iRide: A Cooperative Sensor and IP Multimedia Subsystem Based Architecture and Application for ITS Road Safety

Muslim Elkotob and Evgeny Osipov

Department of Computer Science and Electrical Engineering
Luleå University of Technology, SE–97187 Luleå, Sweden

Abstract. In this paper we present iRide (intelligent ride), an IP Multimedia Subsystem (IMS) application for warning drivers about hazardous situations on the road. iRide takes real-time information about road conditions and traffic situations from a wireless sensor network installed directly in the road surface. Upon logging to the iRide system, users start to receive periodic updates about the situation on the road along their route ahead. iRide is able to predict hazardous situations like slippery surface or dangerous distance to the nearest car and help drivers avoid accidents. We describe the service and the supporting network architecture of iRide. We discuss the major challenges associated with designing an IMS application for ITS, an intelligent transport system. Having a prototype implementation working on a small scale, we take it to the next step to perform system dimensioning and then verify the feasibility of having such a system using OPNET simulations.

1 Introduction

In this paper, we describe a new solution associated with the design of one part of a communication framework for a cooperative road infrastructure system (CRIS). CRIS aims at making the road surface intelligent and is being developed in the scope of the iRoad project comprising a constellation of Swedish governmental, industrial and academic partners [8]. iRide is based on road marking units (RMU) containing a set of sensors measuring instantaneous properties of the road. They are connected to a microcontroller with a low-power radio transceiver. RMUs are joined to form a wireless sensor network.

When it comes to warning a driver about a hazardous situation beyond the visibility of on-board safety systems and the line of sight of the driver, we chose the IP Multimedia Subsystem (IMS). In this article, we present iRide, an IMS application and the supporting communication architecture for preventive hazard warning in the iRoad CRIS. Information about road conditions is collected in the network of on-road sensors and transmitted via a 3G backhaul link to the IMS servers for real time processing and hazard analysis. iRide updates users about the situation ahead their route. To the best of our knowledge, iRide is among the first attempts to apply the IMS framework in the context of ITS.

R. Mehmood et al. (Eds.): EuropeComm 2009, LNICST 16, pp. 153–162, 2009.

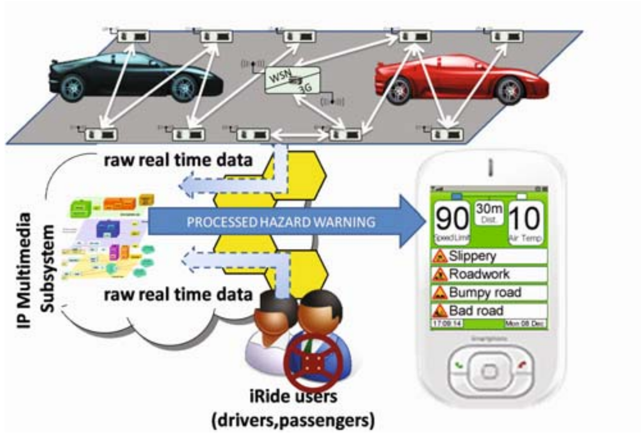

Fig. 1. iRide actors and information flow

This paper is structured as follows. In Section 2 we overview the major design challenges when developing the iRide application. In Section 3 we present the iRide service logic and network architecture with some insights on performance requirements and evaluation. Section 4 provides an extensive overview of related work. Section 5 is a discussion of future and an outlook on subsequent research and development for iRide. We conclude the paper with Section 6.

2 Design Space and Solution Outline

In Figure 1, the high level logic behind the iRide application is illustrated including a screenshot of the client-side midlet on a smartphone. The wireless sensor network (WSN) formed by intelligent road marking units (RMU) installed directly in the road surface is able to continuously monitor road properties. Information observed by the sensors is transmitted wirelessly over multiple hops to a gateway to a 3G network. Inside the WSN, data is transmitted using low power radio technology. Currently, the iRoad RMUs are equipped with IEEE 802.15.4-based radio transceivers. The 3G modem is installed on a road side unit (smart sign, fence, camera pole). In [1] the feasibility of such transmission in real-time was demonstrated. A WSN-3G gateway forwards raw data over the IMS infrastructure to the iRide application server (AS) for further processing. iRide users entering the intelligent road area login to the system from their mobiles.

