Vehicle-to-Infrastructure Messaging Proposal Based on CAM/DENM Specifications

José Santa, Fernando Pereñíguez, Antonio Moragón, Antonio F. Skarmeta
Computer Science Faculty, University of Murcia, Murcia, 30100 Spain
Email: (josesanta|pereniguez|amoragon|skarmeta)@um.es

***Abstract*—The *European Telecommunications Standards Institute* (ETSI) has proposed a middleware solution (called *facility*) to support vehicular safety and traffic efficiency services needing continuous status information about surrounding vehicles or wanting to send asynchronous warning notifications to vehicles. The former capability is offered by the Cooperative Awareness Basic Service, while the latter is provided by the Decentralized Environmental Notification Basic Service. Reference packet formats for both the *Cooperative Awareness Message* (CAM) and *Decentralized Environmental Notification Message* (DENM) have been specified by ETSI, together with the general message dissemination guidelines. This paper presents an implementation prototype that has been developed to validate both CAM and DENM messaging capabilities. We provide implementation decisions and demonstrate the usefulness of these facilities through two reference services which provide a way to third-party services to gather tracking and tracing information, and send asynchronous road events to drivers within a specific area.**

***Index Terms*—CAM; DENM; message dissemination; V2I; vehicular networks; Intelligent Transportation Systems.**

I. Introduction

Cooperative services in vehicular scenarios are becoming essential for the future connected vehicle within the ITS (Intelligent Transportation Systems) research field. They are supposed to decrease road fatalities, improve the capacity of roads, diminish the carbon footprint of road transport and enhance the user experience during travels. Although there are many vehicular services envisioned for the short, medium and long term, these can be categorized in the next groups [1], [2]: *safety*, *traffic efficiency* and *infotainment*. Although many of these services have hitherto been proposed together with application-level protocols designed from scratch, this is not a scalable way of developing services within the same information system. With the exception of some services, such as multimedia or common Web access, many ITS services have common communication requirements:

- **Periodic status exchange**. ITS services typically need to know about the status of vehicle or roadside terminals. This implies the periodic exchange of data packets with information about location, speed, identifier, etc.
- **Asynchronous notifications**. This kind of messages are used to inform about a specific service event. In contrast to the previous status messages, the reliable delivery of these messages to a single terminal or a group of them is usually a key requirement.

Examples of the usage of the first communication type can be found on traffic efficiency services such as remote vehicle monitoring, which gathers periodic status data from vehicles, or safety services such as cooperative collision avoidance, which requires kinematic information about surrounding vehicles to detect potential impacts. Asynchronous notifications are mainly found in safety services, such as slippery pavement or post-collision warning. Hence, due to the proliferation of these two communication strategies, within the ISO/ETSI reference ITS communication architecture [3]–[5], the ETSI has developed two basic messaging services included in the reference stack as facilities available to the ITS applications. These are the Cooperative Awareness Basic Service [6], defining the Cooperative Awareness Message (CAM), and the Decentralized Environmental Notification Basic Service [7], which specifies the Decentralized Environmental Notification Message (DENM).

CAMs are a kind of heartbeat messages periodically broadcasted by each vehicle to its neighbors to provide information of presence, position, temperature, and basic status. On the contrary, DENMs are event-triggered messages broadcasted to alert road users of a hazardous event. Both CAM and DENM messages are delivered to vehicles in a particular geographic region: to the immediate neighborhood in case of CAMs (single hop), and to the area affected by the event for DENMs (multi-hop). The conditions under which CAMs and DENMs are transmitted and the message format is described in the corresponding international standards, although there are implementation decisions that are left for the developer.

CAM and DENM implementations on reference communication stacks are limited, and the same happens with the support on commercial products. Due to this, its usage on real services is an issue not sufficiently treated in the literature neither. In this paper, a reference communication stack that follows the ETSI/ISO reference ITS communication architecture guidelines is used as the basis for developing an implementation prototype of both CAM and DEMN facilities. Its usage is validated through two example services providing vehicle tracking/tracing and road warning functionalities.

The paper is organized as follows. First, Section II places the work in the research literature. Next, Section III presents the reference network architecture and summarizes the communication stack used, while Section IV details the implemented CAM and DENM facilities. Section V describes the two services developed to test these functionalities and, finally, the paper is concluded in Section VI.

