

Sensor Network Architecture for Cooperative Traffic Applications

Teemu Leppänen, Mikko Perttunen, Jukka Riekk
Department of Electrical and Information Engineering
University of Oulu
Oulu, Finland
{teemu.leppanen, mikko.perttunen, jpr}@ee.oulu.fi

Pekka Kaipio
Department of Information Processing Science
University of Oulu
Oulu, Finland
pekka.kaipio@oulu.fi

Abstract - The advancement of information and communication technologies accelerates the development and deployment of intelligent transportation systems. In the future cooperative traffic system, actors collect information about their immediate environment sharing it with each other, enabling all traffic entities to become context-aware. In this work, we contribute a practical data collection and processing system using mobile phones as sensor nodes. The system is demonstrated with two example applications, producing the visualizations of the travelled route annotated with the travel mode and the detected anomalies of the road surface. Modular component-based architecture enables the reuse of the system in future applications and the developed reconfigurable components facilitate dynamic additions of data processing algorithms into the system in runtime.

Keywords - Cooperative systems; sensor networks; wireless sensing.

I. INTRODUCTION

The increasing bandwidth available for wireless channels enables the advancement of intelligent transportation systems towards cooperative traffic systems. In [1], cooperative traffic was given the following description: “Road operators, infrastructure, vehicles, their drivers and other road users will co-operate to deliver the most efficient, safe, secure and comfortable journeys. The vehicle-to-vehicle and vehicle-to-infrastructure cooperative systems will contribute to these objectives beyond the improvements achievable with stand-alone systems.” These actors in traffic can be provided with information, for example, traffic disorders, road conditions, current weather conditions and usage statistics as well as route planning based on dynamic traffic information [2].

Mobile sensor networks offer several advantages over fixed sensor networks [3]. First, the coverage of a sensor network can be extended easily, reducing costs. Second, nodes can store data locally and provide data only when requested, which contributes towards scalability and energy saving. Third, if a fixed sensor node fails or experiences network failures, mobile data collectors can be used to compensate the data loss. Fourth, localized algorithms in nodes can be utilized in data collection and processing tasks.

Participatory sensing is a method for collecting data from many unknown and independent contributors in collaboration [4]. As mobile phones are today widely used, equipped with sensors, interfaces to external sensors and they have processor, memory, battery and communication units, they can be also seen as sensor nodes [5]. Mobile data

collectors, however, have communication issues when transferring data to the sinks as bandwidth is limited [3]. Also, battery life, processing power and memory capacity are limited. Normal phone usage will limit the data collection capability [5]. Data routing is influenced by the destination address, the quality of the service parameters and contents of the data [3].

Several participatory sensing systems have been reported. In the Mobile Millennium project [6], GPS enabled mobile phones are used for the collection of traffic data, which is then fused with data from sensors in the road infrastructure. The system is used for monitoring and estimating the traffic flow in real-time. The produced estimates are then transferred back to the phones. Google Maps for Mobile [7] offers live traffic conditions monitoring by retrieving the speed of vehicles from the GPS data from the mobile phones. TJam [8] uses GPS receivers to predict traffic jams by measuring the velocity of the vehicles. Users' coordinates are transmitted to the service and the probabilities of congestion in the given region are sent back. In Nericell [5], mobile phones with accelerometer sensors are used in vehicles to detect road bumps, among other features. The detection software is installed in the mobile phone itself and they introduce the concept of triggered sensing where less energy consuming sensors are used to trigger the usage of other sensors.

The Finnish Cooperative traffic program envisions sustainable traffic using extensive information sharing based on novel technologies and services for the levels of transport system [9]. In the Sensor Data Fusion and Applications project as a part of this program, our focus is on utilizing new sensor data sources and data fusion methods for generating new potential applications [9]. Our vision includes the usage of mobile phones with integrated sensors, giving information on the behavior of the actors in traffic. Instead of focusing on tailored applications on the field, *our goal is a system to be used to demonstrate and analyze a variety of cooperative traffic scenarios and applications working together*. The current work takes a bottom up approach, contributing a practical data collection and processing system with the ability to integrate heterogeneous data sources. We emphasize data fusion capabilities and pluggable data processing components. A visualization application producing annotated maps has also been developed, enabling end-to-end demonstrations. In this paper, we report our first results, the system prototype and the usage of two example applications, a real-time travel

mode and road surface anomaly detection, with the prototype.