The functional logic behind the iRide warning application is shown in Figure 2. The system jointly processes the information about road conditions and car motion. The results of this processing are prognoses of hazardous situations. The system can determine when the speed of an iRide user is too high for a currently slippery road or that the distance to the ahead going car is decreasing too fast and there is a risk of collision. When such an event is detected, iRide sends a warning to the respective users. For our implementation, we used the Mobile Java Communication Framework (MJCF)[10].

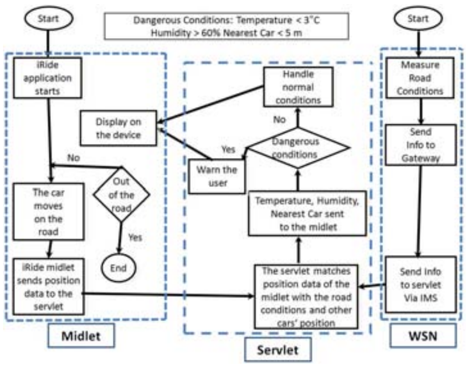

Fig. 2. iRide service logic

3 iRide Design and Implementation

This section presents several important aspects for iRide design and implementation. One is the service logic where the major states and the transitions leading to them are outlined. Another aspect is the data representation and information flow. Then the signaling flow within the iRide system is discussed.

3.1 iRide IMS Architecture and Service Logic

The iRide service logic and architecture are illustrated in Figures 2 and 3 respectively. An essential part of the iRide application is the multi-hop wireless sensor network integrated with a wireless infrastructure connecting IMS clients to the main system (Figure 3). The raw data from each road marking unit is transmitted and then via a GPRS or 3G modem to a radio access network (RAN). In the application server data is entered into the database indexed by the coordinates of the RMUs. Data from the user terminal travels a similar RAN-GGSN-AS path. WSN and IMS client information paths meet at the IMS control plane. Table 1 shows the data involved in the prediction of hazardous situations in iRide.

The data in Table 1 is transmitted to the application server using SIP protocol messages. The IMS client displays the right audio-visual primitive to the iRide user based on the command received from the application server. The iRide process on the application server predicts hazardous situations by doing joint processing of data received from the client's midlet and the WSN. Upon arrival, all messages are time-stamped at the server. We achieve virtual synchronization between the WSN and mobile terminal clocks. The set of events and the corresponding warning signals generated by the servlet are shown in Table 2. The distance parameter D is calculated by the servlet for every pair of back-to-back cars based on the position information supplied periodically by the midlets. Our servlet maintains two lists of X and Y coordinates one for each direction of the intelligent road. When a new car enters the appropriate segment, its coordinates are appended to the tail of the list. In this way, the order of records in the list

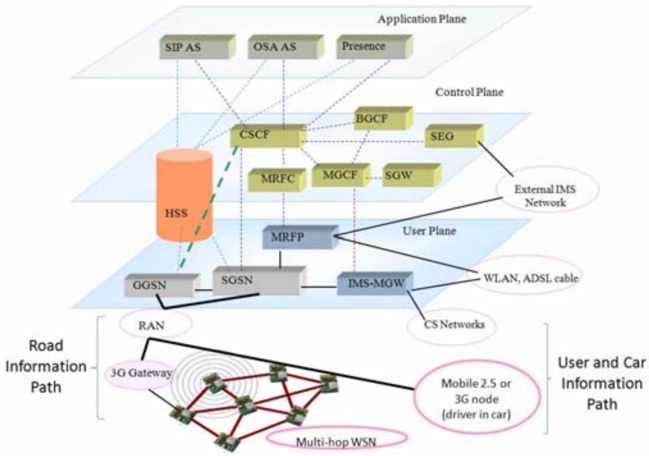

Fig. 3. IMS architectural model of iRide

Table 1. *iRide* data table

Data	**Function and Purpose**	**Source of data**
Msg_id	Midlet identifier for sorting requests	terminal
Road_unit_id	RMU identifier for the servlet	road
Km_no	Absolute position of RMU on the road	road
T	Temperature of the road surface	road
H	Humidity around the car	road
Pos	Position of a car on the intelligent road	terminal

indicates the relative position of the cars on the road. The servlet process goes periodically through each list in a round robin manner, calculates the Euclidean distance between each pair of cars, and checks sensor measurements.

