Efficient Coordination and Transmission of Data for Cooperative Vehicular Safety Applications

C.L. Robinson
Coordinated Science Lab
University of Illinois
Urbana Champaign, IL, USA.
clrobnsn@uiuc.edu

L. Caminiti, D. Caveney, K. Laberteaux
Toyota Technical Center,
2350 Green Road, Ann Arbor, MI, USA.
(lorenzo.caminiti, derek.caveney,
ken.laberteaux) @tema.toyota.com

ABSTRACT

This paper presents a method for efficient exchanges of Data Elements between vehicles running multiple safety applications. To date, significant efforts have been made in designing lower-layer communication protocols for VANET. Also, industry and government agencies have made progress in identifying and implementing certain vehicular safety applications. However, the specific environment of VANET-enabled safety applications lends itself to significant efficiencies in how information is coordinated within a vehicle and transmitted to neighboring vehicles. These efficiencies are instantiated in what we call the *Message Dispatcher*. The Message Dispatcher is an interface between multiple safety applications and the lower-layer communication stack. This Message Dispatcher concept was recently contributed to the Society of Automotive Engineers (SAE) and has become an underlying principle in their safety message standardization process. It has also been implemented in vehicle demonstrations at the Toyota Technical Center (TTC) in Ann Arbor, MI.

Categories and Subject Descriptors

D.2.11 [**Software Engineering**]: Software Architectures—*Data abstraction, Domain-specific architectures, Patterns.*; C.2.1 [**Computer-communication Networks**]: Network Architecture and Design—*Wireless communication*; C.2.6 [**Computer-communication Networks**]: Internetworking—*Standards*

General Terms

Design, Standardization, Performance, Management.

Keywords

DSRC, Wireless, Vehicle Safety Applications, Message Dispatcher, Message Composition, Channel Loading.

This material is based upon work partially supported by a Toyota Contract at the University of Illinois.

VANET'06, September 29, 2006, Los Angeles, California, USA.
Copyright 2006 ACM 1-59593-540-1/06/0009 ...$5.00.

1. INTRODUCTION

This paper describes a method for creating message packets for wireless vehicle-to-vehicle (v2v) and vehicle-to- infrastructure (v2i) communications, which is envisioned to enable a new generation of cooperative vehicular safety applications [1, 2]. It addresses several real world implementation issues, restrictions and objectives and describes methods to meet these goals.

On an average day in the United States, vehicular collisions kill 116 and injure 7900. More health care dollars are consumed in the United States treating crash victims than any other cause of illness or injury [3, 4, 5]; the situation in the European Union is similar, with over 100 deaths and 4600 injuries daily, and the annual cost of € 160 billion [6]. Governments and automotive companies are responding by making the reduction of vehicular fatalities a top priority [4, 7].

Dedicated Short Range Communications (DSRC) [8, 9, 10] is the leading wireless technology under consideration for vehicular safety applications [1]. Significant progress has been made in standardizing the lower layer protocols for DSRC [8, 10]. Safety applications and related technologies continue to be examined by industry/government consortiums, such as the Crash Avoidance Metrics Partnership (CAMP[1]), the Car2Car Communications Consortium[2], the Advanced Safety Vehicle (ASV) Project[3], and others.

The task of specifying safety message composition and creation, the area of this paper, belongs to the Society of Automotive Engineers (SAE). Specifically, the draft[4] standard SAE J2735, Dedicated Short Range Communication (DSRC) Message Set Dictionary [11] defines several static message definitions, as well as a method to efficiently and dynamically create messages. This flexible method, called the *Message Dispatcher* (MD), was contributed to SAE by the authors of this paper, and is fully described below.

