A FLEXIBLE APPLICATION LAYER PROTOCOL FOR AUTOMOTIVE COMMUNICATIONS IN CELLULAR NETWORKS

Yi Chen[1], Guido Gehlen[2], Gordian Jodlauk[2], Christoph Sommer[3] and Carmelita Görg[1]
chenyi@comnets.uni-bremen.de, guido.g.gehlen@ericsson.com,
gordian.jodlauk@ericsson.com, christoph.sommer@informatik.uni-erlangen.de,
cg@comnets.uni-bremen.de
[1] Communication Networks, University of Bremen, Germany
[2] Ericsson GmbH, Eurolab R&D, Germany
[3] Computer Networks and Communication Systems,
University of Erlangen-Nuremberg, Germany

ABSTRACT

The German research initiative "Adaptive and Cooperative Technologies for the Intelligent Traffic" (AKTIV) aims at the long-term improvement of both road traffic safety and traffic flow. The goal of one of its projects, namely "Cooperative Cars" (CoCar), is the development and the evaluation of tailored application layer protocols for car-to-car (C2C) and car-to-infrastructure (C2I) communication using cellular communication technologies. The project focuses on 3rd generation (3G) cellular technologies, namely the Universal Mobile Telecommunications System (UMTS), and upcoming 3G-based cellular mobile technologies. This paper presents the results of the application layer protocol design and the mapping to the standardized protocol framework of the Transport Protocol Expert Group (TPEG). The presented design aims at the development of a flexible and extensible suite of platform-independent CoCar-specific protocols in order to meet different user and communication requirements with high service quality and low end-to-end delays. A prototype system implementation provides the basis for first evaluations.

KEYWORDS

Protocol Design, TPEG, Car-to-Car, Car-to-Infrastructure, Cellular, UMTS, HSPA, MBMS

INTRODUCTION

Mobile communication is considered as a key technology for innovative cooperative vehicle applications in the near future. Various C2C and C2I communication technologies have been developed and are still under development, like e.g. Dedicated Short Range Communication (DSRC) and Wireless Access in the Vehicular Environment (WAVE) including the WiFi based specification 802.11p. In Europe, a lot of research by the car industry has in the last couple of years focused on the use of WiFi ad-hoc and multi-hop technologies for car communications. Some interesting and related projects are the Global System for Telematics (GST) (1), PReVENT (2), COMeSafety (3) and Safespot (4).

Cellular communication technologies, however, received less attention in these projects or are only used as a feedback channel in uplink direction, although UMTS-based multicast and broadcast technologies, e.g. the Multimedia Broadcast Multicast Service (MBMS), are able to support cooperative vehicle applications. The project CoCar uses a set of application layer protocols in both up- and downlink direction. These protocols utilize the benefits of new features of cellular communication systems, e.g. the current High Speed Packet Access (HSPA) system improves downlink and uplink data rates and transmission latencies that typically lead to Round Trip Times (RTT) of about 50ms. Fig. 1 illustrates the overall CoCar system architecture. Using 3G infrastructure, vehicles are able to report traffic information to dedicated Reflector components deployed in the CoCar system. These reflectors take care of the timely dissemination of received information in the immediate vicinity of the reporting vehicle. In a second step, they relay it to a higher-level component, the Aggregator, which uses Geocast Service components to take care of setting up geo-casts in affected regions and to facilitate broad dissemination of relevant information. The Aggregator component collects all the traffic information received in a wider area and derives a global view of the traffic situation, generating statistics and road traffic forecasts.

