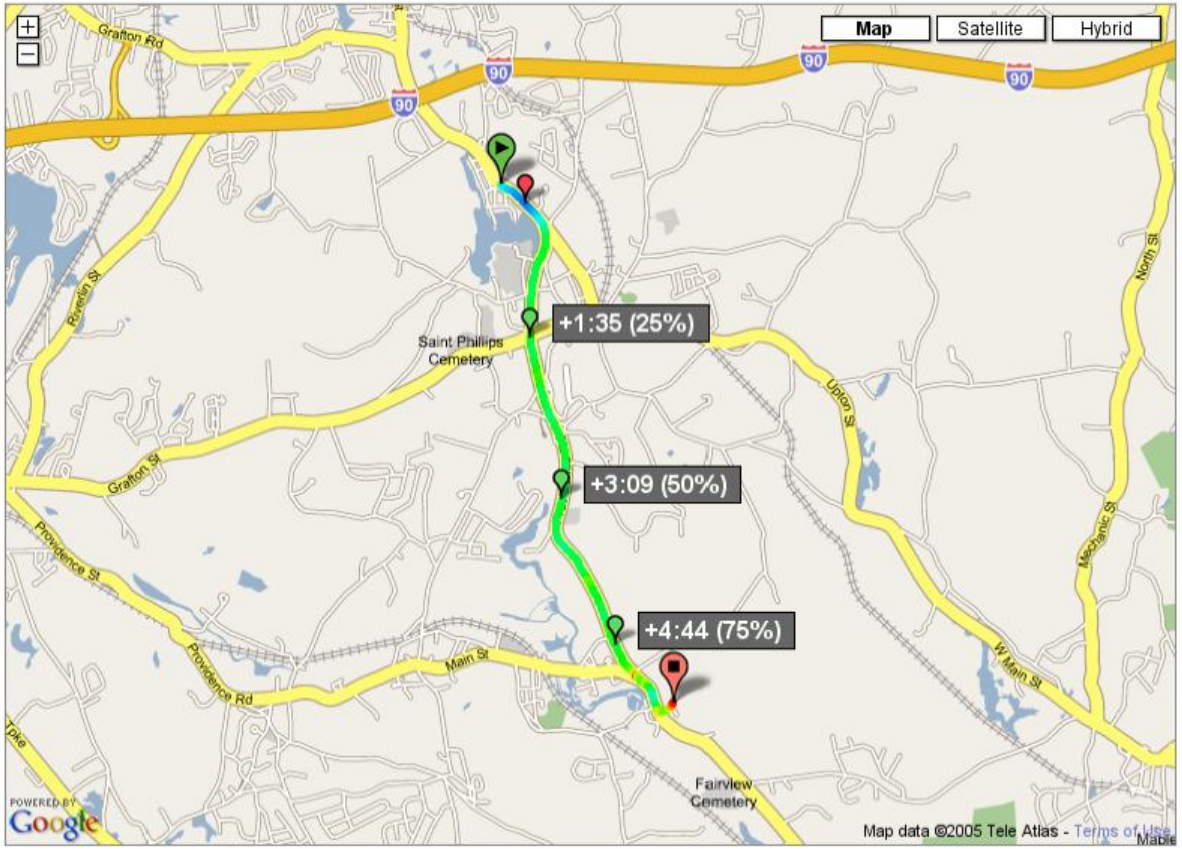
Sensor Data Overlays:

speed

Avg. Speed:	33.62 mph
Max. Speed:	58.06 mph



* To zoom in on the map, use the +/- buttons, or press **z** to define the zoom region and **x** to zoom out.



Case studies – Road Traffic Monitoring

- Road traffic analysis: GPS adapter and continuous queries to keep track of routes and delays
  - Commute time analysis

Route	Avg. Dist.	Avg. Time	Std-dev
Freeway	9.94 miles	19:52	02:14
City Streets	9.83 miles	29:34	02:19
Frontage Road	9.27 miles	31:51	03:54