As described in Section 2, the automotive applications that have the greatest safety potential rely heavily on single-hop broadcast communication with nearby vehicles and infras-

[1]CAMP comprises the following vehicle companies: BMW, DaimlerChysler, Ford, GM, Honda, Nissan, Toyota, and Volkswagen. It works in partnership with NHTSA. Several of CAMP's project reports can be found at http: //www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html
[2]http://www.car-2-car.org/
[3]The ASV is a partnership by Japans 14 automobile, truck, and motorcycle manufacturers, sponsored by Japanese Ministry of Land, Infrastructure and Transport (http://www.mlit.go.jp/).
[4]SAE J2735 is expected to be formally ratified in July 2006.

tructure. Further, it appears likely that implementation of vehicular safety applications will be executed quasi-autonomously by each manufacturer. The eventual Safety VANET (SVANET) environment is expected to have devices from various manufacturers implementing distinct (although potentially cooperative) applications. Moreover, many vehicles are likely to run multiple safety applications concurrently (e.g. an emergency brake light application, a lane change warning application, an intersection collision warning application, etc.). Each application is likely to have different, although overlapping, Data Element requirements (i.e. many safety applications may require the vehicle speed, vehicle location, current turning radius, etc.).

The expected SVANET environment, with semi-autonomous devices running multiple applications, each with overlapping Data Element requirements, has the potential to generate packets with redundant data. For example, it is not efficient for several applications within a single vehicle to separately (and redundantly) send the vehicles current speed. Also, information that changes slowly or infrequently (e.g. windscreen wiper status) does not need to be sent frequently.

The Message Dispatcher (MD) addresses this Data Element coordination issue. Briefly, the MD sits between the safety applications and the lower-layer protocols to coordinate the data requirements of each application with the goal of reducing the redundancy of the broadcast data. This has the desirable effect of reducing channel utilization, thereby potentially mitigating channel congestion and its corresponding data loss or delay.

The message dispatcher has been successfully implemented into cooperative vehicle safety demonstrations at the Toyota Technical Center (TTC), confirming its feasibility and flexibility.

The remainder of this paper is organized as follows. In Section 2, the unique characteristics of Safety VANET are described. This includes the current trends in safety applications and their Data Element requirements. The section also identifies several ideal features of a SVANET system. Section 3 outlines the Message Dispatcher architecture. Section 4 illustrates how the MD has been successfully implemented into cooperative vehicle safety demonstrations at the Toyota Technical Center (TTC). Here the MD has be used when two safety applications are running simultaneously. Section 5 evaluates the MD against the goals outlined in Section 2. Conclusions as well as potential extensions are made in Section 6.

2. PROBLEM FORMULATION

This section describes cooperative vehicular safety applications and their communication requirements. It further provides other attractive features of a data exchange system for vehicular safety.

2.1 Vehicular Safety Applications and their Required Data Elements

Significant efforts, involving the vehicle industry and government agencies mentioned in Section 1, have been made to identify which communication-enabled vehicular safety applications will provide the greatest benefits. The deliberations by the US National Highway Traffic Safety Administration (NHTSA), the US Department of Transportation (USDOT), and the Vehicle Safety Communications Consortium (VSCC) of CAMP have identified eight such applications[5] [1, 2]:

1. Traffic Signal Violation Warning
2. Curve Speed Warning
3. Emergency Electronic Brake Lights
4. Pre-Crash Warning
5. Cooperative Forward Collision Warning
6. Left Turn Assistant
7. Lane Change Warning
8. Stop Sign Movement Assistance.

The communication requirements of these eight applications are shown in Table 5. Note that communication frequency ranges from 1-50 Hz, the size of the packet ranges 200-500 bytes, and the maximum communication range spans from 50-300 meters. Further, high-level Data Element requirements are specified and several of these Data Elements (e.g. Position and Heading) are needed by multiple applications.

Responding to these identified applications, the SAE [11] defines over seventy vehicle Data Elements (e.g. heading, acceleration (with varying precision: 4bit, 8bit, 16 bit), headlight status and brake status). Of these Data Elements, thirty of the most commonly used elements are selected for a common message set, as listed in Table 6. However, very few of these applications have actually been implemented or fully developed. Hence, the exact usage characteristics of the safety messages were in flux at the time the safety messages were being defined and standardized.

