Demo Abstract: MobiQuery - A Spatiotemporal Data Service for Sensor Networks

Sangeeta Bhattacharya, Octav Chipara, Brandon Harris, Chenyang Lu,
Guoliang Xing, Chien-Liang Fok
Department of Computer Science and Engineering
Washington University in Saint Louis
Saint Louis, MO 63130
{sangbhat, ochipara, bbh2, lu, xing, liang}@cse.wustl.edu

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication; C.2.1 [**Network Architecture and Design**]: Distributed networks; C.3 [**Special-Purpose and Application-Based Systems**]: Real-time and embedded systems

General Terms

Algorithms, Design, Experimentation, Performance

Keywords

Wireless Sensor Networks, Spatiotemporal Data Service, MobiQuery

The MobiQuery middleware provides a new data service that allows a mobile user (e.g., a human or a robot) to periodically gather information from surrounding areas. For example, a fireman fighting a wild fire may request timely update of a temperature map within one mile around his location to maintain awareness of the fire condition. A key feature of MobiQuery is its capability to provide robust guarantees on a set of critical spatiotemporal constraints on multi-hop sensor networks, even when nodes run on low duty cycles to conserve energy. The spatiotemporal constraints include the following.

- *Spatial constraints:* The evolving query area constrains the valid data sources contributing to the query result, i.e., all and only the nodes surrounding the user should contribute to the current query result as the user moves through the sensor field.
- *Temporal constraints:* These consist of *delivery rate* and *data freshness*. A new query result must be delivered to the user before the end of each querying period. Furthermore, each query result must be aggregated from *fresh* sensor data, where the freshness constraint of a sensor datum is defined by its maximum validity interval after it is read from the sensor.

MobiQuery has three phases of operation, prefetching, query dissemination, and data collection. The low duty cycles of sensor network nodes may result in poor data fidelity due to sleeping nodes missing the query. To account for this, MobiQuery adopts a novel *just-in-time prefetching* mechanism that forewarns nodes in estimated future query areas allowing them to reconfigure their sleep schedules and prepare the results in time. Our prefetching mechanism assures that all nodes in each query area will be able to deliver data to the user under the temporal constraints, while minimizing the storage cost for maintaining query-related states. The query dissemination phase distributes the query to all nodes in the query area. As the query is disseminated, a *query tree* that spans the entire query area, including the sleeping nodes, is constructed. During the data collection phase, the query tree is used to route and perform in network aggregation on the sensor data. The aggregated data is held by the root of the query tree until the user arrives, after which it is delivered to the user.

This demo evaluates the ability of MobiQuery to provide robust spatiotemporal guarantees in a network with low duty cycles and users with different motion profiles. It uses 18 MICA2 motes running MobiQuery arranged in a 6x3 grid deployed on the provided 6' x 2' table. Each mote is equipped with a MTS101 sensor board to collect temperature readings. The motes provide the MobiQuery service to a mobile user, which is emulated by a Crossbow Stargate mounted on top of an Acroname PPRK robot. The Stargate controls the PPRK robot via a dual RS232 serial port PC Card and provides an interface to the sensor network. The robot measures about 9 inches wide and 3 inches tall and travels at around one tenth of a meter per second. The Stargate uses the interface board as a base station to collect data from MobiQuery that it displays graphically on a laptop as a temperature map of the sensor field. For presentation purposes, the Stargate relays the temperature map to the laptop for display via 802.11b. During the demo, each mote logs the actions that it takes. At the completion of the demo, these records are aggregated and displayed on the laptop such that the operations performed by MobiQuery may be visualized.

The demo is divided into two parts: Part 1 demonstrates the inner workings of MobiQuery and Part 2 shows MobiQuery's ability to adapt to changing motion profiles of the user.

In Part 1, the PPRK travels along a preprogrammed path though the sensor field, collecting and displaying temperature data as it moves. When the PPRK reaches the end of its path, a laptop collects log files from the motes. The log files contain the actions taken by each individual mote. A Java program compiles these actions into a script which is played back in a graphic interface on the laptop's LCD. The graphics show how MobiQuery propagates Prefetching messages and builds a query tree. It demonstrates how MobiQuery provides periodic data update to the robot under spatiotemporal constraints while maintaining low duty cycles.

For Part 2, the PPRK operates autonomously. Using data collected through MobiQuery, the PPRK navigates to the coolest areas of the sensor field. To ensure that the demo is dynamic and truly demonstrates the flexibility of MobiQuery, a heat lamp is used to change the temperature in different areas of the sensor field. Because the PPRK must decide at each intersection which way to go, it dynamically update the motion profile provided to MobiQuery.

SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA.
ACM 1-58113-879-2/04/0011.