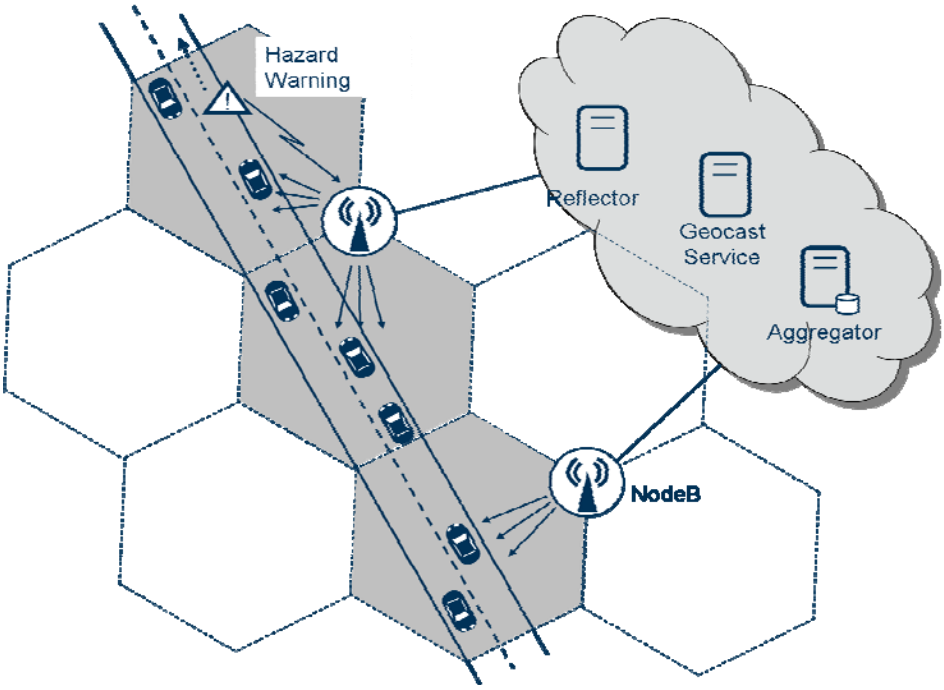

Fig. 1. Core components of the CoCar system architecture

The bulk of CoCar messages, as well as all messages in the CoCar core system, are exchanged using a flexible, interoperable protocol based on the TPEG standard, which will be presented in the following chapter. It should be noted that to cope with functional and performance requirements, to guarantee reliable and timely delivery of messages, some use cases in part also rely on a set of lightweight protocols, which, however, are considered to be out of scope for this paper.

THE TRANSPORT PROTOCOL EXPERTS GROUP PROTOCOL

The standardized TPEG protocol provides application services by transferring Traffic and Travel Information (TTI) via one or more delivery technologies following the standards of e.g. Digital Audio Broadcasting (DAB), Digital Video Broadcasting (DVB), the Association of Radio Industries and Businesses (ARIB), the Advanced Television Systems Committee (ATSC) or the Internet. The TPEG specifications comprise two separate message representations, TPEG Binary and tpegML. TPEG Binary is a space-efficient description used for digital radio delivery, while tpegML is an XML implementation developed for Internet services. Currently three message classes, called applications in TPEG terminology, are standardized through CEN and ISO:

- Service & Network Information (SNI) application (6)
- Road Traffic Message (RTM) application (7)
- Public Transport Information (PTI) application (8)

Others still under development are:

- PKI: Parking Information
- CTT: Congestion and Travel-Time information
- TEC: Traffic Event Compact (by the German national project mobile.info)
- WEA: Weather information for travelers

In the first phase of the CoCar project a representative set of applications that are relevant for safety, traffic flow and additional services for drivers have been selected. These pilot applications have been classified and structured with regard to different communication methods and parameters. For most of these applications, the TPEG data model introduced in (5) can be considered a perfect fit for the various requirements. TPEG-SNI, TPEG-RTM and TPEG-PTI, as well as other currently defined message classes cover most of the relevant TTI applications. However, none of them can fulfill all of the CoCar pilot application requirements. The gaps are listed below.

- As a unidirectional protocol in a broadcasting environment, TPEG messages are only designed to provide information by a service provider. In CoCar pilot applications, a flag is added to indicate whether a message was generated by a vehicle.
- Some information defined in CoCar pilot applications are missing in TPEG applications, e.g. message priority, values of acceleration and deceleration, planed route, and heading information etc.

