Mobile Macroscopes: The CarTel Project

Sam Madden
MIT CSAIL
http://cartel.csail.mit.edu

With Hari Balakrishnan, Vladimir Bychkovsky, Bret Hull, Yang Zhang,
Kevin Chen, Waseem Daher, Michel Goraczko, Hongyi Hu, Allen Miu, Eugene Shih

Supported by a research grant from Quanta Computer and NSF

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

What is CarTel?

- Distributed software system that makes it easy to:
  - collect,
  - process,
  - deliver,
  - visualize & analyze

  data from mobile sensors (cars, phones, etc)

- Goals + Challenges:
  - Heterogeneous data
  - Lots of data (“media-rich” sensors)
  - Intermittent network connectivity + mobility
  - Programmability

Portal

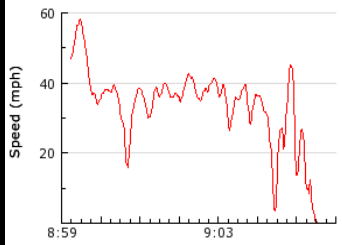
Trace Detail: Fri,03/10/06 - 8:59 PM

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

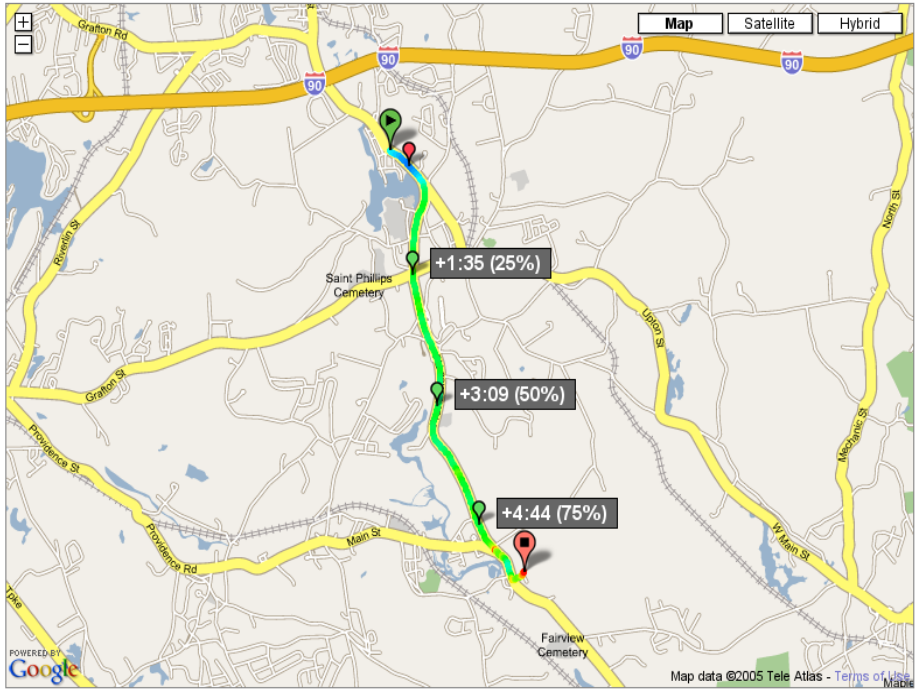
Sensor Data Overlays:

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.



Embedded Linux: Pentium-class, 128MB Ram, 1GB storage

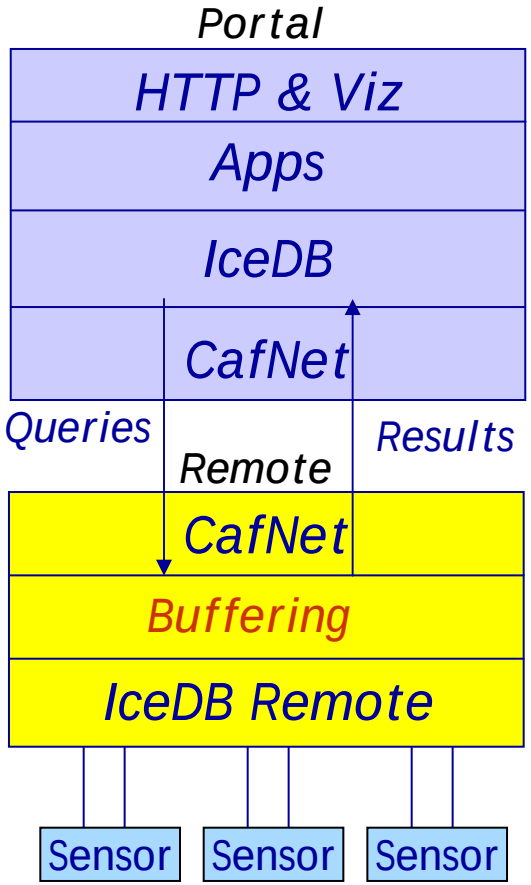
QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

CarTel Software Components

- *CarTel Portal*: Centralized, visual user interface
- *IceDB*: Intermittently connected DB
  - Centralized declarative queries
  - Executed in distributed fashion by mobile nodes
  - *Delay-tolerant continuous query processing*

```
SELECT oil_temp
FROM obd
BUFFER IN engine
PRIORITY 1
```

- *CafNet*: CarTel's network stack
  - H
    c



This talk: study properties of network, using CarTel itself to do this (MobiCom '06)

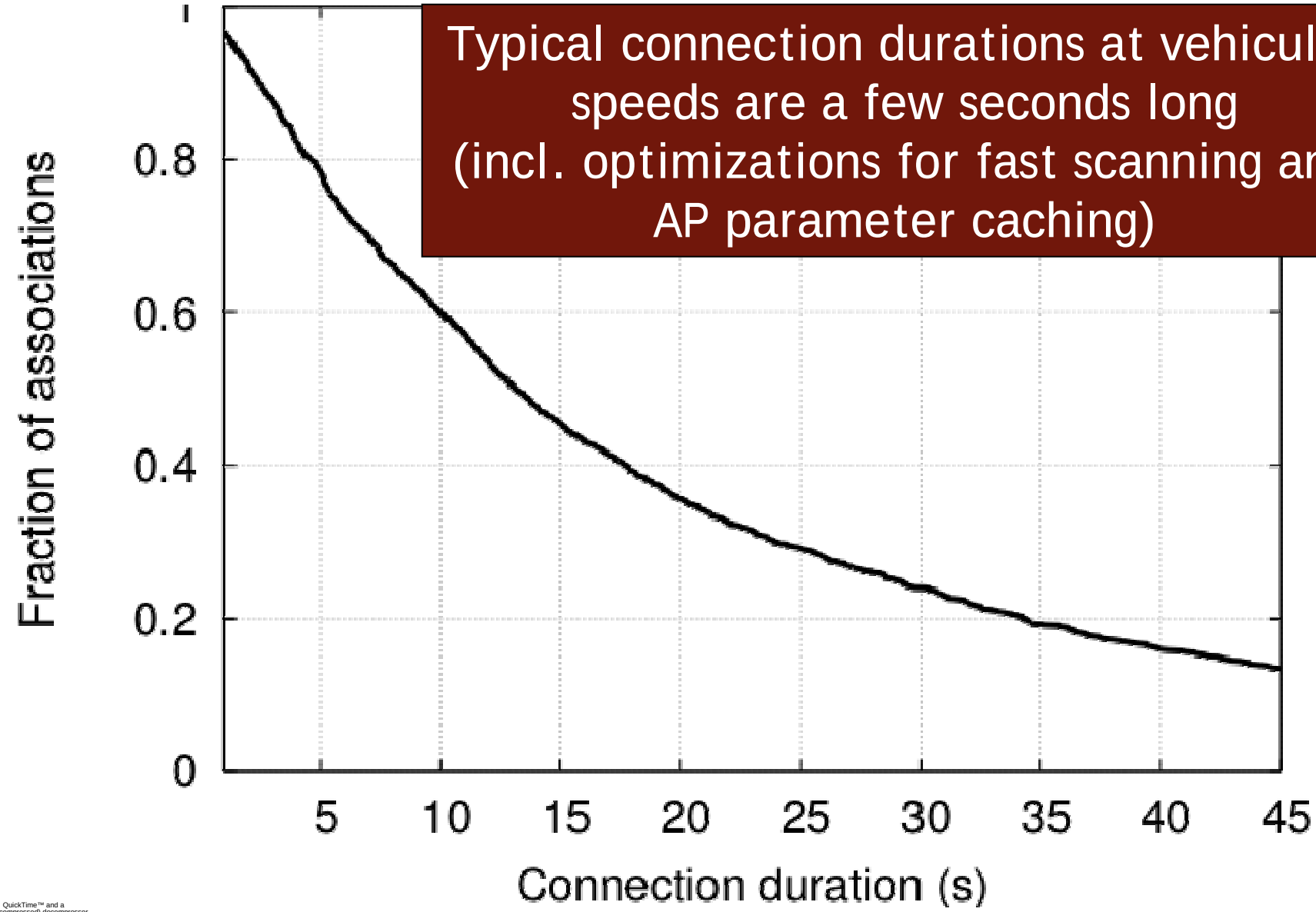
QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.