A snapshot of the current iRide IMS client implementation is shown in Figure 1. The color scheme of the graphical warning profile and the intensity of the appropriate audio warnings depend on how critical the situation is. We also animate the relative position of the iRide user to the closest car in front or behind (whichever is more critical) and show the actual distance numerically. The warning profile includes graphical primitives for road signs associated with particular iRide signals, such as "slippery road", "bumpy road", etc. Some information can be supplied into the system by road authorities such as "road work".

3.2 iRide Implementation Details in MJCF

In MJCF, midlets use a complex data structure called a record store which is responsible for Authentication Authorization and Accounting (AAA) as well as

Table 2. iRide events and actions in the prototype implementation

Events	Warning or Danger alerts
$T \leq 3°\text{C}$	Warning: slippery road
$D \leq 5\text{m}$	Danger: minimum distance too small
$H \geq 70\%$	Warning: limited visibility due to fog
$(H \geq 70\%)\&\&(5\text{m} \leq D \leq 15\text{m})$	Danger: critical distance in bad visibility conditions
$(T \leq 3°\text{C})\&\&(5\text{m} \leq D \leq 15\text{m})$	Danger: critical distance on slippery road
$(H \geq 70\%)\&\&(T \leq 3°\text{C})\&\&$ $(5\text{m} \leq D \leq 15\text{m})$	Danger: critical distance on slippery road with limited visibility

IMS presence information for iRide users. A servlet on the backend handles the record store data in a watcher-list. For midlet-servlet communication, we create a class called MessageProtocol whose simple attributes contain the data shown in Table 1. iRide communication is implemented using the MJCF built-in IMS methods publish, unPublish, setNote and the IMS class geoPriv. The setNote method is used to convey the current presence status. The geoPriv class on the servlet is used to update the location coordinates of the midlet. Once the midlet updates its state information, it calls the publish method. Upon exiting the application, the unPublish method is called. The message sequence chart in Figure 4 summarizes the signaling used in iRide.

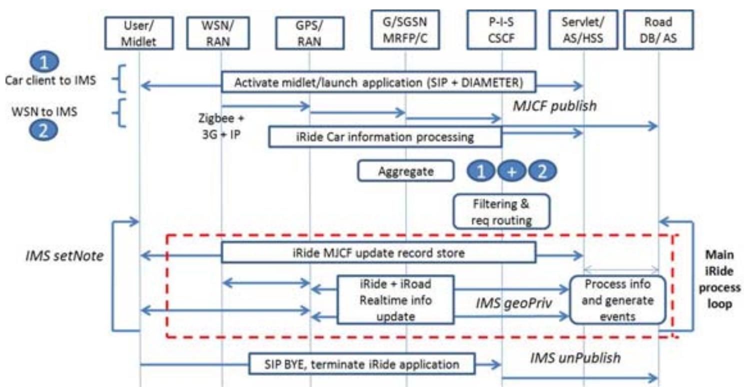

Fig. 4. Message sequence chart and signaling flow

3.3 iRide System Requirements

Here we dimension the bottleneck performance regions of the IMS architecture for worst case iRide scenarios. The road is split into segments each of which is assigned an iRide server process responsible for tracking and warning all vehicles. The major iRide IMS bottleneck is the Call Session Control Function (CSCF) proxy, the place where requests from individual users meet. If the frequency of user requests exceeds the CSCF capacity, iRide will be unable to provide continuous service to all users. The performance of the CSCF proxy can be significantly improved by adding extra processing units (blades) [7], [13]. Assume an iRide covered road segment of 100 km. Taking an average of 10 cars per 50 m (totally crammed road), we have 20000 cars in the segment. The smallest iRide cycle for information update for each user is 5 seconds. We used an average SIP message size of 800 bytes and 3 messages per transaction causing the arrival rate of iRide requests at the CSCF proxy to be about 4000 requests per sec (rqps). Downlink traffic for warnings gives an additional 2000 rqps. The total load on the proxy sums up to 6000 rqps. The fastest CSCF proxy known to us has eight blades and preemptive FCFS request scheduling; it is able to process up to 2500 rqps [7]. Therefore we need three such proxies to ensure continuous service.