Table 6 further shows our preliminary analysis of which elements will be used by each of the eight NHTSA safety applications [1]. These choices may be refined over time, but it is clear from Table 5 and Table 6 that several Data Elements will be useful to multiple applications - although perhaps at different frequencies and distances. It is further quite likely that some of the SAEs sixty-one Data Elements, such as *AirBagCount*, *AirTemperature*, and *WiperRate*, will be used far-less frequently. This leads to the conclusion that a message with fixed contents will either be very large or not be able to meet all application requirements.

2.2 Broadcast Characteristics

As illustrated in Table 5, safety messages tend to be locally broadcast with a maximum transmission range of 300 meters. This is motivated by the fact that messages sent by one vehicle will contain Data Elements that are often useful to multiple vehicles in the nearby vicinity. It often matters less "who" is receiving the data, but more "where" the recipients are in relation to the sender. As DSRC radios are required to communicate at least 300 meters, we assume that safety messages broadcast their messages in a single hop[6].

Further, it is likely that the nearby neighbors of a vehicle will change frequently. It is likely unnecessary, and difficult, to maintain an updated topology of 1-3 hop neighbors. Again, 1-hop broadcasts seem appropriate.

[5] The authors are aware of similar deliberations in Europe and Asia, with similar results.

[6] Extensions, such as dynamic power control [12] and geographical flooding [13], are also possible in Safety VANET. The Message Dispatcher concept readily extends to these situations.

2.3 Desirable Architectural Features

In addition to simply providing information useful for the defined applications, there are several goals which an architecture should satisfy. These goals are driven both from a technical standpoint as well as an business perspective in that real world deployment and proliferation considerations need to be made.

Future Proof: Vehicles will broadcast data that is likely valuable for multiple surrounding vehicles, as well as multiple safety applications in each of these vehicles. However, despite the significant efforts thus far to reach consensus, creating and testing between manufactures of many of these safety applications is an ongoing effort. As such, a scheme that is future-proof to newly-defined, evolving or upgraded applications is essential.

Flexibility: It seems likely that in the heterogeneous marketplace for vehicles, different vehicles on the same road will be running different subsets of safety applications. An ideal scheme should be sufficiently flexible to account for this, as well as several other requirements mentioned elsewhere in this section.

Extensible: It is conceivable that not all safety applications will be universally standardized. Hence, a mechanism for adding support for non-standardized Data Elements (e.g. proprietary to one or more manufacturers) would be desirable. This would ensure that applications would not be restricted by constraining the information they are able to communicate.

Single Interface: From an implementation perspective, it is attractive to construct an architecture where policy and self-policing between various applications within a single vehicle can be managed in a single entity. Further, authentication and other security primitives would ideally be managed collectively across safety applications.

Separation of Concern: By providing a layered architecture that abstracts the message sending interface from the application designer, a separation of concern for the application designer is achieved. Thus enabling easier, more modular, development.

Low Bandwidth usage: The available bandwidth is a finite resource and should be conserved where possible. Real-world testing by the VSCC [2] demonstrates that the channel capacity is an issue that will need to addressed for large-scale deployment and in stressed traffic environments.

Information Rate: Some Data Elements such as headlight status change infrequently. Thus, an architecture should be able to distinguish these properties and transmit information when it is appropriate. This notion can be parametrized as decreasing the $b_E = \frac{\text{Bit Rate}}{\text{Entropy Rate}}$ ratio.

Recognize vehicle capabilities: Not all vehicles will be able (or willing) to measure and transmit certain pieces of information and this should be reflected in the message construction.

Enable Product Differentiation: Vehicle manufacturers desire the ability to provide unique applications and services to their customers. The functionality of these services should not be limited to the applications that are currently deployed or enabled by other vendors.