II. State of the Art

Despite CAM and DENM constitute the basic set of messages established by the standardization forum to assist the operation of ITS applications, it is difficult to find research works contemplating the usage of CAM/DENM facilities. For example, there are a number of works where the information contained in CAMs and DENMs is used to develop a new geographic routing protocol [8] or a local dynamic map [9] to store information collected by vehicles. Another work acknowledging the importance of CAM/DENM messages can be found in [10] which presents an ITS communication infrastructure based on LTE cellular technology. Similarly, authors in [11] demonstrate the usefulness of CAM and DENM through a testing platform used to evaluate relevant safety applications. Unlike all the previous works, this paper is especially oriented to detail a standard-compliant implementation of the CAM/DENM facilities in base to the packet structure and the transmission algorithms used. This contribution will serve as a reference for future works considering standardized messaging in vehicular networks.

Regarding the reference services presented in this paper, in the literature we can find numerous works that have also attempted to develop a vehicle tracking service. Nevertheless, unlike our solution, none of these works take advantage of the CAM and DENM communication capabilities. For example, some solutions [12], [13] propose a vehicle tracking system based on video cameras deployed along the road. These works typically develop an algorithm to identity vehicles and infer valuable information such as speed or trajectory. Alternatively, other solutions [14] propose a vehicle tracking systems based on smartphone GPS positioning.

Similarly, the achievement of an efficient message dissemination in vehicular networks has been an issue extensively investigated. In fact, we can find a vast amount of work in this area [15], [16]. All these investigations have in common the objective of achieving an optimized message propagation among vehicles (i.e. in the vehicular ad-hoc network) by relying on different strategies. Unfortunately, these works neither contemplate the usage of standardized messages nor give the infrastructure the opportunity of generating messages to be disseminated in a certain road segment (i.e. it is assumed that alert messages are generated by vehicles and propagated in a V2V fashion). As explained later on, we propose a message dissemination service where road warnings are generated by an infrastructure party, like the road operator.

III. System Architecture

The base vehicular architecture used here for developing the CAM and DENM messaging was initially presented in [17]. This section briefly presents the communication stacks of the different entities and presents the new modules integrated in the architecture for supporting the CAM and DENM messaging capabilities.

A. Base Communication Architecture

The design showed in Fig. 1 shows the instantiation of our communication stack following the ISO/ETSI reference ITS communication architecture [4], [5]. In the diagram we can distinguish three main ITS station (ITS-S) nodes: Vehicle ITS-S, integrating the on-board networked nodes for accessing the whole network; Roadside ITS-S, which provides local wireless connectivity and data processing; and Central ITS-S, which includes the necessary nodes for providing infrastructure services. As observed, the communication stacks follow the ISO/ETSI six-layer scheme, with considers, from bottom to top, access technologies, networking and transport, facilities and, finally, applications; and, additionally, management and security on the left and right planes, respectively.

In the vehicle, the whole stack functionality is split in two nodes: vehicle ITS-S host and vehicle ITS-S router (also known as mobile router or MR). The mobile router includes the needed functionalities to hide networking tasks to in-vehicle hosts. An unlimited number of hosts could connect with the ITS network by means of the access router through a common WiFi (e.g. 802.11a/b/g) or an Ethernet connection. To maintain external communication with roadside equipment and the control centre, 3G/UMTS, WiMAX, WiFi and 802.11p (ETSI G5) are currently supported. IPv6 connectivity is supported by the rest of elements included within the networking and transport layer of the mobile router, provided with Network Mobility (NEMO) [18], which is in charge of maintaining reachability for the whole in-vehicle IPv6 network. The stack on the Vehicle ITS-S Host is in charge of executing final applications that could access remote services. A GPS device in the lower layer enables the host to be geolocated. As observed, this stack includes a common networking middleware based on the Transport Control Protocol (TCP) and User Datagram Protocol (UDP). An essential part of the host protocol stack is the facilities layer. As can be seen in Fig. 1, a Java Virtual Machine (Java VM) is used as the basis for the Open Service Gateway Initiative (OSGi) framework. OSGi acts as the manager of the lifecycle of middleware parts and applications. Above OSGi, a set of facilities operating as middleware are provided to make easier the development of new applications.

The communication stack instantiated in the roadside ITS-S access router acts as network attachment point for vehicles using short/medium-range communication technologies. Similarly to the vehicle ITS-S Mobile Router, the available wireless technologies to communicate with vehicles are WiFi (802.11 a/b/g), 802.11p (ETSI G5) and WiMAX.