EXTENDING TPEG WITH A NEW APPLICATION

One straightforward way of extending TPEG to match the project’s requirements would be to integrate the missing CoCar information elements into one of the TPEG applications that are already being standardized. The TPEG-RTM application, for example, already covers the traffic information required by the CoCar pilot applications. As this way of extending TPEG would, however, mean modifying standardized applications, it is not deemed feasible for CoCar.

Therefore, a new TPEG application, the “CoCar Application”, which builds upon – but is independent from – other standardized TPEG applications, has been developed. Conformant to the TS series tpegML, the CoCar application messages are formatted using XML as well. By this means, the flexibility of CoCar pilot applications is improved: On the one hand, it can be used alone without any other TPEG application; on the other hand, it can be easily integrated into a TPEG message stream and transferred to the client user with other applications.

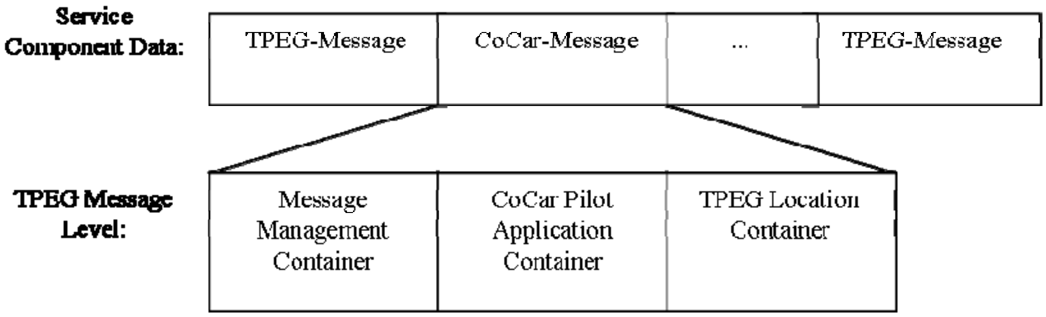

Fig. 2. Structure of a CoCar Message

In order to guarantee that the messages are readable by both TPEG and CoCar application users, the CoCar application messages are structured in the same way as TPEG application messages. Fig. 2 illustrates a structured CoCar message following the TPEG message containers concept. At the application level, a CoCar message is transmitted as service component data in a parallel manner with TPEG message. Similar to TPEG messages every CoCar message has three elements which reflect the demands of the message content. A CoCar message component includes:

- One Message Management Container which contains management information for the whole message. The management information includes application_id, message_id, version_number, priority, timestamp, start/stop time, etc.
- One CoCar Pilot Application Container holding a CoCar pilot application.
- One Location Container with the location reference for the overall traffic message. The location reference defined in TPEG specifications supports three different location referencing methods, tpeg-locML, AGORA-C and RNS-TMC, for the map-based as well as non map-based applications.

The standalone CoCar application is designed to cover the selected CoCar pilot applications. It inherits the TPEG message structure and data types, and adapts existing TPEG tables. Fig. 3 gives an overview of how the CoCar Application fits into the existing TPEG structure.