From wigle.net
IN
OUT
I-93
Mbta Commuter Rail
Boston and Maine Railroad
Little Mystic Ch
Boston Nat Hist Pk
Charles River
US 1
C G Integrated Support Co
3
Boston Harbor
Charles River
The Esplanade
Frog Pond
Charles River Basin
Public Garden
Fort Point Channel
Penn Central Railroad
Storrow Memorial Dr
Boston University
Muddy River Pond
Copley Square
I-90
Fenway Park
Home of the Red Sox
Mother Church of Christ Scienc
Fort Point Channel
Muddy River
Northeastern University
Orange Line Railroad
Penn Central Railroad
©WiGLE.net

Experimental Setup

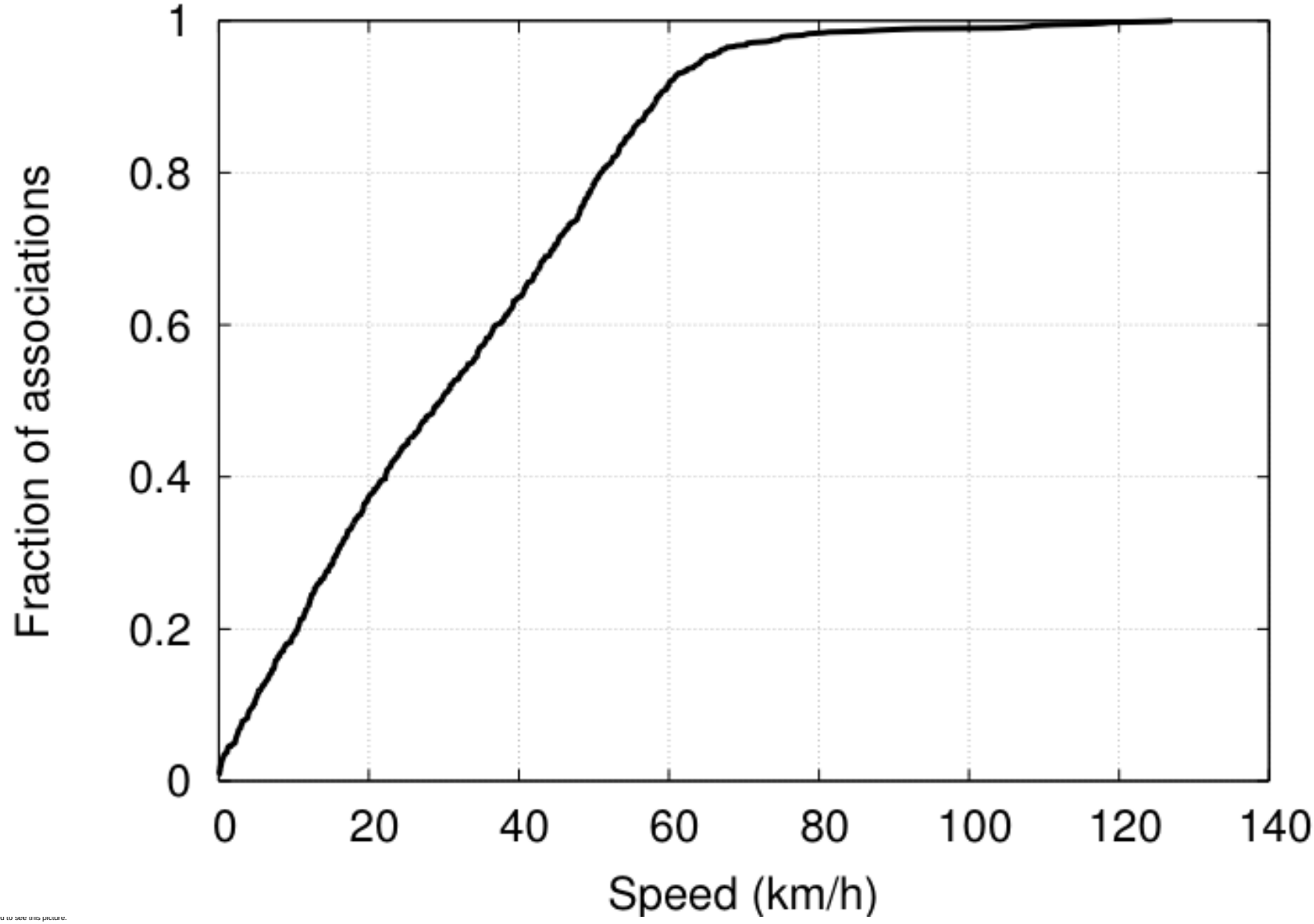
- 6 cars equipped with CarTel box and software
  - Driving normally in parts of the Boston area
  - ~32K access points (APs) mapped in all on a relatively small number of distinct routes
  - ~300 drive hours
- Fast scanning of WiFi access points, caching of AP parameters to speed up connection establishment
- Careful (small and unobtrusive) TCP data transfers to measure throughput, latency, loss rates
- Track performance statistics: connection durations, throughput distributions, etc.

Connection Duration Distribution



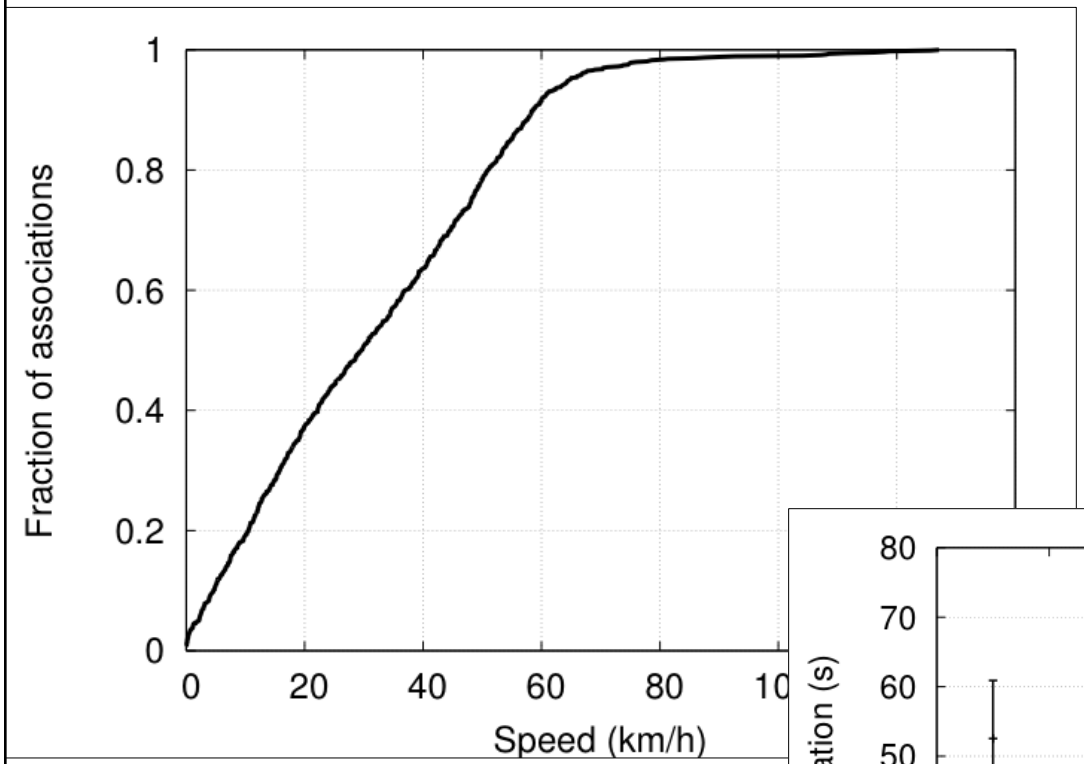
Typical connection durations at vehicular speeds are a few seconds long (incl. optimizations for fast scanning and AP parameter caching)