Finally, in the upper part of Fig. 1 we can seen the communication stacks for both the Central ITS-S Application Server (ITS-S AS) and the Central ITS-S Home Agent (Central ITS-S HA). The former hosts the services offered to vehicles and third party infrastructure entities, while HA is used for maintaining the connectivity of vehicles upon the change of point of attachment, acting as NEMO HA. In this sense, the modules included in the HA network layer are equivalent to

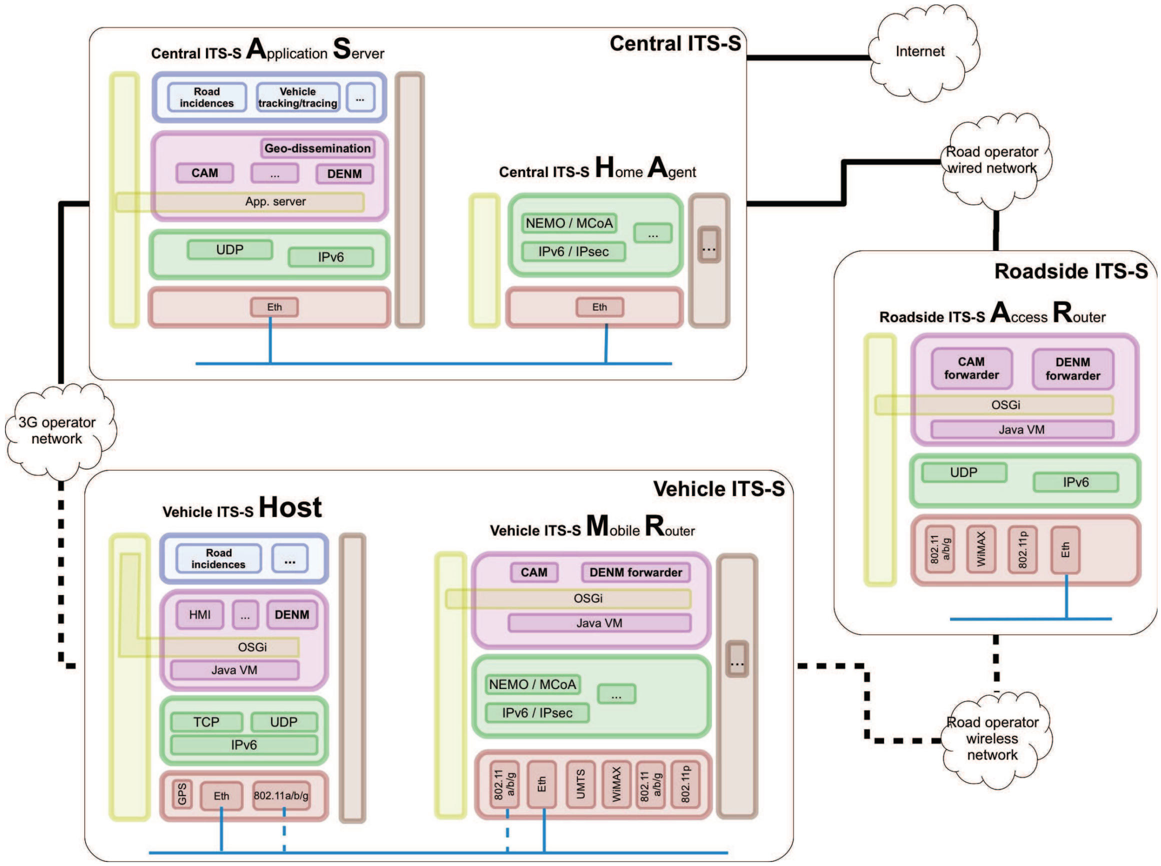

Figure 1. Overall architecture of the reference vehicular network

the ones included in MR.

B. CAM/DEMN Messaging Modules

A set of new modules have been integrated in the several communication stacks showed in Fig. 1 to support CAM and DENM messaging and provide two reference services using this capability.

In the vehicle side, the ITS-S MR has also been provided with a facility layer that operates over a Java basis to host OSGi-based facilities. As can be seen in Fig. 1, two particular facilities are included here for generating CAM messages and forwarding DENM messages from the infrastructure to Vehicle ITS-S Hosts. The host integrates a facility for decoding DENM messages containing asynchronous events, and a reference application that uses this information for showing traffic incidences on the road using a graphical interface is also included.