  - Traffic hot spot heuristics
  - Image acquisition

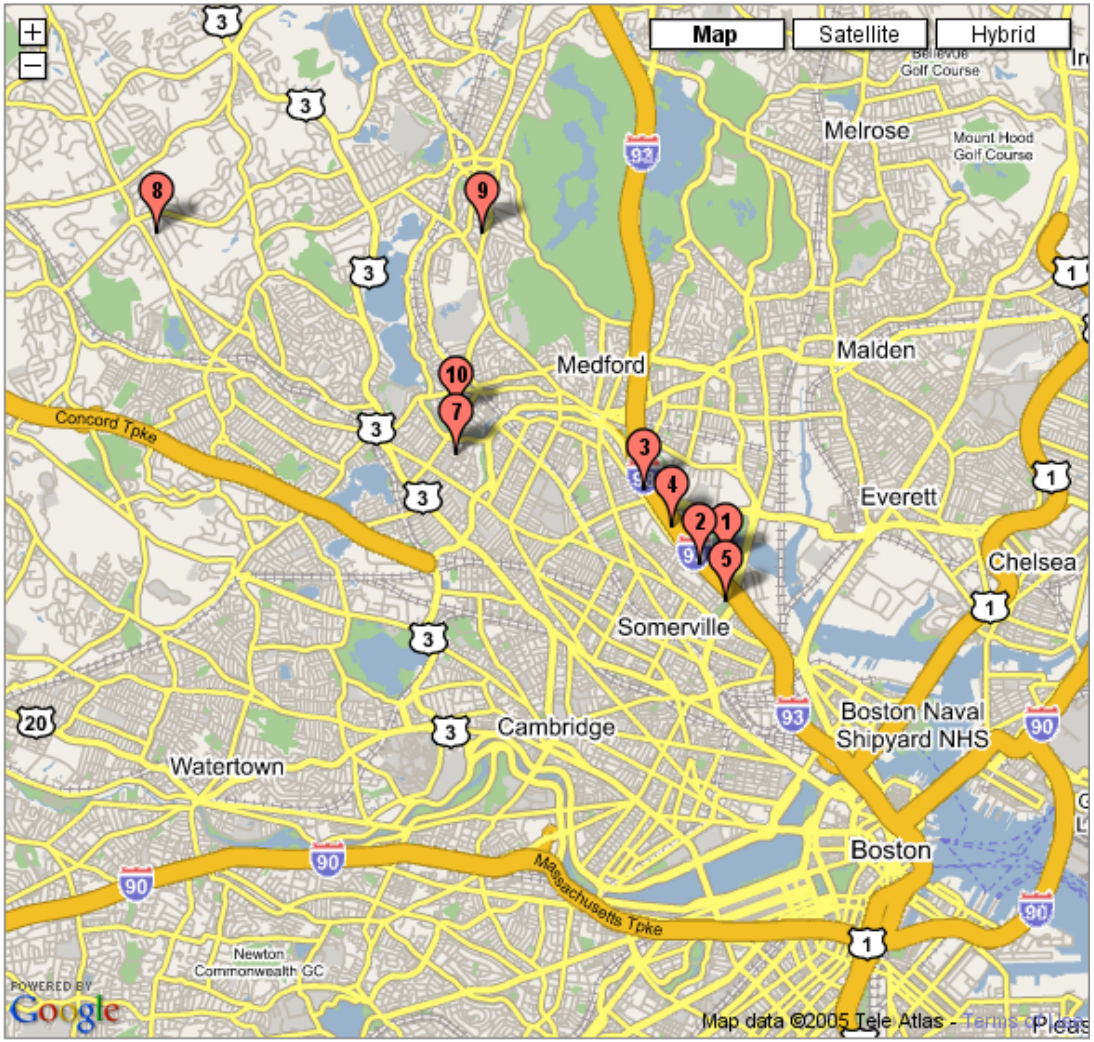
Traffic hot spots – Boston area

Traffic Hotspots

Start: 7 am End: 10 am Grid: .005 Update

Rank	Avg MPH	Std-dev	Count
1	25.3 mph	26.7 mph	176
2	29.3 mph	24.6 mph	315
3	33.0 mph	22.8 mph	267
4	39.4 mph	22.5 mph	245
5	16.0 mph	18.0 mph	729
6	32.9 mph	17.5 mph	187
7	20.4 mph	17.1 mph	635
8	34.2 mph	16.2 mph	53
9	17.9 mph	15.9 mph	365
10	17.2 mph	15.3 mph	313

* To zoom in on the map, use the +/- buttons, or position the cursor over the map and press **z** to define the zoom region and **x** to zoom out.



Traffic hot spots – Seattle area

Traffic Hotspots

Start: 10 am End: 1 pm Grid: .005 Update

Rank	Avg MPH	Std-dev	Count
1	40.7 mph	28.1 mph	69
2	19.3 mph	22.1 mph	91
3	56.0 mph	21.9 mph	39
4	23.2 mph	18.5 mph	52
5	16.5 mph	17.7 mph	51
6	24.9 mph	17.3 mph	197
7	11.8 mph	15.7 mph	111
8	25.7 mph	15.6 mph	186
9	16.4 mph	15.2 mph	128
10	14.9 mph	14.9 mph	2770

* To zoom in on the map, use the +/- buttons, or position the cursor over the map and press **z** to define the zoom region and **x** to zoom out.