3.4 iRide Estimated Performance

Figure 5 shows the topology we constructed using the OPNET [11] simulator. It includes several data gateway machines that aggregate ZigBee traffic and 3G traffic from road units and drivers in the vicinity of a particular subnet (labeled mobile subnet). All road and user traffic is aggregated into a main iRide data gateway and then sent via the core network towards a load balancer that dispatches requests to 3 CSCF units.

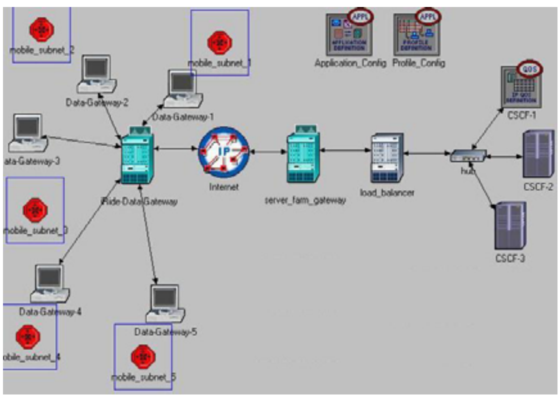

Fig. 5. OPNET middle tier and backend topology network for iRide

iRide relies mainly on SIP-based instant messaging and small file transfer that contains images or audio warnings. Dainotti et al. [6] have classified traffic and its behavior when it comes to identifying invariance in the behavior of TCP-based applications such as SMTP and HTTP. According to the statistical traffic analysis studies conducted, above 80% of all traffic has a lognormal distribution for inter-packet arrival times. This means that with comparable packet sizes for iRide system messages going through the access network, IP core, and into the IMS system, data traffic will have a single-tailed lognormal distribution. With

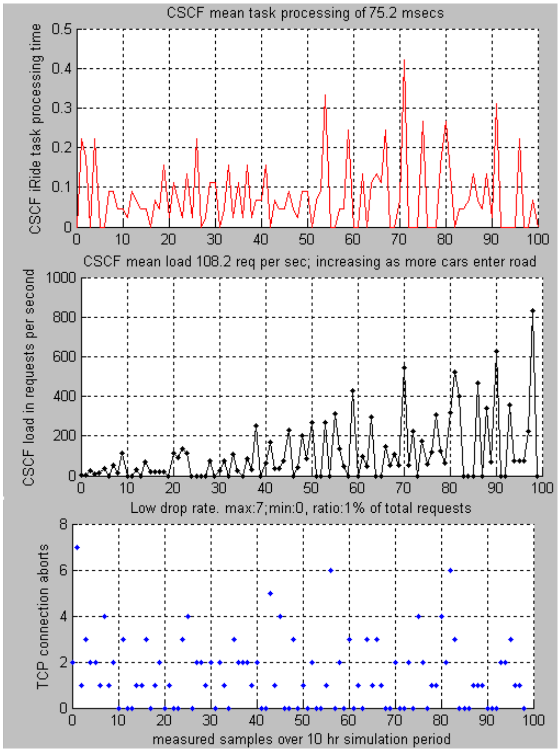

Fig. 6. Performance of CSCF server units and TCP-SIP sessions in iRide

1% packet loss rate in the IP backbone, we obtain a connection abortion count in the range [1-7]. With the load balancer acting as a dispatcher and sending requests to the CSCF module with the least load, the load exerted on a single CSCF module was increasing over the 10 hr simulation period but still having the bulk of load values in the range of 0-200 tasks per second (Figure 6).

The overall delay cycle of a message in iRide is 200 msecs in each direction plus 100 msecs processing delay adding up to half a second. A car moving at 80 kmh covers 11 meters in that time; at 100 kmh it covers 14 meters. This is the sensitivity range of iRide. A warning has to be issued when the critical distance approaches the calculcated values if moving at the respective speeds.