The ITS-S AR plays an important role, forwarding the CAM messages received from vehicles in the coverage area to the Central ITS-S, and doing the same action for the case of the DENM messages received from the Central ITS-S, which are broadcasted to all vehicles within the wireless range of the Roadside ITS-S by using IPv6 multicast. The new two modules that cover these capabilities are showed on the top of the facilities layer of the Roadside ITS-S AR. Although there are several possible communication technologies to be used in the communication between vehicles and Roadside ITS-Ss, for the case of the transmission of CAM and DENM packages, the 802.11p interface is found the most appropriate, since an attachment is not needed. When vehicles are driving at high speeds this aspect is of paramount importance to assure the maximum connection time.

In the Central ITS-S, the HA entity is not modified, since it works at network level. Nevertheless, as can be seen in Fig. 1, the Central ITS-S AS does include a facility to decode the CAM messages received from the vehicles, a facility to generate DENM messages, and an extra one for the geo-dissemination of the incidences using DENM notifications according to their location. This last module uses information about the Roadside ITS-S deployment to decide to which ones a DENM notification must be sent. In the upper layer of the Central ITS-S AS, one can see the central part of the two reference services implemented to notify vehicles about road incidences and maintain tracking and tracing information about vehicles.

IV. Message Dissemination Facilities

This section details the content of the CAM and DENM messages chosen in our architecture, together with the facilities developed and the algorithm used to generate them.

A. CAM Messaging

CAM messages are generated by the Vehicle ITS-S MR, forwarded by the Roadside ITS-S AR and, finally, received

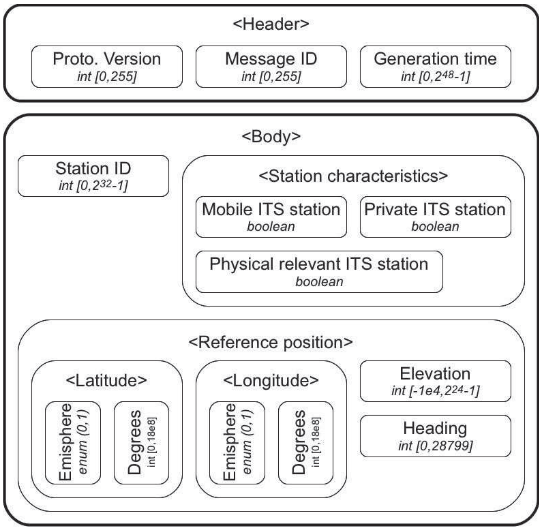

Figure 2. Structure of the CAM message

by the Central ITS-S AS, where ITS services are hosted. In this last entity the messages are decoded and the information contained is made available to the services. However, the most noteworthy module is the one provided in the MR, whose functionality is later covered in more detail.

First, the structure of the CAM message is presented in Fig. 2, which is compliant with the ETSI standard [6]. For the sake of clarity, extra optional fields have not been considered because they are not necessary for our envisaged services. Apart from common fields in the header, such as the protocol version, the message identifier (0 for CAM) or the generation time (in milliseconds since January 1st, 1970), the specific information in the message is contained in the body and the reference position parts. A different station identifier is chosen by each vehicular and roadside unit in our case, and the station characteristics are specified accordingly. Our stations are private, since we are not a public authority, and vehicle stations are physically relevant, since they are installed on moving vehicles. All reference position information is taken from GPS but the heading, which is computed by the CAM facility provided in the MR considering the last positions. In [6] there is an additional optional field for extra CAM parameters, but it has been obviated since its description in the current version of the standard is still incomplete. The ASNLab ASN.1 library for Java[1] has been used for generating both CAM and DENM packeting and parsing code.

As said before, the most important facility regarding the management of CAM messages is included in Vehicle ITS-S MR. Algorithm 1 details the operation of this module, interpreting the textual description given in [6]. As can be checked, the generation of a new CAM message depends either on the availability of new GPS information or the need to transmit a message at a minimum mandatory rate. In the first case, there are conditions that have to be satisfied so that a new CAM message is generated, taking as a reference the last