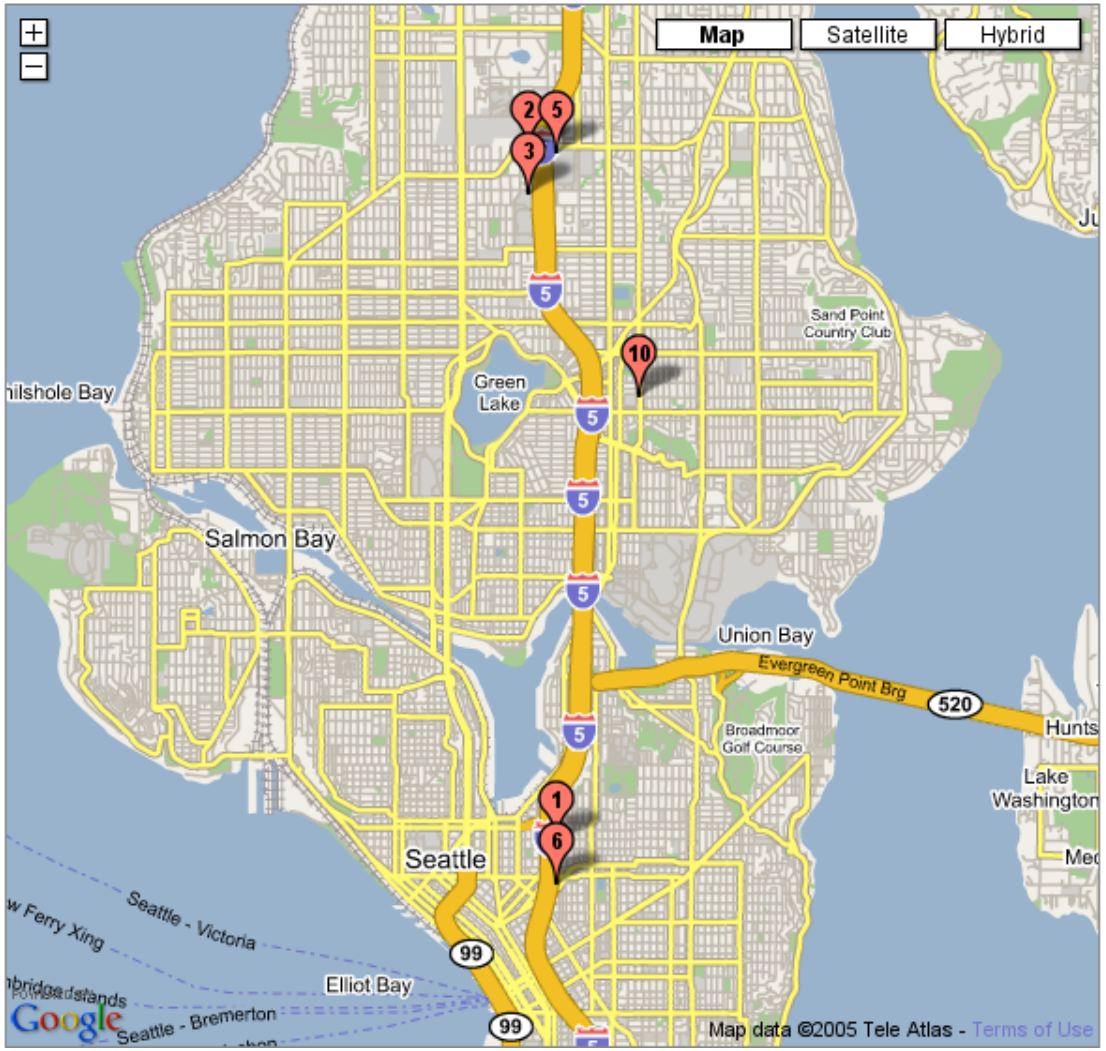


Image acquisition

Wide area WiFi Measurements

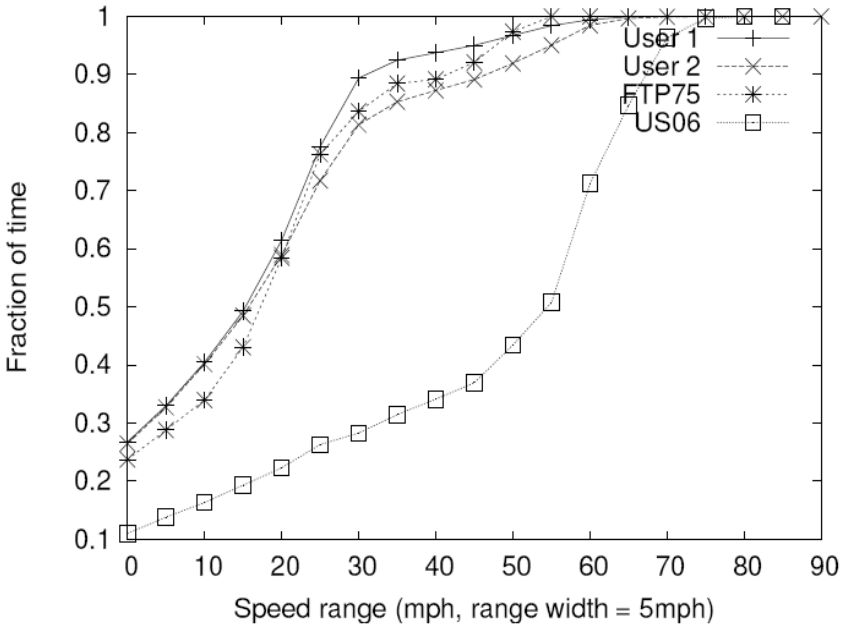
- 65% of on-line households have installed Wi-Fi access points Jupiter Research.
- WiFi sharing?
- Collect over 290 "drive hours" of data about Wi-Fi connectivity in urban environments.
- Data:
  - Wi-Fi scans – reports of nearby APs
  - Wi-Fi associations - attempts to establish link-layer connectivity with APs
  - IP address acquisitions – attempts to acquire an IP address using DHCP.
  - Ping and upload statistics.
- 32000 Aps, 5000 associations, 2000 gave IPs.

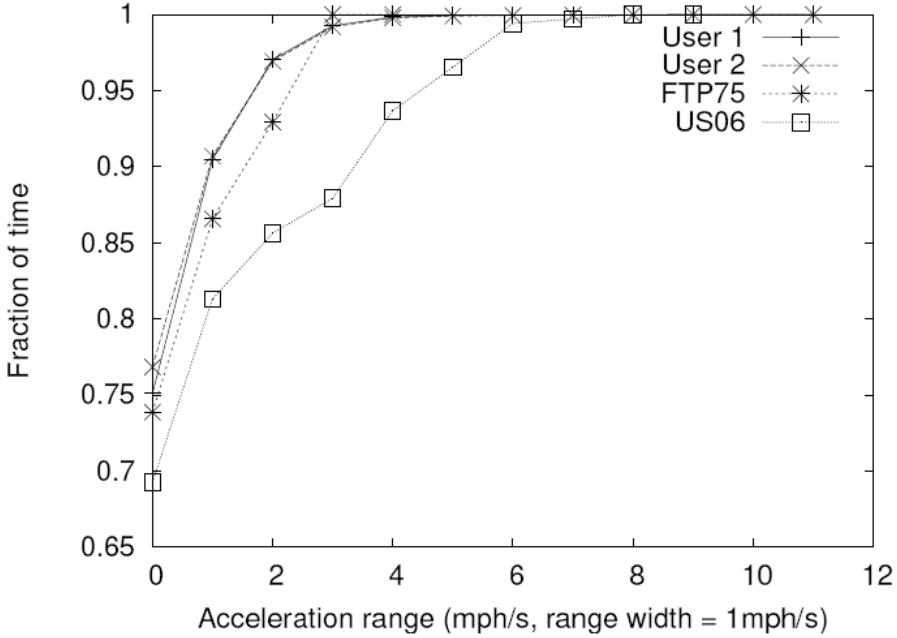
Mean association duration	25 seconds
Mean time between connections to Internet	260 seconds
Median upload throughput	30 KBytes/s

Automotive Diagnostics

- Federal Test Procedure (FTP75) rates cars for fuel economy and emission levels.
- Test criticized so EPA introduced US06 that has harder acceleration and moving speeds (not used for fuel economy purposes).
- OBD Data
  - Emissions, engine status, fuel consumption, troubleshooting codes, engine load, fuel consumption and pressure, engine RPMs, engine timing, air intake temperature, engine throttle position and oxygen sensor status.

Automotive Diagnostics





Conclusions & Future work

- Hundreds of millions of automobiles (to which embedded computers can be attached) and over a billion-phone equipped people. HUGE Potential
- Cartel provides software to collect, process, deliver and visualize data from sensors located on mobile devices to a portal.
- Future work:
  - Aggregate information across users.
  - Process and analyze more data obtained from OBD sensors.
  - Incorporate simple routing algorithms.
  - More queries and users.