As described in the following sections, the Message Dispatcher architecture provides a sound and efficient solution to the envisioned vehicular safety data exchange environment.

3. MESSAGE DISPATCHER

The basic architectural concept of the Message Dispatcher is illustrated in Figures 1 and 2. The Message Dispatcher's responsibility is to coordinate all the data exchange requirements of the applications running on a vehicle. The MD accomplishes this by serving as an interface between the applications and the communication stack.

Safety applications will send Data Elements to be broadcast to the MD. The MD will then summarize these Data Elements across applications and create a single packet comprising the minimum set of the Data Elements to be transmitted (See Figure 1). The MD would also consider data requirements of other surrounding vehicles or roadside units as described in Section 3.3. This packet is then sent to the DSRC radio for broadcast. Any vehicle that receives a message would provide all on-board applications with the Data Elements they require, as shown in Figure 2.

The Message Dispatcher design can be divided into two broad topics, first the definition of a Data Element Dictio-

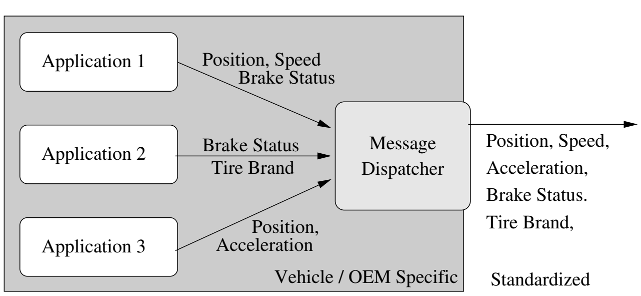


Figure 1: The Message Dispatcher is assimilates data requirements from all the on-board applications and compiling a single message using a dictionary of defined Data Elements and standardized message construction guidelines.

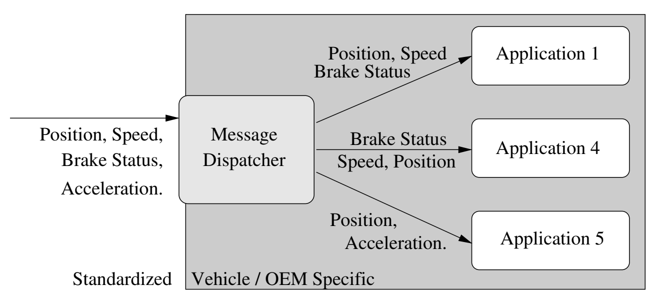


Figure 2: A receiving Message Dispatcher is responsible for separating and disseminating Data Elements from the received message to all on-board applications as well as managing data requirements for surrounding vehicles.

nary (Section 3.1), and second the specification of how these elements should be combined into a message (Section 3.2).

3.1 Data Element Dictionary

The MD's primary function is coordinating Data Elements for the vehicle's safety applications. This section describes how Data Elements are identified and formatted. It further describes our proposal for adding new elements to the SAE Data Element dictionary.

The SAE standard [11] identifies over 70 such elements in its Data Element dictionary. Each element in the Data Element dictionary is defined using:

- A Standard Name
- A Unique Identifier
- A Unit of Measure
- Accuracy of measure
- Range of measure
- Size (# of bytes)
- A description

As an example, consider this Data Element representation in Table 1.

Name	DE_VehicleLatitude
Unique ID	70
Unit	microdegrees
Accuracy	LSB is 1 microdegree
Range	-900000000 to 900000000
Size	32bits
Description	The latitude position of the center of the vehicle, expressed in micro degrees and based on the WGS-84 coordinate system.

Table 1: An example of the fields used to define Data Elements for vehicle latitude.