[1]http://www.asnlab.com

Algorithm 1: CAM message generation algorithm

Input: A new position read p, if any; a record of previous positions $pHist$; the last CAM message sent $lastCam$
Output: A new CAM message is sent and $lastCam$ is updated, if applicable; $pHist$ is updated with p, if applicable

```
while true do
    time = System.getTime()
    heading = calcHeading(pHist, p)
    lastPos = lastPosition(pHist)
    lastHist = pHist \ lastPos
    lastHead = calcHeading(lastHist, lastPos)
    speed = calcSpeed(pHist, p)
    if p ≠ null then
        lastSpeed = calcSpeed(lastHist, lastPos)
        if distance(p, lastCam.pos) ≥ D_THRESHOLD or
        |heading - lastCam.heading| ≥ H_THRESHOLD or
        |speed - lastCam.speed| ≥ S_THRESHOLD then
            cam = newCam(time, p, heading, speed)
            sendCam(cam)
            lastCam = cam
        pHist = pHist ∪ p
    else
        p = lastPos
        heading = lastHead;
    if time - lastCam.time ≥ T_THRESHOLD then
        cam = newCam(time, p, heading, speed)
        sendCam(cam)
        lastCam = cam
    System.wait(CHECK_PERIOD)
```

one sent: a distance covered threshold, an orientation change threshold, and a speed change threshold. This last value is not included at the moment in the encoded final CAM message to be sent, but is saved to make easier the algorithm. According to the standard, $D\_THRESHOD$ is 5 m, $H\_THRESHOLD$ is 4°, $S\_THRESHOLD$ is 1 m/s, and $T\_THRESHOLD$ is 1 s. Since the maximum CAM generation rate is 10 Hz, the value $CHECK\_PERIOD$ is said to be established with 100 ms in the standard, however our Java-based implementation needs some time to execute the previous code and we had to adjust this constant to 80 ms.

B. DENM Messaging

DENM messages are generated by Central ITS-S AS due to a request from one of the services requiring the notification of an asynchronous event. The Geo-dissemination module included in the AS is in charge of distributing DENM messages to Roadside ITS-S ARs within the event area, which finally transmit the messages to the vehicles in the surroundings.

Regarding the DENM structure, Fig. 3 shows the fields chosen. Again, this message if compliant with the standard [7]. The header is identical to the one included in CAMs, but the body contains more fields. First, the decentralized situation management group includes general information about the event: a pair of station identifier and message number for the univocal identification of the event, a version of the event, the expiration time, the sending frequency, the probability of the event to be true, and an additional negation field that allows another station to cancel the event. The second

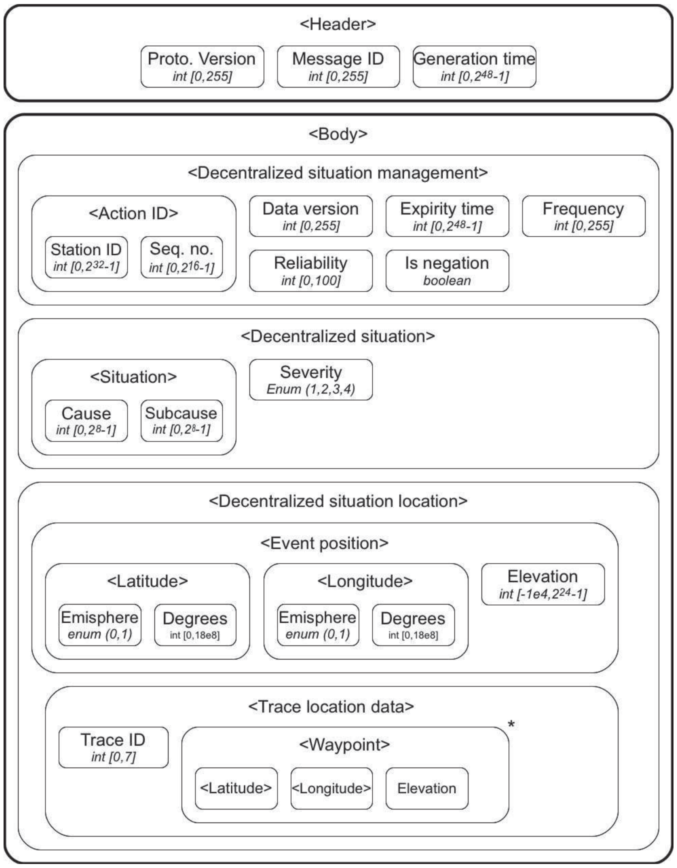

Figure 3. Structure of the DENM message

group includes specific details about the event informed in the message. The fields chosen are the situation cause and sub-cause, which detail the class of the event notified according to a predefined list in the standard, and the severity, which warns about the danger level using a set of four possible values ("informative", "obstacles", "danger", "highest danger"). The final group includes the event location data. The event position part includes the coordinates of the hazard in the same format as in the CAM case, and extra trace location data, which defines an itinerary approaching to the event position. This last information is given in the form of a set of waypoints that lead to the event position.