The rest of this paper is organized as follows. In Section II, we give requirements and scenarios for a cooperative traffic sensor network and describe our system. In Section III, we compare the developed system against the requirements and represent field test results. Finally, in Section IV, thoughts for future work are given.

II. SYSTEM

We have identified various cooperative traffic scenarios to sketch the requirements for the initial data processing system. In some of the scenarios, the target user is a car driver, in some a pedestrian and in others a road operator. Based on the scenarios, we identified several requirements, for example: 1) a road operator should be able to query the number of vehicles passing a specific road section during a specific historical time interval, 2) A car navigator should be able to query up-to-date road condition information, which may be a result of fusing weather forecasts with road condition data and locally reported temperature measurements, 3) A road operator may wish to query the vibration profiles of a number of passenger cars from a specific road section during summer and to compare them with the vibration profiles in winter.

While we identified several scenarios, the specific sensors and the applications using a data processing middleware are not known. Thus, services offered by the middleware cannot be tailored and optimized as suggested in [10]. Furthermore, isolated data sources and heterogeneous communication networks set challenges for the systems [10]. Also, the lifespan of data is not known and quality and quantity vary.

A. General Requirements

We adapt several general requirements for sensor networks from [11] to the scope of cooperative traffic data processing middleware: 1) architecture should be data-centric as we will handle large amounts of heterogeneous sensor data, 2) asynchronous communication should be used because of communication outages, 3) component-based architecture and modular components follows from the diversity of the applications, 4) data processing middleware should have means for self-configuration to achieve runtime scalability with additional modules and to react to dynamic changes in network, 5) means for self-maintenance are required in case of sensor node failures, 6) support for unknown types of future sensors is required, 7) data processing components should be easily deployable to the system and 8) the system should be able to reconfigure functionality and launch components based on data requests from the client applications. Furthermore, from the project goals we derive the following requirements: 9) component chaining is required to implement a modular system with data fusion capabilities, 10) output interfaces should be well defined and extendable to enable the usage of the resulting information in a wide variety of client applications and 11)

real-time performance is a major issue when considering the usage of information in traffic.

B. System Description

The purpose of this prototype system is to demonstrate the chosen applications and test the feasibility of the system. The two applications function as follows: middleware receives vibration and GPS location data from sensors, extracts feature vectors from the data, classifies the vectors and then visualizes the results in a web browser. Currently, we have three main components in the prototype system: mobile phones as sensor nodes with the data collection software and sensors, the middleware for data processing in a remote platform in public network and a client application showing a map in a web browser on end-user workstation. The developed middleware architecture is shown in Figure 1. The system functions also as a sensor network test bench for data processing algorithms under development.

Our choice of architecture is data-centric and using centralized database, as we want all the data to be rapidly propagated for the needs of any client application. We would like to have as much as possible raw data in our system, not to limit the future use of the collected data sets. Also we do not want to limit the data acquisition methods available, thus streaming, polling and event-based acquisition all can be used (req. 6). Data producers are decoupled from consumers, because the commonly available sensor nodes in the system such as mobile phones should not be concerned about the data processing needs of client applications. The remote data processing platform is located in the infrastructure, because we use global endpoints for entering sensor data to the system and for heterogeneous communication methods, regardless of the physical location of the sensor nodes or the applications (req. 2).

In the developed mobile sensor node, data are collected in frequency of 30 hertz from the built-in accelerometer sensor and GPS receiver. Data are stored temporarily on the phone memory and sent in given intervals as HTTP POST requests over GPRS network to the remote platform. Current implementation is done in Python for the Nokia N95 mobile phone.

We selected Global Sensor Network (GSN), a sensor network middleware developed in EPFL [12], as a middleware platform for our system. As we are looking for complete open source implementation, for example partially released CarTel developed at MIT cannot be considered [13]. Lightweight Java implementation, the usage of a subset of modules and minimal configuration needs make GSN deployable to large sets of system configurations. Simple API and the number of already implemented features reduce the amount of required implementation work. The dynamic deployment of sensor nodes within the GSN is handled just by adding component configuration files to the system [12] (req. 7). The basic configuration defines the input/output data streams for a component. Chaining and data element fusion are enabled through defining the data input and output streams for the components (req. 9). The runtime reconfiguration of system functionality can be done by modifying the configuration file, and then GSN will

dynamically start the required Java objects in the system (reqs. 4 and 8). The objects are alive only as long as required, and there is built-in fault tolerance system for components [12] (req. 5). Also, the dynamic use of components will not interfere with on-going data processing as the data streams are shared [12]. Addition of new data types in the system introduces the data schema evolution problem in the current components. This can be solved in GSN, as it offers means for filtering out unwanted data items for the input data stream of a component (req. 6). In case several components request the same data items, overlapping data queries are internally handled in the GSN middleware. The output data in all phases of processing can be saved to a database (req. 1).