Impact of Car's Speed



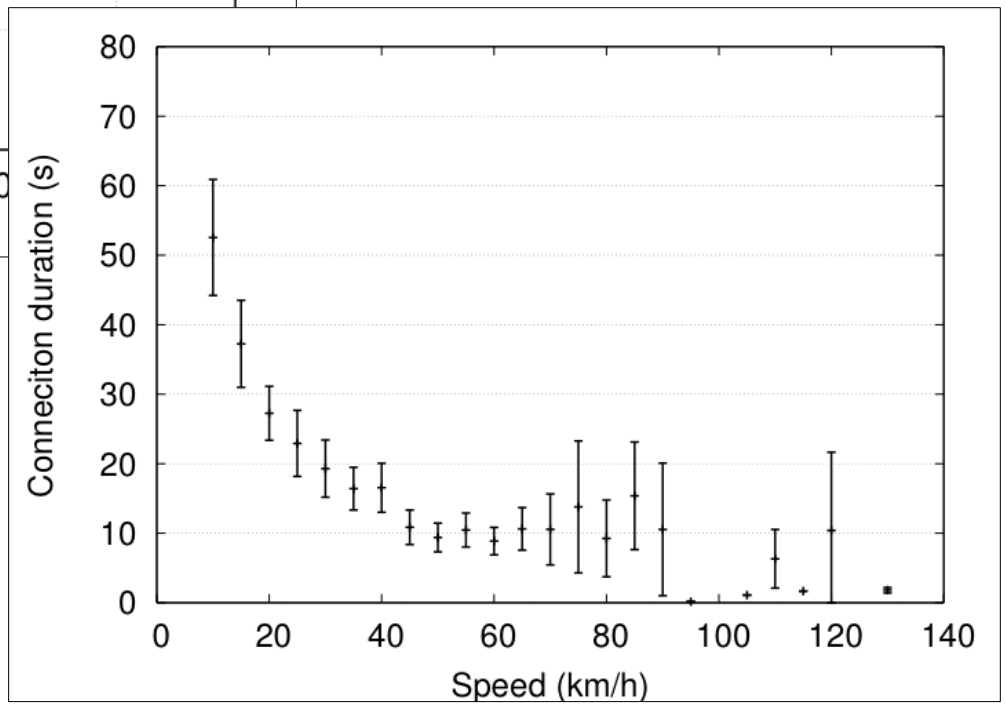
TIFF (Un
are needed to see this picture.

Impact of Car's Speed



Connections establish at range of speeds

Very little data at higher speeds (system isn't optimized for subsecond connections yet)