Having presented the iRide IMS architecture, service logic, signaling flow, and an overview of system requirements and dimensioning, we move on to the next section where we discuss related work in the areas of IMS and ITS.

4 Related Work

iRide, which uses information from a wireless sensor network together with an IMS infrastructure for improving road safety, is a new use-case scenario for IMS.

In this section, we analyze some related work in the area. One line of work concerns integrating new access technologies with IMS such as cable in [9]. In the scope of iRide, the challenge is to integrate a WSN with IMS. We conjecture that native support for WSN/3G gateways in the IMS architecture is essential for future IMS-ITS related applications. Blum et al. [4] focus on service creation and delivery for SMEs within IMS-based environments and following a Software Oriented Architecture (SOA) paradigm. Bachmann et al. [3] point out IMS client development challenges. While designing and implementing the iRide service architecture, we used some experiences from [2]. In contrast to the results of the existing ITS related projects [5], [14], [12], we do not require any hardware or sophisticated mechanisms in the car. iRide only requires that its users have smartphones able to run the iRide Java midlet.

Steuer et al. [17] present a connectivity management solution for vehicular telemedicine applications in heterogeneous networks. The solution is relevant to iRide because it involves communication between vehicles (ambulances) and the closest medical center in order to remotely handle a stroke by the time paramedics arrive on the spot. Williams et al. [19] survey the state of the art in ITS indicating that it has become a key feature to have reliable bi-directional links between a vehicle and an infrastructure. GLIDE (Green Light Determining System), ERP (Electronic Road Pricing System), and EMAS (Expressway Monitoring and Advisory System) are discussed. Singapore is used as a showcase in the paper. Furthermore, some standardization efforts in the ITS area are outlined, including the ISO TC204 WG1 on System Architecture initiative and WG4 on Automatic Vehicle Identification. In the V2I (vehicle to infrastructure) area, the paper outlines key projects including: CVIS [12], SAFESPOT [14], and COOPERS [5] (Europe) and SMARTWAY [16] (USA).

The Idris automatic vehicle detection and classification patented technology and software [15] uses in-ground loop technology recording inductance as a vehicle passes over, whereas we use vibration sensors as a way to detect passing vehicles. Idris includes a Single Stopped Vehicle (SSV) algorithm to automatically detect when a vehicle abruptly stops and disrupts the normal flow of traffic. iRide has at its core, a car warning system which is based on a near-distance detection algorithm we used in our implementation.

5 Discussion and Further Work

In iRide, key information elements are road surface conditions (humidity, temperature) and positions of cars. We considered a rather simplified case where either all iRide users are equipped with GPS-enabled smart phones or a specialized device in the car communicates this information to user terminals.

Future work will include methods of getting position information for cars where no GPS support is available. Information will be acquired directly from the road that is able to detect vehicles by observing the vibrations in the asphalt and changes in the magnetic field of the environment. The feasibility of such passive autonomous detection of vehicles is described in [18]. Another issue associated with current cooperative ITS and tracking of cars in safety applications is revealing the car's identity and the associated privacy issues. In iRide we do not detect driver identity, thus preserving user privacy. IMS treats the client side with the driver's device as a Java midlet (IMS client) with a unique process id coupled to the backend side Java servlet on the application server. This property of iRide gives some hope for wider acceptance of the application.

The IMS architecture and framework provides methods that allow good joint handling of the two sets of data in iRide. IMS has a mature evolving architecture but still needs some adjustments to be suitable for integration with new access technologies (e.g. WSN in iRide) and safety-critical applications (e.g. ITS). Openness, modularity, and interworking are the key success factors needed for ITS to work properly with IMS and our paper together with its proof-of-concept implementation is an eye-opener. We would also like to acknowledge the efforts of the whole development team that contributed to iRide which includes besides us authors Giuseppe Lisi and Jonas Innala.