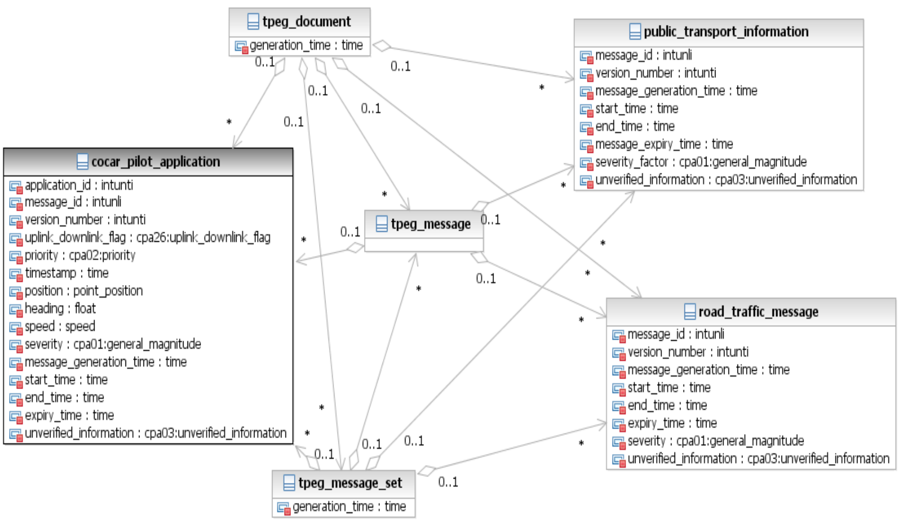

Fig. 3. UML Model of a TPEG Document containing the new CoCar Application

As the root element of tpegML, a tpeg_document component comprises the combination of tpeg_message or tpeg_message_set or application specific messages (e.g. TPEG_PTI, TPEG_RTM and extended CoCar pilot application). In the cocar_pilot_application box, it not only contains the message management elements defined in TPEG specifications, but also extends this field with CoCar specific elements. The extensions listed in Tab. 1 are mandatory information required by each CoCar pilot application.

Name	Type	Occ.	Notes
application_id	intunti	1	Identifies the CoCar application
uplink_downlink_flag	cpa26:uplink_downlink_flag	1	A flag for distinguishing uplink and downlink messages
priority	cpa02:pritority	1	Identifies the event priority
timestamp	time	1	Date and time for the generation time of the event
heading	float	1	Measured heading, 0º-north, 180º-south, 90º-east, 270º-west, course over ground
speed	speed	1	Speed over the ground (SOG)

Tab. 1 CoCar Mandatory Information

More details on the proposed protocol architecture and its integration into the TPEG protocol framework will be given in the following section.

UMTS-BASED COCAR APPLICATION SERVICE STRUCTURE

Fig. 4 shows a UMTS-based CoCar Traffic Information Center (TIC) and an integration of CoCar specific protocol stack with existing protocols. The TIC is designed to exchange TTI information with specially equipped vehicles through current cellular network UMTS. The location of TIC is at the Internet. It communicates with wireless networks via the UMTS Core Network (CN) components, e.g. the Gateway GPRS Support Node (GGSN) and provides interfaces to other TTI service providers. The CoCar specific protocol is built upon the Internet Protocol (IP) on top of the UMTS protocol stack. The selection of transport layer protocols, e.g. the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP), depends on the functional requirements as well as on performance requirements of the applications, like the message transmission delay. The TIC is an essential component that can collect and process CoCar specific application messages from authorized cars and TTI service providers.

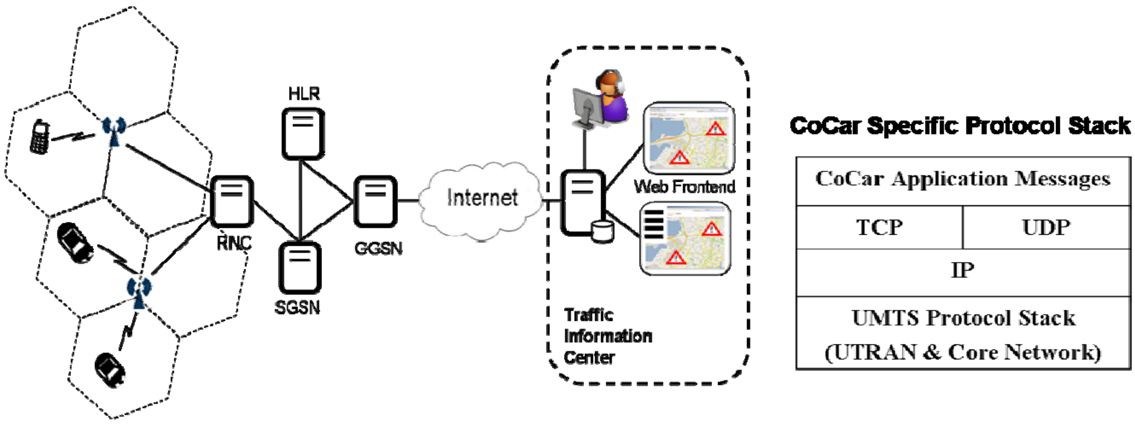

Fig. 4. CoCar specific Traffic Information Center and Protocol Stack

PROTOTYPICAL IMPLEMENTATION

As shown in Fig. 5, the prototype implementation of the TIC comprises of functional modules according to the needs of the system functional requirement that developed in the requirement analysis. As the implementation language is Java, all the messages processed in the TIC are presented in Java objects.