Using this data dictionary, a message can be constructed by creating a string of unique identifiers followed by the value of the Data Element. However, this overhead can be reduced when related Data Elements are grouped into a "Data Frame". For example, latitude is frequently updated and transmitted along with longitude. By formatting latitude and longitude into a single "position" Data Frame, overhead is reduced. Other Data Frames can be similarly constructed (e.g. a "brake" Data Frame comprising braking force, measured deceleration, and anti-lock brake status). Other groupings can readily be created by considering Table 6. Each Data Frame consists of Data Elements in a specified order and thus their unique ID's are not required within the Data Frame. Note that a "Data Frame" can comprise a single element, in which the "frame" id and the "element" id are equal. Several data frames have been defined in the SAE standard.

Adding or modifying a Data Element under this architecture is relatively straight-forward. The data dictionary would need to be updated and re-submitted to a central authority (currently the SAE) for updating the standard. While waiting for the standards body to act, new elements can be introduced by a light-weight tagging scheme, as mentioned in Section 3.2.

3.2 Message Construction

Each message is constructed using the Data Elements and Data Frames specified by the data dictionary. The message dispatcher can choose to include elements (either in a Frame or as an individual Data Element) so as to meet latency, network loading or application demands. In the SAE standard [11], the message has been divided into three sections. The first section is used to include Data Frames using their unique identifier followed by the series of Data Elements comprising the Data Frame. The second section is used to include individual Data Elements that have not already been included in the first section. Each Data Frame, whether it comprises several Data Elements or only one, will begin with its unique frame identifier. The third and final section is reserved for the inclusion of ad-hoc or newly defined terms by using a lightweight labeling scheme.

The lightweight nature of the schema is achieved by defining a single escape character that is used to indicate the start and end of a Data Element, thus enabling transmission of variable length data. The escape character will immediately be followed by a unique tag of fixed length that identifies the subsequent data. Note that this tag may not yet be defined in the Data Dictionary and hence the Message Dispatcher would poll the subscribed applications for knowledge of the incoming Data Element and forward appropriately. The specific method of how applications register their requirements with the MD are discussed in Section 4.4.

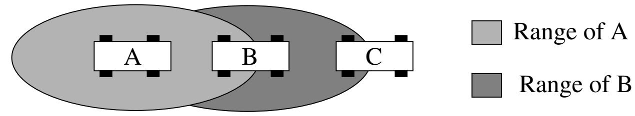


Figure 3: Vehicle A sends a message that the Message Dispatcher on vehicle B matches on subsequent transmissions. Although Vehicle C is beyond the range of interest of Vehicle A it too begins to match the message resulting in a "racing" condition.

3.3 What and When to Send

Determining what elements are required to be sent by a vehicle in order to satisfy surrounding vehicles requirements has not been completely addressed by either CAMP or SAE, and this is not an issue unique to the MD. No matter how Data Elements are communicated, there is still no specified way to request information from a newly encountered vehicle. One solution is to send a large packet, comprising all defined Data Elements, at the maximum required rate among all elements. However, this generates very large b_E ratios and potentially congests the wireless channel. While not the main contribution of this paper, we now propose two possible methods for solving the problem of requesting specific Data Elements that can easily be implemented using the MD architecture:

- *Match Received Message*: If a Message Dispatcher receives a message with a Data Element it is not currently transmitting, it should, if possible, match the contents of the incoming message on its next transmit. Further, in order to avoid a "racing" situation shown in Figure 3, the original sender should include a Data Element that indicates whether its message contents should be matched. Thus, in order for a vehicle to receive the information it requires, a single round-trip delay will be incurred.

- *Request Data Elements*: Send a Data Element that specifically requests certain elements. This can be achieved by simply defining another Data Element and including it in the data dictionary. This solution would be useful in probe type applications where a roadside unit would request information from passing vehicles.