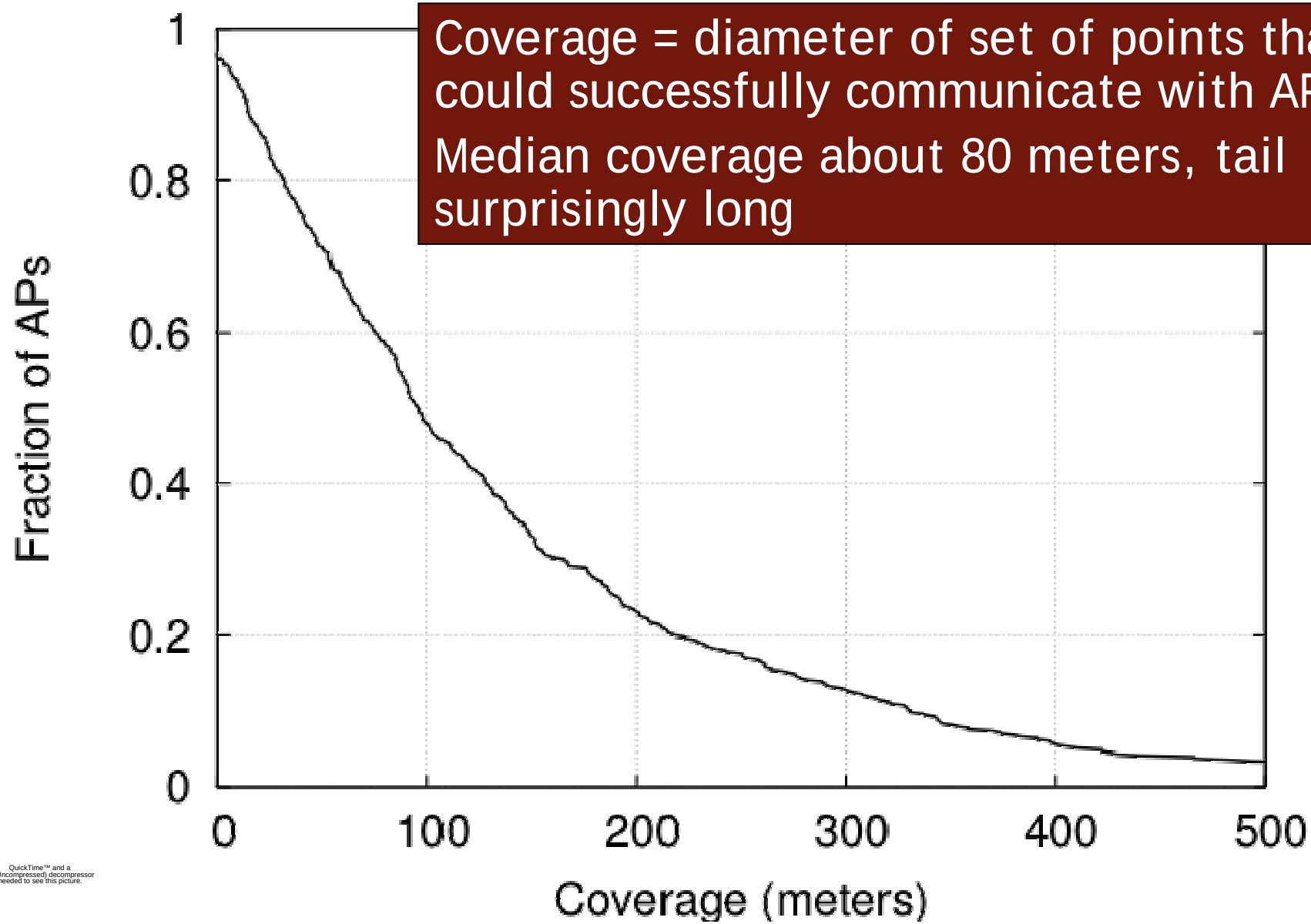


QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Coverage Region

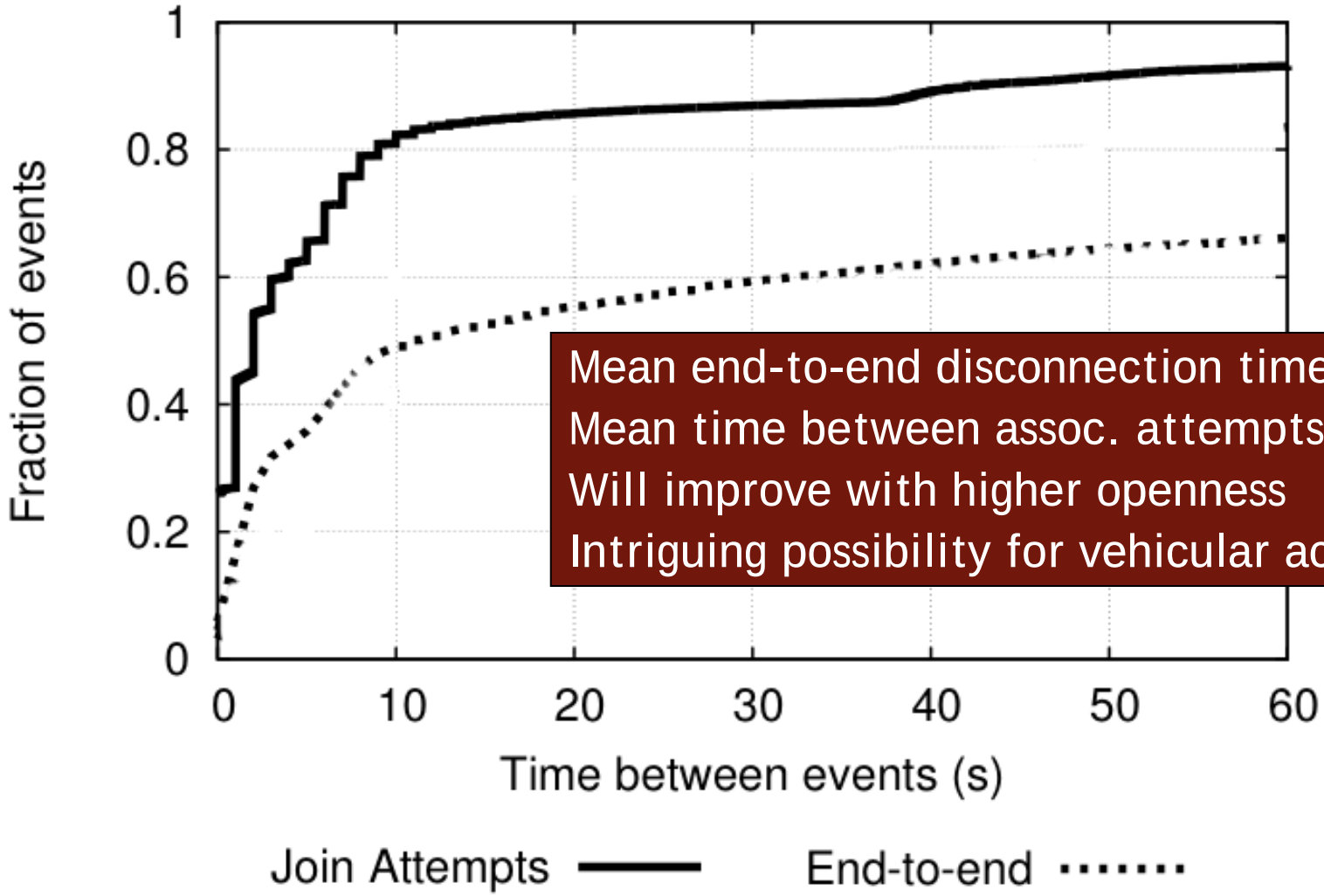
Coverage = diameter of set of points that could successfully communicate with AP

Median coverage about 80 meters, tail surprisingly long



QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

Disconnection Duration Distribution



Mean end-to-end disconnection time 260 s
Mean time between assoc. attempts 23 s
Will improve with higher openness
Intriguing possibility for vehicular access

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

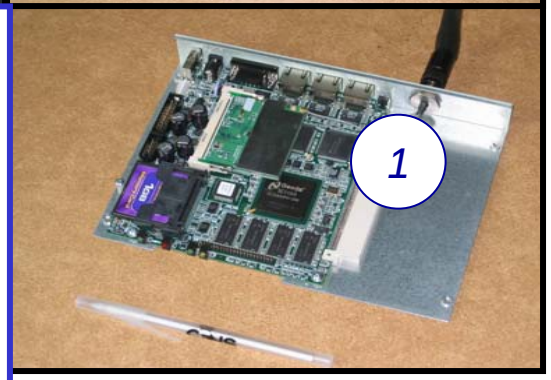
Other Vehicle-based Applications

- Smart route finding & congestion mgmt
  - Past + current data
- Fleet management
  - E.g., trucks, taxis, buses
- "Geo-logging" for road warriors, teenagers, etc.
- Visual mapping (images, video) of regions
- Civil and environmental monitoring
- Wireless network monitoring
  - "can you hear me now"

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

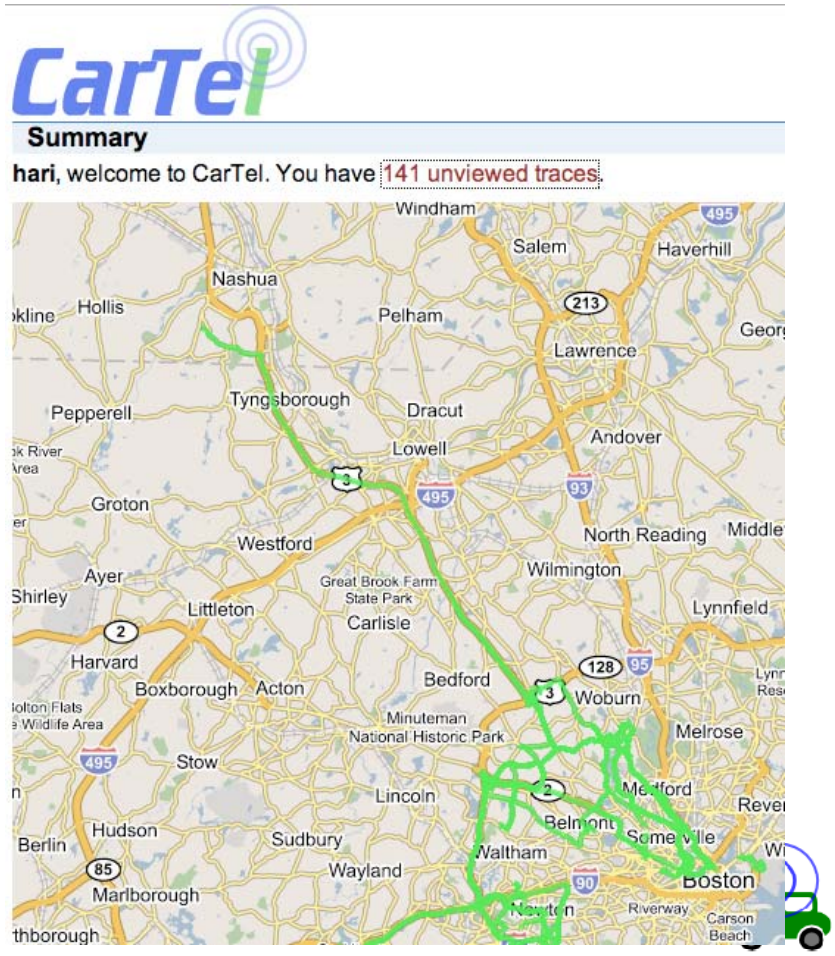
Current Status

- Various components built and being evaluated
- Small deployment and experiments with 6 cars
- “Alpha” deployment underway (20 cars)



1. **Embedded Computer**
2. **WiFi Antenna**
3. **GPS Receiver (Seattle)**
4. **OBD-II Adapter**

QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.