The received messages are parsed in the message parser and distributed according to their types. If it is an urgent message, it is forwarded to the reflector directly. As the reflector is responsible for one UMTS cell, the message will be delivered to the clients that are covered by this cell. All the received messages will be stored in the database connected to aggregator, including the messages processed in reflector. The aggregator collects messages from front-end system and external systems, and forwards messages to a geo-cast service if necessary. Geo-cast is a special case of broadcast, it uses cell ID to distinguish the location of the clients, which means messages are broadcasted to the clients who have an identical cell ID.

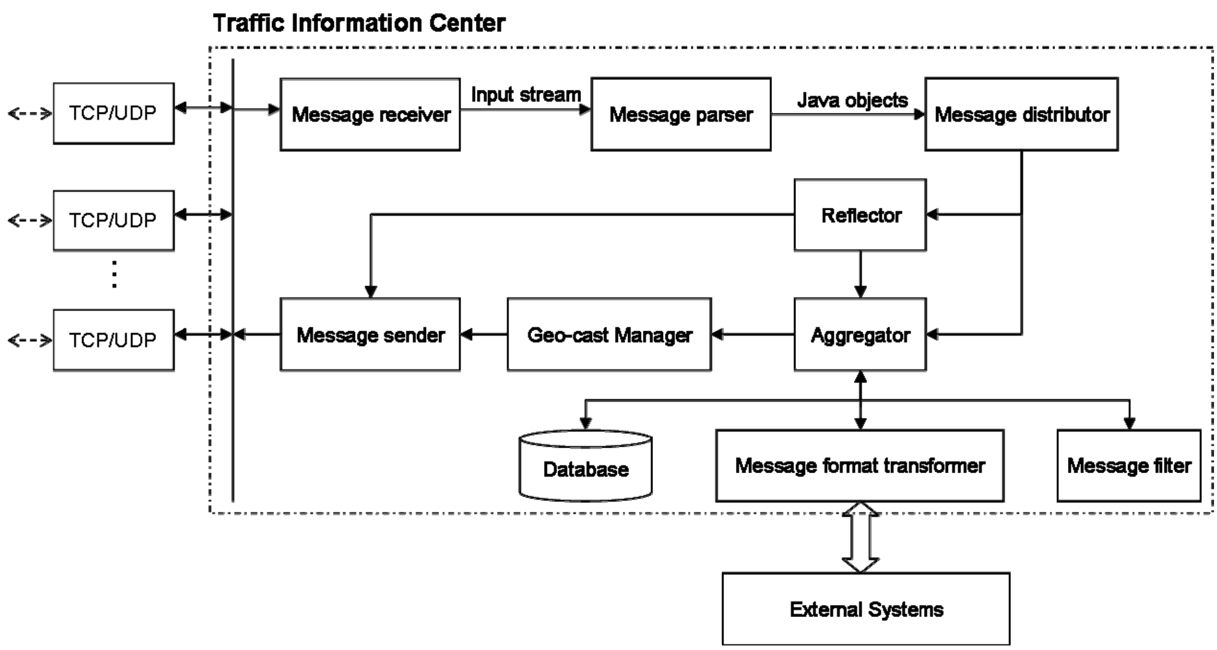

Fig. 5. Traffic Information Center Overview

- **Message Receiver**: listens to the network connections and forwards the received data stream to message parser.
- **Message Parser**: processes incoming data streams and transforms them to the TIC understandable Java objects.
- **Message Distributor**: distributes messages to reflector and aggregator according to the application urgent level.
- **Reflector**: quickly reflects received messages to vicinity area, normally, it is a cell area. If the area is larger than one cell area, the reflector will forward the messages to aggregator.
- **Aggregator**: exposes retrieved information to a variety of clients (e.g. vehicles, media services or other applications).
- **Geo-cast Manager**: provides broadcast, geo-cast and multi-cast services. If necessary, the geo-cast manager will setup a connection with broadcast services provided by cellular network, e.g. MBMS provided by UMTS network.
- **Message Format Transformer**: supports multiple message formats, e.g. XML, binary and formats defined in external systems. The interfaces to external systems are implemented in this model. Message can be freely converted within it.
- **Message Filter**: filters, modifies, updates stored messages and logically merges duplicated messages according to the predefined criteria.
- **Message Sender**: as message receiver, message sender is another interface to access network protocols in downlink.
- **Database**: stores incoming and outgoing messages and deletes the message when they are expired.

PERFORMANCE ANALYSIS

During the design of application protocols, XML is selected for application message presentation. XML is well-known as a simple, flexible, platform-independent language for representing structural information exchanged across application components, but the processing of XML messages is highly inefficient. In order to evaluate the effort for processing XML formatted messages, a test server of TIC is setup.

To generate a realistic data traffic load a combined vehicular and network traffic model (9) which is based on the road traffic simulator SUMO and the network simulator OMNeT++ is being used. It covers a large area with over 2000 moving cars which are reporting their positions and driving status every second. Assume only the pilot application messages Floating Car Data (FCD) are forwarded to the TIC. According to the designed applications in the previous section, the FCD message has a size of 1541bytes in XML. Thus, the data traffic throughput is calculated at the test server, which has a maximum value around 24Mbits/sec and mean value 15.7Mbits/sec.

To date, we have three distinct parsing approaches, DOM, SAX and StAX, to access, and might be modified, varied elements and attributes specified in an XML document (10). Another approach different from XML parsers is object binding. This is a special approach that directly processing XML content within object-oriented Java applications. It requires a Java API called XML Binding (JAXB) to transparently exchange data between XML content and Java objects.