4. EXAMPLE APPLICATIONS

This section illustrates how the MD concept can be leveraged to efficiently construct a message containing Data Elements for different safety applications. It further describes how the MD has been successfully implemented in several vehicles at the Toyota Technical Center (TTC), in Ann Arbor, MI, USA. The two implemented safety applications, the Emergency Brake Warning (EBW) application and the Intersection Violation Warning (IVW) application, are described in the following subsections:

4.1 Emergency Brake Warning

The Emergency Brake Warning (EBW) application alerts the driver when a preceding vehicle performs a severe braking maneuver, as shown in Figure 4 and 6. The application operates by augmenting the brake-light notification system (the rear tail lights in most vehicles) with a system that uses DSRC. When vehicle 1 brakes sharply, a message is transmitted indicating that it is undergoing emergency braking. Surrounding vehicles that receive the message must then discern if the event is relevant, since they can ignore warnings from vehicles traveling behind, far ahead, or in the opposite direction.

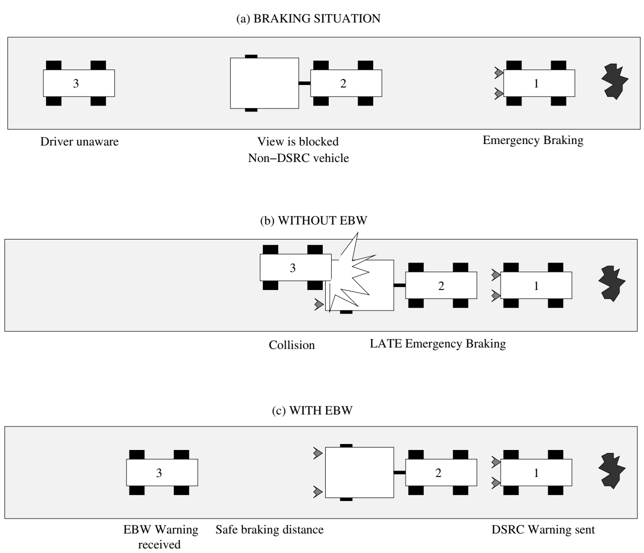


Figure 4: Brake lights are often difficult to see if there is a blocking vehicle. EBW provides a brake warning by using wireless communication to allow safe stopping.

Method of Operation: When there are no severe-braking events occurring, it is likely that all DSRC-equipped vehicles send out a Heart-Beat Message (HBM) comprising a minimal set of Data Elements and Frames as shown in Column 2 of Table 2. Consider Figure 4. When vehicle 1 performs severe-braking a request would be passed to the message dispatcher to begin transmitting the Data Elements shown in Column 3 in Table 2 at the indicated frequency. Of special interest is the data with ID *AH:EBWBreadcrumb* which represents a sampled-path history of the vehicle. This element has not been defined in the Data Dictionary and is appended to the message using the extensible schema described in Section 3.2. On reception, vehicle 3 can estimate if the braking car is in its forward path using the *AH:EBWBreadcrumb* data and can take appropriate action. Since the path history is only used when a severe-braking event occurs it would not be necessary to send until a severe-braking event occurs.

4.2 Intersection Violation Warning

The Intersection Violation Warning (IVW) application warns the driver if violating a red light seems imminent (See Figure 5). Other vehicles approaching the intersection are also warned that an approaching vehicle has issued a warning. The application sends different Data Elements depending on the driving situation but many are similar to the EBW application, although they may have different frequency requirements, as shown in column 4 in Table 2. In addition, some required data is unique to the IVW application and has not yet been defined in the Data Dictionary (e.g. *AH:IVWMap*).