Current Status

CarTel AutoPortal

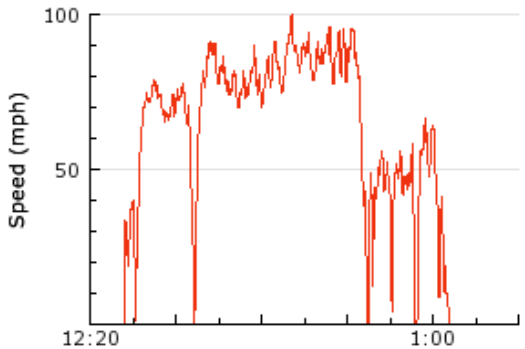
Trace Detail: Sat,03/04/06 - 12:23 AM

Duration:	00:38:21
Distance:	41.78 miles
Vehicle:	cartel7

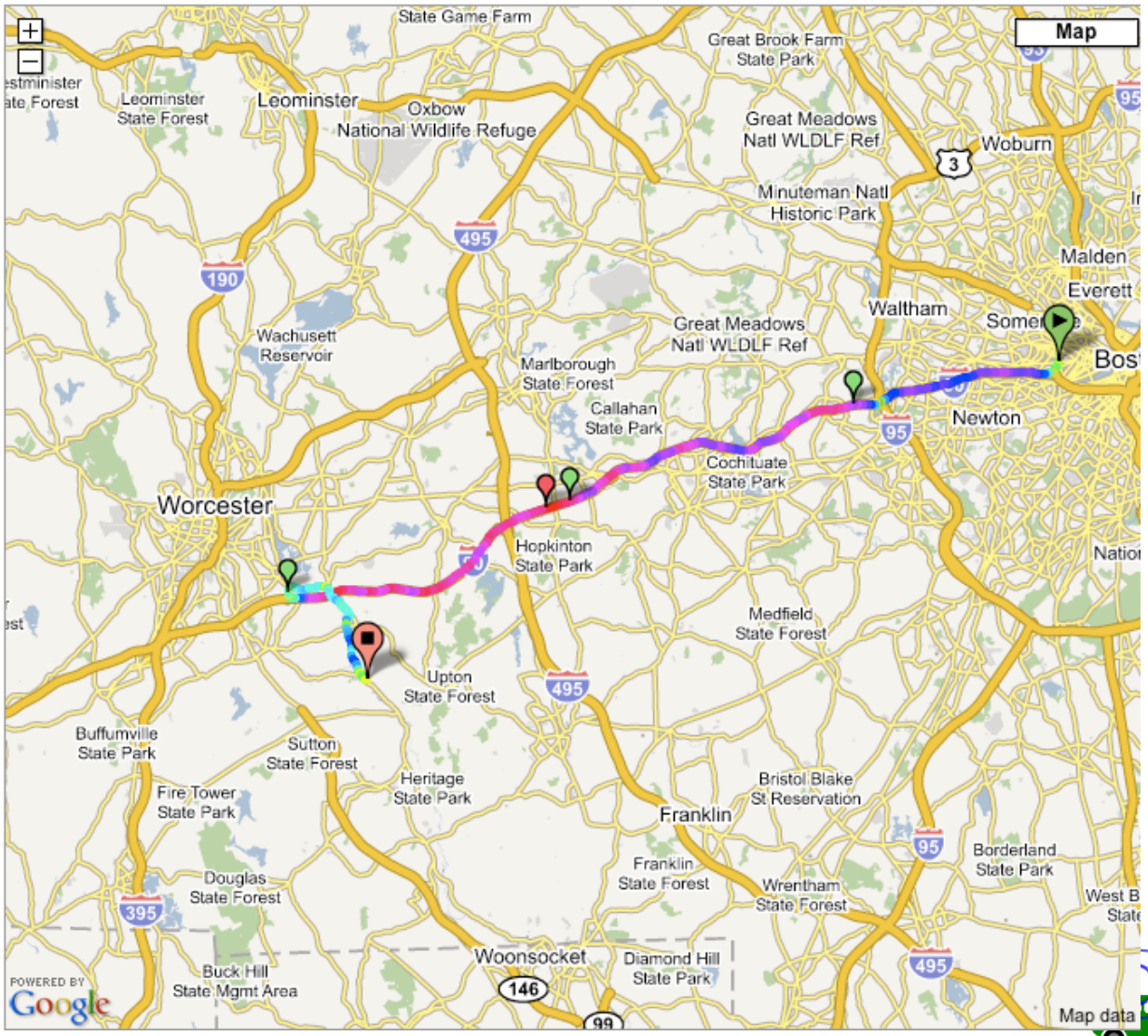
Sensor Data Overlays:

speed

Avg. Speed:	65.37 mph
Max. Speed:	99.84 mph



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



Lessons & Challenges

- Disconnectivity
- Prioritization
- Live, real-time, visualizations
- Lack of ubiquitous networking infrastructure
- Good environmental sensing hardware
- Performance of Postgres on low-end embedded hardware

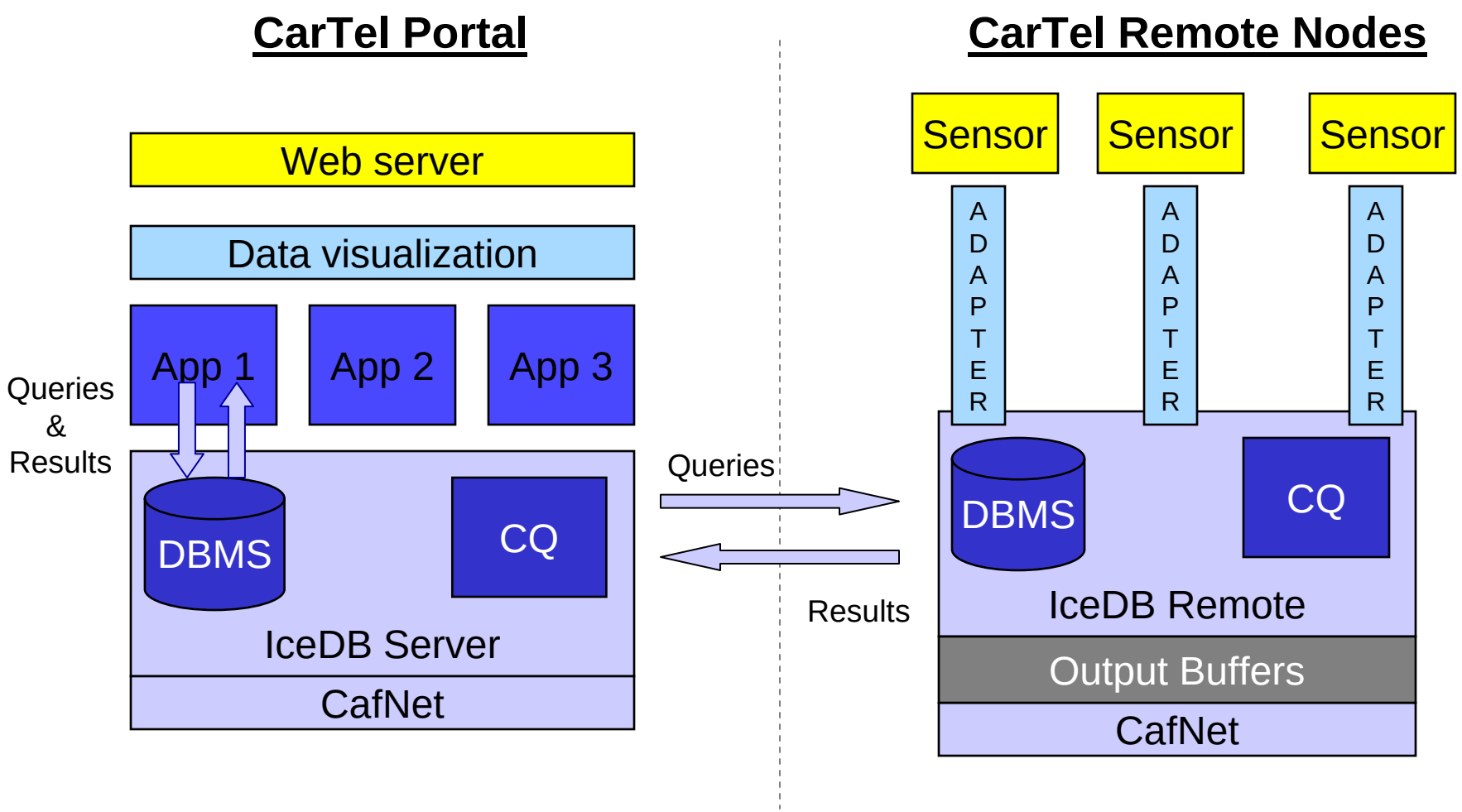
Conclusion

- Mobile sensor networks can sense at much higher scale over large areas than static networks
- Several applications: traffic, fleet management, automotive diagnostics, **wireless network monitoring**, civil/environmental monitoring, …
- Key challenges: heterogeneous data, intermittent connectivity, programmability, privacy
- In urban areas, Wi-Fi is a viable uplink technology
- For more info: http://cartel.csail.mit.edu

Questions?