Unlike the CAM case, the distribution logic of DENM messages is divided between two entities: the Central ITS-S AS and the Roadside ITS-S AR. In the first node the message is created and distributed to the ARs within the hazard area. The procedure followed in this case is detailed in Algorithm 2. This function is called by a particular service wanting an event to be disseminated, and most of the data required to create a DENM message are provided. If a previous DENM message was generated for this event, the data version field should be updated accordingly. With this field it is also possible to indicate the termination of the event. Moreover, it is needed a mapping between the type of the event provided by a service and the required cause and sub-cause fields, and the same with the gravity of the event. Finally, it is needed to find the Roadside ITS-Ss under the coverage of the event. For this

Algorithm 2: DENM message generation algorithm

Input: A new event e is received from one of the services; a record of the previously generated DENMs $dHist$; a record of the previously generated event traces $tHist$

Output: A new DENM message is generated and sent to involved Roadside ITS-Ss, if applicable; $dHist$ is updated with a new DENM message, if applicable

```
sequenceNo = generateSequenceNo()
previousDenm = findDenm(dHist, e)
dataVersion = 1
if previousDenm ≠ null then
    if e.termination then
        dataVersion = EVENT_END
    else
        dataVersion = (previousDenm.dataVersion + 1) % 256
    delDenm(dHist, previousDenm)
mapSituationCause(e.type, ref cause, ref subcause)
mapSeverity(e.gravity, ref severity)
trace = findTrace(tHist, e.waypositions)
if trace == null then
    trace = newTrace(e.waypositions)
    tHist = tHist ∪ trace
denm = newDenm(INVALID_TIME, STATION_ID,
sequenceNo, dataVersion, e.expirityTime, e.frequency,
e.reliability, cause, subcause, severity, e.pos, trace)
rsus = findRsusInEvent(e.pos, e.dissZone)
sendDenm(denm, rsus)
if previousDenm == null then
    dHist = dHist ∪ denm
```

search, the event dissemination area provided by the service is considered, although this information is not included in the DENM message because the current version of the standard does not specify this field yet.

On the other hand, the Roadside ITS-S AR executes the code presented in Algorithm 3 as part of its DENM forwarder module. In this procedure AR continuously checks the frequency of active events before retransmitting a new DENM message within the coverage area. When a DENM is received from the Central ITS-S AS, it is added to list of active events, if it did not exist. Once the event has passed the expiration time or it includes a data version indicating the end of the event, the corresponding DENM message is removed from the list of active events. The constant $CHECK\_PERIOD$ is not specified in the standard, but in our case it is necessary to save computational resources. It is currently used with a value of 100 ms.

V. Reference Services

The CAM and DENM facilities provided along the different modules of the vehicular network previously described have been used for developing two reference services. A vehicle tracking and tracing service has been implemented to take advantage of the CAM messaging, while a road incidence notification service has been developed with the aim of using DENM messaging. This part of the article details these two reference implementations that validate the CAM and DENM messaging, considering the base vehicular testbed presented in [17].

Algorithm 3: DENM message retransmission algorithm

```
Input: A new DENM d is received from the Central ITS-S, if
       any; a record of previous DENMs dHist
Output: A DENM message is retransmitted, if applicable;
        dHist is updated with d, if applicable
while true do
    time = System.getTime()
    if d ≠ null then
        previousDenm = findDenm(dHist, d)
        if previousDenm ≠ null then
            delDenm(dHist, previousDenm)
        dHist = dHist ∪ d
    for each previousDenm ∈ dHist do
        if previousDenm.expirityTime < time and
        previousDenm.dataVersion ≠ EVENT_END then
            if previousDenm.time == INVALID_TIME or
            time - previousDenm.time ≥
            1/previousDenm.frequency then
                previousEvent.time = time
                sendDenm(previousDenm)
        else
            removeDenm(previousDenm, dHist)
    System.wait(CHECK_PERIOD)
```

A. Vehicle Tracking

The vehicle tracking service is mainly oriented to a back end system requiring tracking and tracing information of vehicles along the road infrastructure. Although it is identified as final application on the top of the Central ITS-S AS, this is also a service for local or third party infrastructure entities, such a logistic company, for instance. A web service interface is provided in the AS to allow the access of an external node requesting or subscribing for vehicle tracking and tracing information.