We utilize web server as a sink node in the remote platform as shown in Figure 1. The received HTTP POST requests are parsed in proxy and forwarded to the middleware by a wrapper component. In the middleware, the data (as key-value pairs: client id, timestamp, location, accelerometer data, application specific type of location) are streamed through MySQL database from component to component at all steps of processing thus even intermediate processing results can be utilized immediately in simultaneous algorithms and rapidly propagated to client applications (reqs. 1 and 3). This is also shown in Figure 1. For the required application-specific data processing, we have developed template components and configurations for the use of data processing algorithms. These components can be deployed to the system any time through endpoint handlers, which will generate new configuration files to the system thus launching new data processing component (reqs. 3, 4, 6, 7 and 8). The data processing algorithms itself and their parameters are defined in the template configuration and can currently be, Java objects, executables or even external services in the network. Also, the required types of location (in the example applications: stationary, walking, jogging, bicycling, driving, anomaly, pothole) for the component are given in the configuration. Finally, the endpoint handler components assemble XML-based GPX format document from requested data to response queries from client (req. 10). The data flow of the example applications is shown in Figure 2. Detailed descriptions of the data processing algorithms used in the applications can be found in [14] and [15].

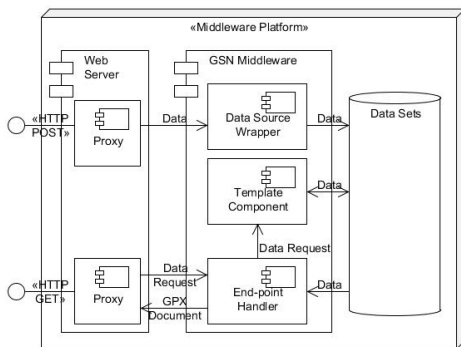


Figure 1. Middleware architecture.

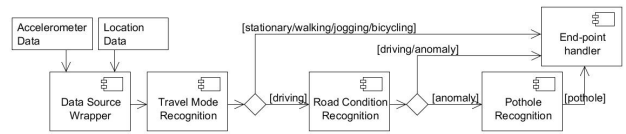


Figure 2. Data flow in the example applications.

The GPX documents are used to create an annotated map in the visualization client prototype in a web browser. The application is developed using Ajax technology and uses HTTP GET requests to receive data from the middleware. The map shows the path history, the current location of the mobile client, its travel mode and detected anomalies as markers. The dynamic content is displayed with OpenLayers on top of maps from OpenStreetMap.

We conducted small scale field testing for the system in real environment by walking and driving a vehicle in the city of Oulu, simultaneously collecting data using mobile phones and transferring it to the sink node in the intervals of 10, 30 and 60 seconds.

III. RESULTS AND DISCUSSION

Considering the requirements for sensor network system implementation, we have demonstrated data-centric system architecture able to dynamically reconfigure functionality and allowing runtime deployment of sensors and data processing components into the middleware. Furthermore, the system has support for asynchronous communication and unknown types of future sensors and applications by providing public interfaces. The implemented example applications prove the system's ability to run multiple applications simultaneously, its ability for self-configuration and implementing component chaining for data processing.

In comparison to existing systems [5-8, 13], we offer in addition runtime pluggable data processing components, which are launched based on client application requests. Google Maps for Mobile offer cooperative traffic applications, which are based on location data solely. CarTel, TJam and Nericell locally process data on the nodes. TJam also uses migratory services in nodes, which is a feature we might consider in the future. The Mobile Millennium uses roadside sensors and historical data, both of these features we would like to have in the future.

In our field tests, using GPRS for demonstration, we calculated average amount of data transferred per data transmission to be 4, 11 and 23 kilobytes for the intervals of 10, 30 and 60 seconds. Measurement data sample size was 100 bytes. We also estimated required bandwidth per node being less than three kilobytes per second for all the intervals. As expected, during field experiments receiving GPS signal was sometimes disrupted in cities and insignificant communication network problems were experienced. The battery lifetime was a limitation when using mobile sensor nodes, but as our phones were often mounted in vehicles, their power system was used. The cost of air time and available bandwidth also varies. Also, we tested the feasibility of GSN middleware in normal desktop

PC (2.40 gigahertz processor with 1 gigabytes of RAM running Windows XP) with randomly generated data sets. We used 50 data sources simultaneously streaming 50 kilobytes of payload to the system every 20 milliseconds. The delivery time from a data source wrapper to a data processing component was less than 1 millisecond per payload. This is very promising result considering real-time capabilities of the system (req. 11). However, the used communication network and programming skills of the developers largely contribute to real-time performance in the prototype.