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

CarTel Software Architecture

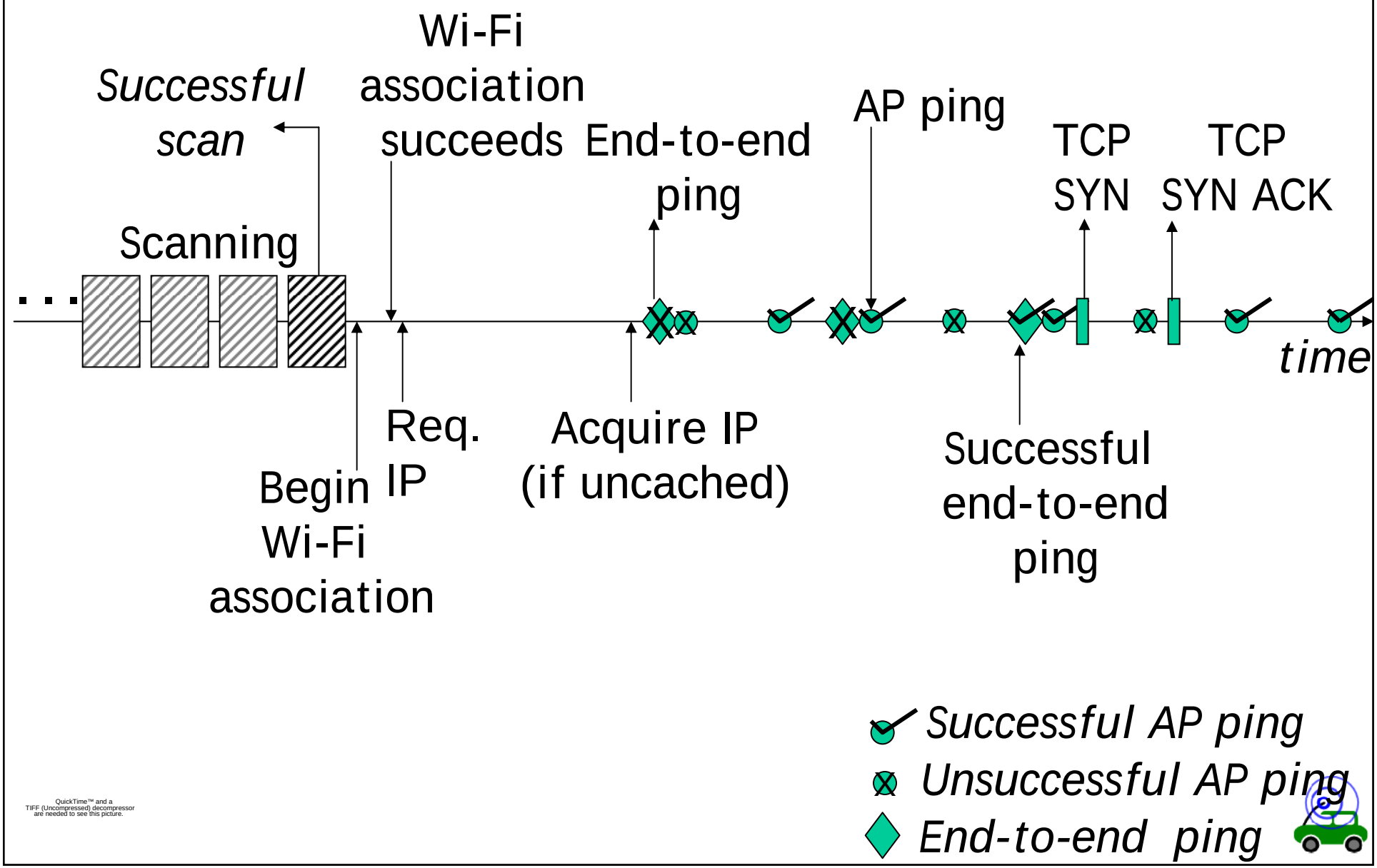


QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

Related Work

- Mobility in sensornets [Princeton/Zebranet,UCLA/NIMS,MIT/Underwater]
- DTN protocols: primarily focused on routing, but cf. [Rutgers/infostations,Intel/DTN-stack,Toronto/DTN-stack,CMU/DOT]
- Dist. query & stream processing: much work, but little when connectivity is intermittent [TinyDB,Cougar,Olston,Borealis]
- Back-end DBMS (geo-spatial, moving object databases)
- Traffic monitoring [JamBayes,traffic.com,smartraveler.com,inrix, Rutgers/TrafficLab,Umass/Diesel]
  - Focused on sensors on roads and/or real-time
  - CarTel instruments cars, mostly historic modeling
- Wireless / Wi-Fi monitoring
  - Very few previous studies of vehicular Wi-Fi

Timeline of Events

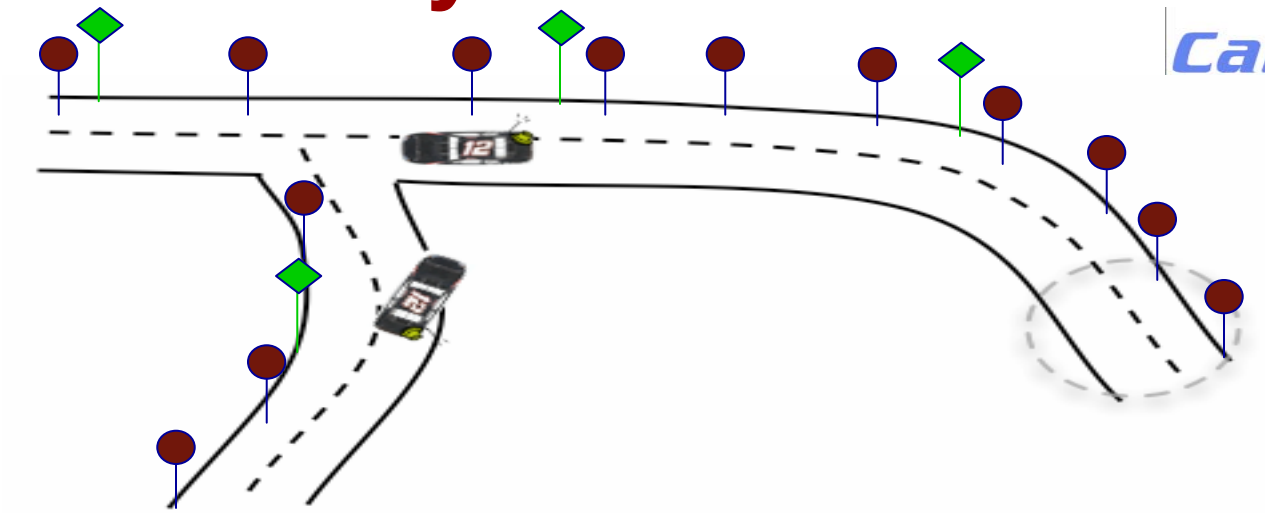


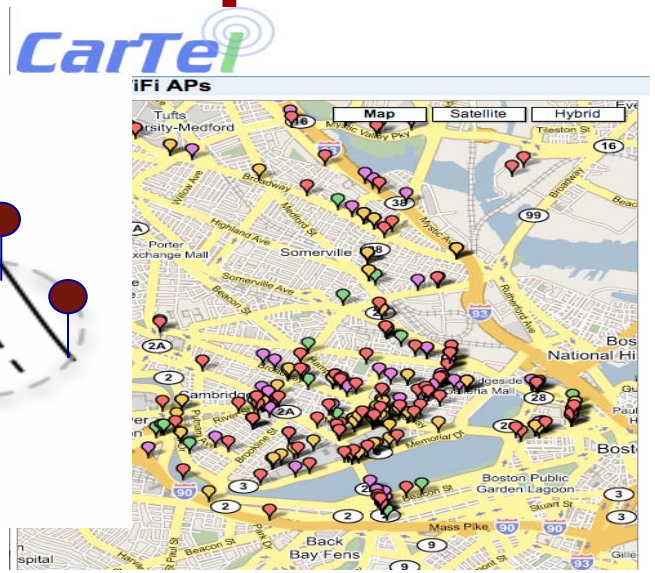
Determining Priorities and Delivery Order

- Q: In what order should data be sent to server?
- A: Utility = f(loc, time, sensor_type, sensor_value)

- Examples: Consider a car with GPS, camera taking pictures once per second
  - In what order should data be sent?
  - Maximal spacing between pictures?
  - Prioritize hot-spots?
  - Take into account what other cars there are?