The basic operation of the service is as follows. The information gathered from the roadside through the reception of the CAM messages by the Roadside ITS-S ARs is used to maintain a data base of the vehicles positions and status during a timeframe. This way, when current tracking information is required, or a tracing request is received asking for the previous positions of a vehicle, the service accesses to this record to give an answer. Moreover, a subscription model is supported to inform interesting parties about new tracking events.

Specifically, a set of functionalities are included in the vehicle tracking service implemented in the form of a web service whose WSDL (Web Service Description Language) specification is omitted due to space limitations. The technology used to generate its corresponding Java code has been Apache CXF[2]. These web services are:

- Continuous vehicle tracking. This feature allows a service to subscribe for receiving asynchronous notifications about the location of a specific vehicle.
- Entry/exit vehicle tracking. This allows a particular service to subscribe to a specific geo-fencing area so that it is informed when a vehicle enter of exit the zone.
- Vehicle within geographical area. Specifying an area, a service can gather information about all vehicle driving within the zone.

The vehicle tracking service has been used in the FOTsis project[3] in two services to be widely evaluated in real roads: Special Vehicle Tracking, to notify the infrastructure and surrounding vehicles about the path followed by a special vehicle (usually trucks transporting dangerous goods), and Advanced Enforcement, which continuously gather status information about vehicles to assure their compliance with applicable regulations.

B. Road incidences

Unlike the previous one, this service directly provides functionality to both the Central ITS-S and Vehicle ITS-S. This can be clearly seen in Fig. 1, where, apart from a software module included in the AS side, there is another one in the Vehicle ITS-S Host.

A service running in the Central ITS-S AS can use the road incidences service to distribute asynchronous events to vehicles driving within a specific geographic area. Upon the reception of the request, the road incidences service uses the Geo-dissemination facility to propagate the content of the notification using a DENM message, which is then distributed among the Roadside ITS-Ss within the radio of the event. In this case, the service offered is mainly in charge of interfacing with local or remote third party applications through a web service interface also developed with Apache CXF. This accepts event dissemination requests containing basic information of the event, such as the type (defined in a set of possible values), reference position of the event, action area, duration and notification frequency. As can be noted, these data are later mapped to standardized codes by the DENM message generation algorithm described above.

At the vehicle side, the application included in the Vehicle ITS-S Host receives DENM notifications and processes the events. A reference graphical front-end has been implemented here for Android mobile devices, as can be seen in Fig. 4. It informs the vehicle occupants about the incidences received over Google Maps. The screenshot provided shows the waypoints that lead to a roadworks event. This application was originally implemented for a joint demonstration between the FOTsis and ITSSv6[4] projects in the 2013 ITS World Congress in Vienna. Additionally, a more complete human-machine interface (HMI) is available for more powerful in-vehicle devices. This has been previously presented in [19], offering dynamic rendering of graphical interfaces, specific and configurable voice recognition commands and spoken alerts, and multimedia capabilities, among others.

The road incidences service has been used in the FOTsis project, within the Safety Incident Management service, which provides real time information to drivers about incidences on the road.

[2]http://cxf.apache.org

[3]http://www.fotsis.com

[4]http://www.itssv6.eu

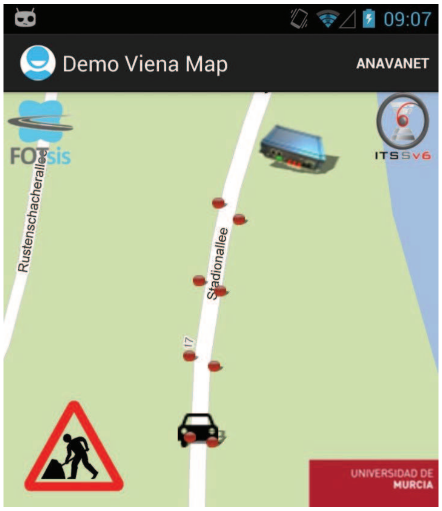

Figure 4. Screenshot of the in-vehicle smartphone application

VI. Conclusion

The contribution of the paper is based on a reference implementation of CAM and DEMN facilities inside a communication architecture that follows current ISO and ETSI guidelines. The packet structure and the algorithms used for the generation and transmission of messages are detailed, being placed as reusable middleware modules (facilities) in the Central ITS Station, Roadside ITS Station and Vehicle ITS Station. The interpretation given to the standard is of paramount importance in the literature, laying the foundations for future works in ITS communications. Moreover, two reference services that use this CAM/DENM basis have been implemented for offering tracking and tracing information about vehicles, and disseminate asynchronous events within a geographical area. These two services have been designed to also allow its usage by other local or third party services. This has been accomplished by using a service composition model based on web services calls.