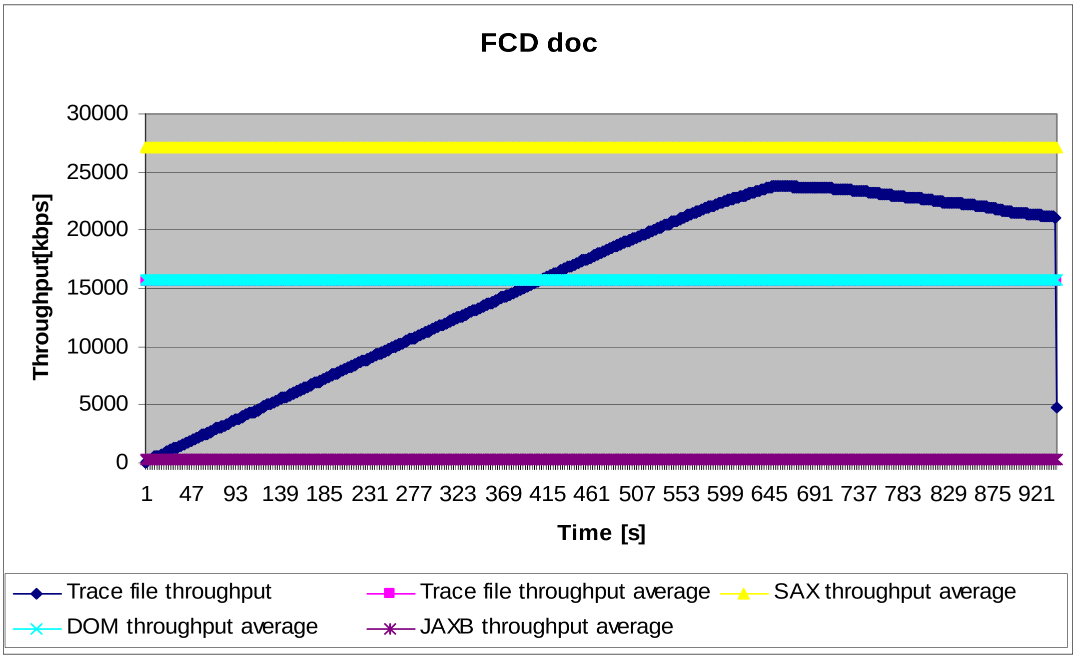

Fig. 6 Processing Capability Comparison of Different XML Processing Technologies

Fig. 6 presents the processing capabilities of SAX, DOM and JAXB. The data traffic throughput and its average value calculated based on the vehicle trace file are shown in the deep blue line and the pink line respectively. The other three lines represent the average processing throughputs of SAX, DOM and JAXB. The hardware and software environments for each measurement are identical. And in order to avoid unstable states, the test system has a 180 seconds ramp up time and a 30 seconds ramp down time. In each test, we calculate transactions for 300 seconds. One transaction is the parsing process of a XML formatted FCD application message. Thus processing throughputs can be calculated.

As it is shown, the yellow line SAX parser has the best performance of these three, since it is a streaming push parser which only scans through the whole XML document once. This also means no modifications are allowed in SAX parser. However, the designed TIC server does not only access XML document content but also need to modify the content fields and add the server information to the received messages. Both DOM and JAXB technologies provide the possibility to modify the XML formatted messages. But their shortcomings are obvious: processing throughputs are much lower than SAX parser.

CONCLUSIONS AND OUTLOOK

This article introduced a new CoCar-specific application protocol which can be used as an extension of the standardized TPEG protocol. The proposed protocol design uses existing and widespread standards like the Internet protocol stack and TPEG in order to guarantee interoperability and fast market introduction. Thus, the designed protocols are able to use existing systems and data sources and couple them to novel cellular communication technologies like the MBMS.