Utility Maximization Example



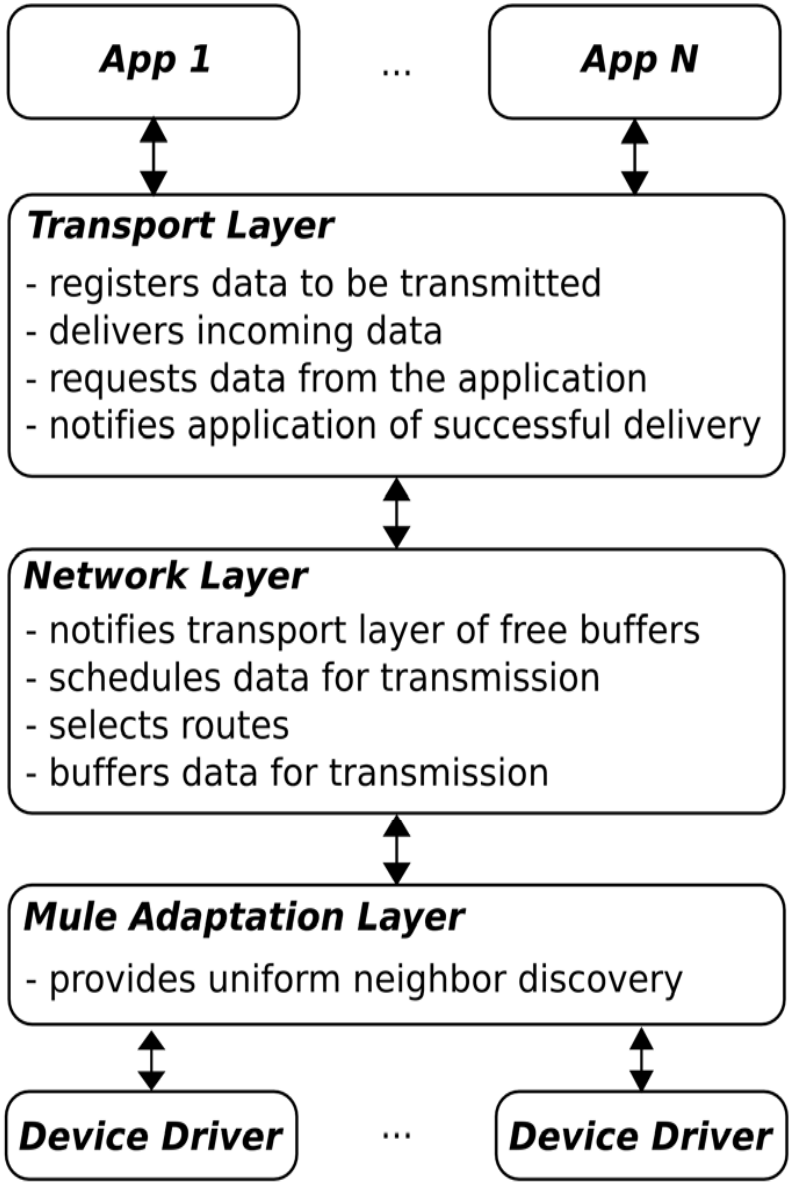


- Suppose data gathered at $P = \{P_1, P_2, \ldots P_n\}$
  - Can be sent at locations $C_1, C_2, \ldots C_m$
- User query for random position *x* at time *t*
- Want: data as current as possible, as near x as possible
  - Minimize: $E = \text{dist}(x, Q(t'))$, where $Q(t')$ is in P, subject to $t - t' < \delta$
  - A "delay-tolerant query"

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

CafNet: A Delay-Tolerant Network Stack

- Data moves through regions of highly variable connectivity
- "Mule" = element that stores data to be relayed toward the destination when "the time is right"
  - I.e., a delay-tolerant network (DTN)
- CafNet delivers results to portal and queries to nodes



QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

CarTel Software Components

- *CarTel Portal*: Centralized, visual user interface
- *IceDB*: Intermittently connected DB
  - Centralized declarative queries
  - Executed in distributed fashion by mobile nodes
  - Delay-tolerant continuous query processing
- *CafNet*: CarTel's network stack
  - Handles variable and intermittent connectivity

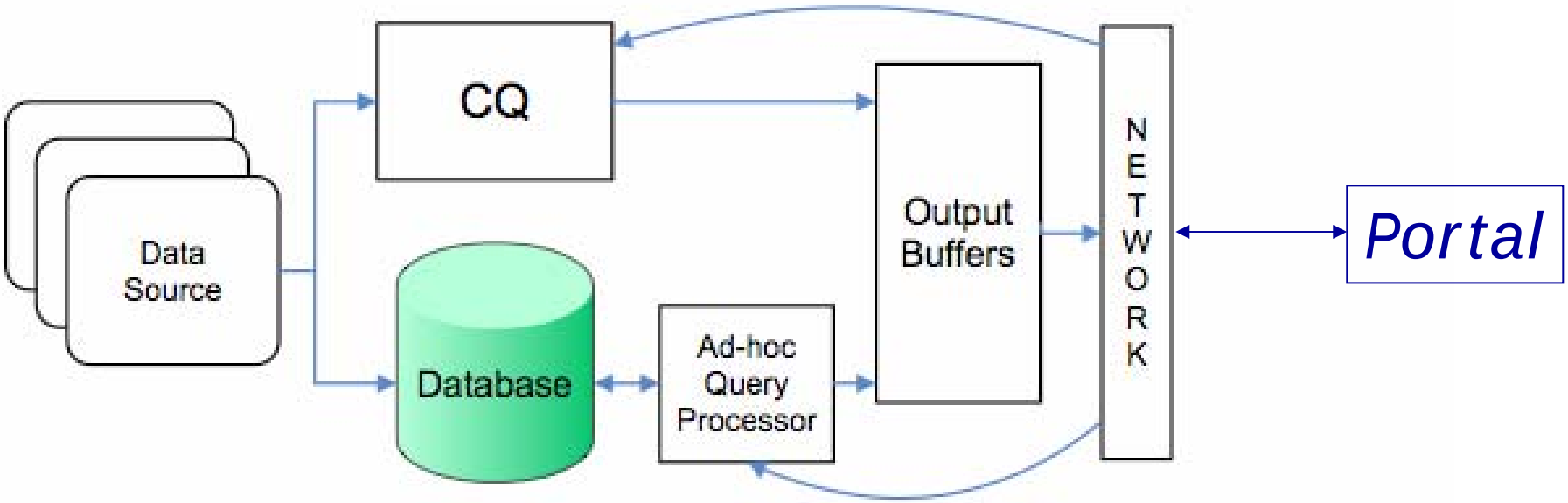
IceDB: Intermittently Connected DB

- Delay tolerant, distributed continuous query engine
  - Highly variable connectivity & bandwidth
- SQL extensions to handle intermittent connectivity
  - To prioritize results
- Adapters for managing heterogeneous data types
  - Meta-data package describing attributes of sensor
  - *Create* local tables for sensor readings
  - *Acquire* tuples from sensor
  - *Parse* sensor readings

Continuous Queries

- Current model for stream processing
  - Process data streams via long-running queries
  - Windowed aggregates, filters, windowed joins, merges, etc.
- Network is assumed to be “always on”
  - Disconnection is a fault to be masked (or a failure occurs)

Delay-Tolerant Continuous Queries



- IceDB stages data into output buffers to hide variable connectivity
- Key idea: Data in output buffers get re-evaluated dynamically, each time a new item arrives into it

Result Prioritization

- Limited BW necessitates deliberate ordering
- Three simple SQL extensions
  - For local (per-box) ordering:
    - `PRIORITY`
    - `DELIVERY ORDER BY`
  - For global ordering:
    - `SUMMARIZE AS`

PRIORITY

- *Idea*
  - Some queries are more important than others
- *Details*
  - Add **PRIORITY** clause to SQL
  - Drain output buffers in priority order