6 Conclusions

The IP Multimedia Subsystem can be applied to many different domains such as conversational, content-delivery, and real-time multimedia interactive services for entertainment and health-care. In the area of intelligent vehicular transportation systems, IMS has not yet been widely applied to the best of our knowledge. We presented an original design and proof-of-concept implementation of iRide, an IMS application for early warning of drivers about hazardous situations on the road. When using IMS in connection with ITS, several architectural and technical challenges appear. A particular challenge addressed in iRide is the merging of two sets of information, one from the wireless sensor network and one from the cars themselves. While developing iRide, we identified a spectrum of issues that need to be resolved before using the application on real roads. We however conclude that all these technological issues are possible to solve and the overall usage of IMS in the context of intelligent transport systems is feasible.

References

1. Hessler, A., Westhoff, D., Osipov, E.: Encrypted persistent data storage for asynchronous wireless sensor networks. In: 13th Annual International Conference on Mobile Computing and Networking, 2007. ACM Mobicom demo. (2007)
2. Alhezmi, A., Elkotob, M., Mrohs, B., Rack, C., Steglich, S.: Next generation service architectures: Challenges and approaches. In: ASWN 2006: Proceedingsof the Sixth International Workshop on Applications and Services in Wireless Networks, Berlin, Germany. Fraunhofer IRB Verlag (2006)
3. Bachmann, A., Motanga, A., Magedanz, T.: Requirements for an extendible IMS client framework. In: MOBILWARE 2008: Proceedings of the 1st international conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications, ICST, Brussels, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), pp. 1–6 (2007)

4. Blum, N., Magedanz, T., Stein, H.: Service creation & delivery for sme based on soa / ims. In: MNCNA 2007: Proceedings of the 2007 Workshop on Middleware for next-generation converged networks and applications, pp. 1–6. ACM, New York (2007)
5. COOPERS Integrated Project. Co-operative systems for intelligent road safety (2008), http://www.coopers-ip.eu/ (accessed 2009-01-12)
6. Dainotti, A., Pescape, A., Ventre, G., Jubin, J.: A packet-level characterization of network traffic. In: Proc. Globecom 2008 (December 2008)
7. Ghosh, P., Roy, N., Basu, K., Das, S., Wilson, P., Das, P.: A case study-based performance evaluation framework for cscf processes on a blade-server. In: ICNS 2007: Proceedings of the Third International Conference on Networking and Services, Washington, DC, USA, p. 87. IEEE Computer Society, Los Alamitos (2007)
8. iRoad. Project web site (2009), http://www.iroad.se/ (accessed 2009-01-12)
9. Mani, M., Crespi, N.: How ims enables converged services for cable and 3g technologies: a survey. EURASIP J. Wirel. Commun. Netw. 8(3), 1–14 (2008)
10. Mobile Java Communication Framework. Ericsson ims innovation development framework (2009), http://www.imsinnovation.com/ (accessed 2009-01-13)
11. OPNET. Opnet simulator (2009), http://www.opnet.com/ (accessed 2009-03-15)
12. CVIS Integrated Project. Cooperative vehicle infrastructure systems (2008), http://www.cvisproject.org/ (accessed 2009-01-12)
13. Rajagopal, N., Devetsikiotis, M.: Modeling and optimization for the design of ims networks. In: ANSS 2006: Proceedings of the 39th annual Symposium on Simulation, Washington, DC, USA, pp. 34–41. IEEE Computer Society, Los Alamitos (2006)
14. SAFESPOT Integrated Project. Co-operative systems for road safety: Smart vehicles on smart roads (2008), http://www.safespot-eu.org/ (accessed 2009-01-12)
15. IDRIS Diamond Consulting Services. Idris automatic vehicle detection and classification technology (2009), http://www.idris-technology.com (accessed 2009-01-12)
16. SMARTWAY. Project web site (2009), http://www.epa.gov/smartway (accessed 2009-01-12)
17. Steuer, F., Geithner, T., Kaschwig, T., Bür, K., Albayrak, S.: A connectivity management system for vehicular telemedicine applications in heterogeneous networks. In: TridentCom 2008: Proceedings, ICST, Brussels, Belgium, pp. 1–10 (2008)
18. Birk, W., Lundberg Nordenvaad, M.: Surface mounted vehicle property sensing for cooperative road infrastructre systems. In: ITS World Congress (submitted 2009)
19. Williams, B.: What's new in intelligent transport systems standards. Synthesis Journal, 51–58 (2008)