In this work, we have not considered security and privacy issues, which are unavoidable when collecting and visualizing data from multiple clients and with the integration of roadside infrastructure data and sensors.

IV. CONCLUSION AND FUTURE WORK

We have started with a generic implementation of sensor network middleware, which meets the given requirements drawn from literature and from the project goals. Considering the goals, we have demonstrated modular component-based prototype sensor network implementation capable of sharing traffic related information collected by mobile sensor nodes and realize the given usage scenarios.

In the next phase, the developed prototype system will be deployed to use with multiple sensor nodes and client applications. Increased scalability needs may require introducing multiple middleware platforms for distributing the system load. An option is to have preprocessing components in the mobile sensor nodes, but this limits the usability of raw data on the middleware for future data processing needs. An important goal of our future work is to implement a common data format and a data fusion model in the middleware. Another feature for development is the retrieval of historical data from the database. This includes information on vehicle type, location, route, event, time, road condition or any other parameter. Also, we will develop interfaces to sensors in instrumented vehicles and in roadside infrastructure, such as video cameras and weather stations. Security and privacy concerns can be addressed by for example introducing user groups with restricted access to the data and results.

ACKNOWLEDGMENT

The authors would like to thank all the personnel in the Sensor Data Fusion and Applications project. This work was

funded by the National Technology Agency of Finland and was carried out in Sensor Data Fusion and Applications project as a part of the Cooperative Traffic research program of the Strategic Centre for Science, Technology and Innovation in the Field of ICT.

REFERENCES

- [1] F. Minarini, "eSafety for Road Transport: Investing in Preventive Safety and Co-operative Systems, the EU Approach," *Advanced Microsystems for Automotive Applications 2005*, pp. 487-492, 2005.
- [2] Cooperative Vehicle-Infrastructure Systems: <http://www.cvisproject.org/> 16.06.2010
- [3] V. Dyo, "Middleware design for integration of sensor network and mobile devices," in *Proceedings of the 2nd international Doctoral Symposium on Middleware*, pp. 1-5, New York, NY, USA, 2005.
- [4] C. Haythornthwaite, "Crowds and Communities: Light and Heavyweight Models of Peer Production," *Hawaii International Conference on System Sciences*, Vol. 0, pp. 1-10, 2009.
- [5] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*, Raleigh, NC, USA, pp. 323-336, New York, NY, USA, 2008.
- [6] The Mobile Millenium Project: <http://traffic.berkeley.edu/> 16.06.2010
- [7] Google Maps for Mobile: <http://www.google.com/mobile/maps/> 16.06.2010
- [8] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Context-Aware Migratory Services in Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 6, 12, pp. 1313-1328, 2007.
- [9] The Finnish Cooperative Traffic Program: <http://www.cooperativetraffic.fi/> 16.06.2010
- [10] Y. Yu, B. Krishnamachari and V. Prasanna, "Issues in Designing Middleware for Wireless Sensor Networks," *IEEE Network*, Vol. 18, pp. 15-21, 2003.
- [11] K. Modukuri, S. Hariri, N. Chalfoun, and M. Yousif, "Autonomous Middleware Framework for Sensor Networks," *Journal of Pervasive Computing and Communications*, vol. 1, pp. 337-345, 2008.
- [12] K. Aberer, M. Hauswirth, and A. Salehi, "A middleware for fast and flexible sensor network deployment," in *Proceedings of the 32nd International Conference on Very Large Databases*, Seoul, Korea, pp. 1199-1202, 2006.
- [13] B. Hull et al., "CarTel: a distributed mobile sensor computing system," in *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems*, Boulder, Colorado, USA, pp. 125-138, New York, NY, USA, 2006.
- [14] J. Kantola, M. Perttunen, T. Leppanen, J. Collin, and J. Riekkii, "Context Awareness for GPS-Enabled Phones," in *Proceedings of ION Technical Meeting 2010*, Manassas, VA, 2010.
- [15] M. Perttunen et al., "Participatory Road Condition Monitoring Using Mobile Phones," unpublished.