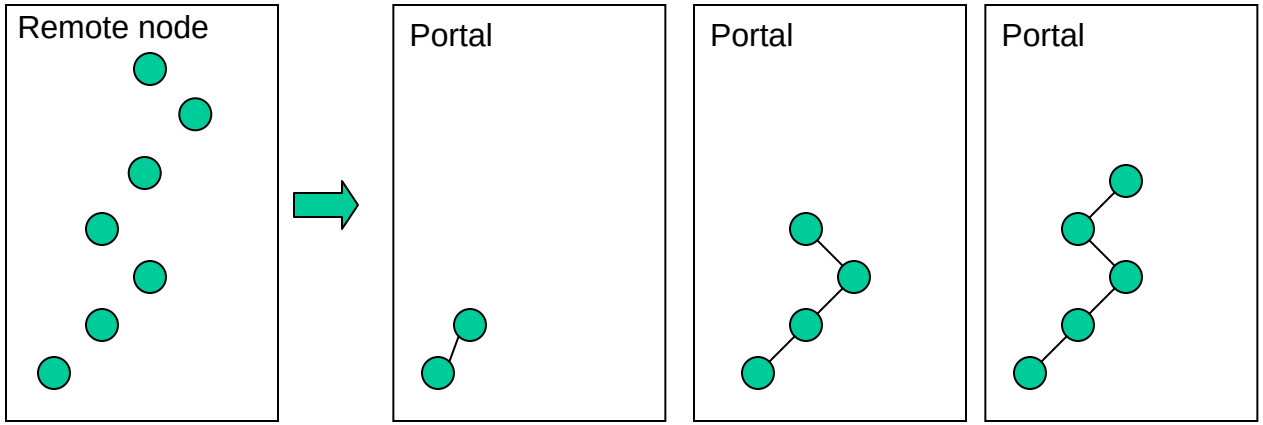
```
SELECT lat, lon, mph
FROM gps
WHERE mph > roads.mph_limit
BUFFER IN driver
PRIORITY 10
```

```
SELECT oil_temp
FROM obd
BUFFER IN engine
PRIORITY 1
```

DELIVERY ORDER BY

- *Idea*
  - Prioritize tuples to maximize incremental marginal utility
- *Details*
  - Query specifies transmission order via **`DELIVERY ORDER BY`** clause
  - User-defined ordering function (e.g. `bisect`)
  - Operates over entire query output buffer

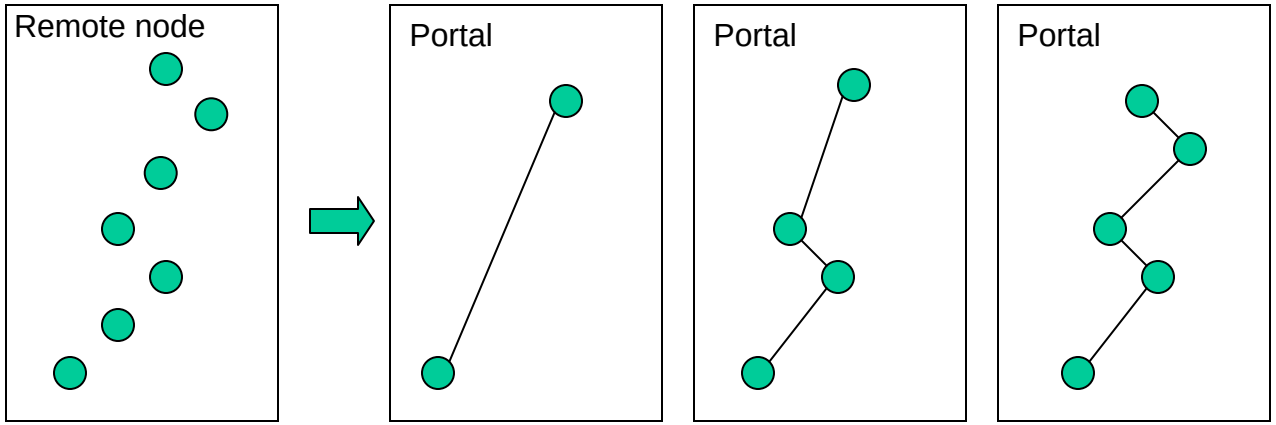
DELIVERY ORDER Example




```
SELECT lat, lon
FROM gps
WHERE insert_time
          > cqtime – 5
EVERY 5 seconds
BUFFER IN gpsbuf
DELIVERY ORDER BY fifo
```

FIFO

t=1 t=2 t=3




```
SELECT lat, lon
FROM gps
WHERE insert_time
          > cqtime – 5
EVERY 5 seconds
BUFFER IN gpsbuf
DELIVERY ORDER BY bisect
```

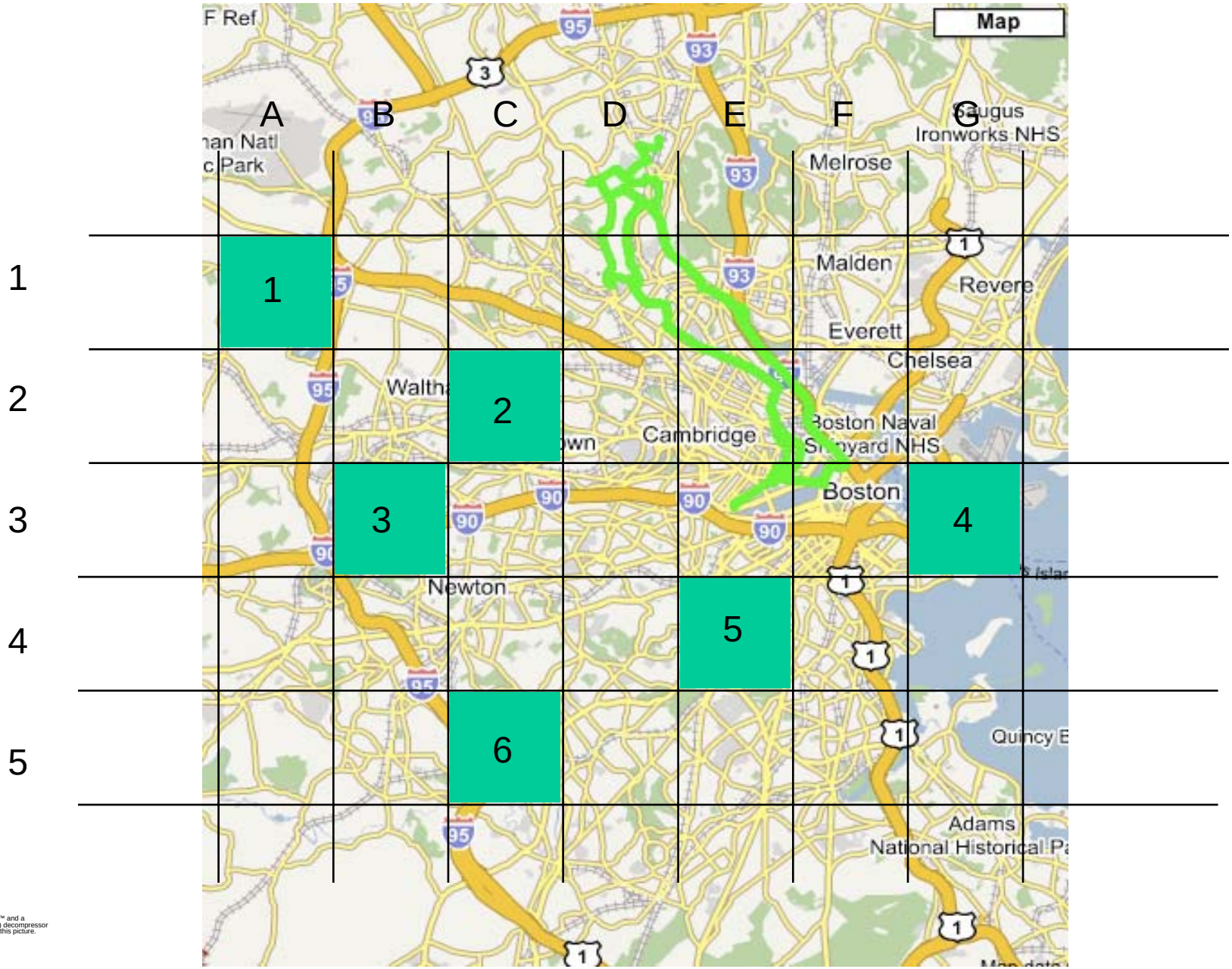
QuickTime™ and a TIFF (Uncompressed) decompressor are needed to see this picture.

BISECT

SUMMARIZE

- *Idea*
  - Nodes send server low-resolution summary of output buffer contents
  - Server sends back transmission ordering
- *Details*
  - Users specify “summarization query” alongside main query
  - Summary defines “summary segments”
  - Server ranks segments using app-defined metric
  - Ranking pushed to nodes to set output ordering

SUMMARIZE Example



QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

SUMMARIZE Example

```
SELECT lat, lon, image
FROM camera
WHERE insert_time > cqtime – 5
EVERY 5 seconds
BUFFER IN cambuf
SUMMARIZE AS
      SELECT floor(lat/100), floor(lon/100)
      FROM cambuf
      GROUP BY floor(lat/100), floor(lon/100)
```