The current ongoing work is mainly based on the evaluation of the CAM and DENM messages attending two perspectives: first, the evaluation of the standard on real traffic scenarios with the reference services described, and, second, the performance analysis of the messaging subsystem when using 802.11p communications. In the medium term it is also envisaged to substitute the geo-dissemination mechanism implemented by another one based on geographical IPv6 multicast.

Acknowledgment

This work has been mainly sponsored by the Ministry of Science and Innovation, through the Walkie-Talkie project (TIN2011-27543-C03), and the European Seventh Framework Program, through the ITSSv6 (contract 270519) and FOTsis (contract 270447) projects.

References

[1] Y. Khaled, M. Tsukada, J. Santa, J. Choi, and T. Ernst, "A usage oriented analysis of vehicular networks: from technologies to applications," *Journal of Communications*, vol. 4, no. 5, 2009.

[2] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions," ETSI TR 102 638, European Telecommunications Standards Institute, June 2009.

[3] T. Kosch, I. Kulp, M. Bechler, M. Strassberger, B. Weyl, and R. Lasowski, "Communication architecture for cooperative systems in europe," *IEEE Communications Magazine*, vol. 47, no. 5, pp. 116 –125, may 2009.

[4] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Communications Architecture," ETSI EN 302 665, European Telecommunications Standards Institute, September 2010.

[5] International Organization for Standardization, "Intelligent transport systems - Communications Access for Land Mobiles (CALM) - Architecture," ISO 21217, International Organization for Standardization, april 2013.

[6] European Telecommunications Standards Institute, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service," ETSI TS 102 637-2, European Telecommunications Standards Institute, March 2011.

[7] European Telecommunications Standards Institute , "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service," ETSI TS 102 637-3, European Telecommunications Standards Institute, September 2010.

[8] M. Aguilera Leal, M. Rockl, B. Kloiber, F. de Ponte-Muller, and T. Strang, "Information-centric opportunistic data dissemination in vehicular ad hoc networks," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, 2010, pp. 1072–1078.

[9] H. Menouar, F. Filali, and A. Abu-Dayya, "Efficient and unique identifier for v2x events aggregation in the local dynamic map," in *ITS Telecommunications (ITST), 2011 11th International Conference on*, 2011, pp. 369–374.

[10] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "Lte for vehicular networking: a survey," *Communications Magazine, IEEE*, vol. 51, no. 5, pp. –, 2013.

[11] R. Nair, B. Soh, N. Chilamkurti, and J. J. J. H. Park, "Structure-free message aggregation and routing in traffic information system (smart)," *Journal of Network and Computer Applications*, vol. 36, no. 3, pp. 974 – 980, 2013.

[12] W. Lu, S. Wang, and X. Ding, "Vehicle detection and tracking in relatively crowded conditions," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 4136–4141.

[13] S. Sivaraman and M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 267–276, 2010.

[14] T. Menard, J. Miller, M. Nowak, and D. Norris, "Comparing the gps capabilities of the samsung galaxy s, motorola droid x, and the apple iphone for vehicle tracking using freesim_mobile," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, 2011, pp. 985–990.

[15] R. S. Schwartz, R. R. R. Barbosa, N. Meratnia, G. J. Heijenk, and H. Scholten, "A directional data dissemination protocol for vehicular environments." *Computer Communications*, vol. 34, no. 17, pp. 2057–2071, 2011.

[16] M. Fogue, P. Garrido, F. J. Martinez, J.-C. Cano, C. T. Calafate, and P. Manzoni, "Evaluating the impact of a novel message dissemination scheme for vehicular networks using real maps," *Transportation Research Part C: Emerging Technologies*, vol. 25, no. 0, pp. 61 – 80, 2012.

[17] J. Santa, F. Pereniguez-Garcia, F. Bernal, P. Fernandez, R. Marin-Lopez, and A. Skarmeta, "A framework for supporting network continuity in vehicular ipv6 communications," *IEEE Intelligent Transportation Systems Magazine*, in press.

[18] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) Basic Support Protocol," RFC 3963 (Proposed Standard), Jan. 2005.

[19] J. Santa, F. Pereniguez, A. Moragon, P. Fernandez, F. Bernal, , and A. Skarmeta, "Ipv6 communication stack for deploying cooperative vehicular services," *International Journal of ITS Research*, in press.