For evaluating the performance of the designed protocol stack, we implemented a trial traffic information center and adaptation modules for providing CoCar application services using a UMTS network. The implemented system can be used as a platform for the CoCar specific applications and provide a test environment for optimization of designed protocols. The performance investigation shows that XML processing is the major performance bottleneck at the test server. As the basic software-only solutions can not fulfill the growing demands of the CoCar services, XML appliances that can accelerate XML processing, security and routing are being considered and XML appliances from different vendors investigated.

ACKNOWLEDGEMENT

This work is carried out in the research Project CoCar (Cooperative Cars) which has been partly funded by the German Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung, BMBF) under the contracts 01BU0690 to -94. The authors are responsible for the content of this publication.

REFERENCES

(1) Global System for Telematics, 11th of January, 2008. [Online]. Available: http://www.gstforum.org

(2) PReVENT, 11th of January, 2008. [Online]. Available: http://www.prevent-ip.org

(3) Communications for eSafety, 11th of January, 2008. [Online]. Available: http://www.comesafety.org

(4) Cooperative Vehicles and Road Infrastructure for Road Safety, 11th of January, 2008. [Online]. Available: http://www.safespot-eu.org

(5) tpegML data model, 10th of January, 2008. [Online]. Available: http://www.itsregistry.org.uk/index.html

(6) ISO, “Traffic and Travel Information (TTI) - TTI via Transport Protocol Experts Group (TPEG) data-streams - Part 3: Service and Network Information (SNI) application,” ISO/TS 18234-3, June 2006.

(7) ISO, “Traffic and Travel Information (TTI) - TTI via Transport Protocol Experts Group (TPEG) data-streams - Part 4: Road Traffic Message (RTM) application,” ISO/TS 18234-4, June 2006.

(8) ISO, “Traffic and Travel Information (TTI) - TTI via Transport Protocol Experts Group (TPEG) data-streams - Part 5: Public Transport Information (PTI) application,” ISO/TS 18234-5, June 2006.

(9) Sommer, C.; Yao, Z.; German, R.; Dressler, F.: On the Need for Bidirectional Coupling of Road Traffic Microsimulation and Network Simulation. In: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (ACM Mobihoc 2008): 1st ACM International Workshop on Mobility Models for Networking Research (MobilityModels'08), Hong Kong, China, 2008

(10) Ajay Vohra and Deepak Vohra, “Pro XML Development with Java Technology”, Apress, 2006