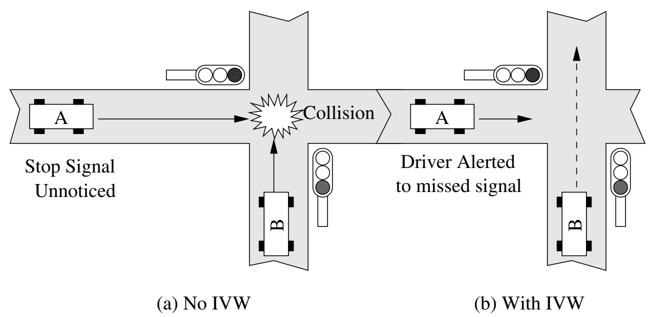


Figure 5: Without IVW vehicle A runs the light and causes a collision with Vehicle B. When IVW is activated in figure (b), both drivers are alerted allowing Vehicle A to stop and Vehicle B to proceed cautiously through the intersection

Method of Operation: When this application is running a roadside unit will transmit information regarding the traffic light including its location, light status, time till color change, dimensions of intersection (*AH:IVWMap*) etc. (See Table 5.) Vehicles running the application will register with the MD to receive all incoming IVW related Data Elements. Vehicles approaching the intersection compare this information with their projected trajectories and determine if a signal violation is imminent. If so, the driver of the potentially violating vehicle is alerted. In addition, a message is sent from the vehicle to the traffic light and surrounding vehicles indicating that a violation is likely. Thus, the message dispatcher on the vehicle will only send IVW data when triggered by a violation event - and will then send at the frequency illustrated in Column 4 in Table 2. In addition, the MD is responsible for routing incoming Data Elements from the traffic light to the IVW application.

Data Element (DE) / Frame (DF)	HBM	EBW	IVW	Message Dispatcher	3 Hz	5 Hz	10 Hz
DF: PositionShort	3Hz	5 Hz	10 Hz	In DE:PositionLong	•	•	•
DF: AccelerationSet4Way	3Hz	-	5 Hz	5 Hz	•	•	
DF: PositionLong	-	5 Hz	10 Hz	10 Hz	•	•	•
DF: PositionConfidenceSet	-	3 Hz	3 Hz	3 Hz	•		
DF: SpeedandHeadingPrecision	3Hz	5 Hz	5 Hz	5 Hz	•	•	
DE: Acceleration	-	3 Hz	-	In DF:AccelerationSet4Way			
DE: AntiLockBrakeStatus	-	3 Hz	-	3 Hz	•		
DE: BrakeAppliedStatus	3Hz	3 Hz	-	3 Hz	•		
AH: EBWBreadcrumb	-	5 Hz	-	5 Hz & to EBW	•	•	
DE: TrafficLightID	-	-	-	To IVW			
DF: TrafficLightLocation	-	-	-	To IVW			
DF: TrafficLightPhases	-	-	-	To IVW			
AH: IVWMap	-	-	-	To IVW			
DE: IVWWarningFlag	-	-	10 Hz	10 Hz	•	•	•
DE: IVWWarningID	-	-	10 Hz	10 Hz	•	•	•
DE: IVWWarningVehPos	-	-	10 Hz	In DF:PositionLong			
...	-	...	...	...	...	...	...

Table 2: A sample of the Data Elements (DE) and Frames (DF) that would be required by the two safety applications (Emergency Brake Warning (EBW) and Intersection Violation Warning (IVW)), as well as the Heart Beat Message (HBM). Column 5 represents the transmit requirements that the Message Dispatcher must satisfy when both applications are active as well as where incoming elements should be routed. Columns 6, 7 and 8 represent the contents of a 3, 5 and 10MHz message as described in Section 4.3. Note that *DF:PositionShort* is not included as it is a subset of the information in *DF:PositionLong*.

4.3 Message Composition

The situation we now pose is that a vehicle is speeding toward a red light. The driver has just been alerted to a potential violation and has begun to brake sharply. Thus, both the IVW and EBW systems are sending out messages from the vehicle. Table 2 lists a *subset* of Data Elements used by the two applications and their required update frequency. Column 5 indicates what requirements the Message Dispatcher must satisfy in its transmission of messages.

The Message Dispatcher combines the different Data Frames and Data Elements required into the minimum size message or set of messages. Duplicate Data Frames and Data Elements requested by both applications are not repeated (ex. *DF:PositionLong*). Further, if a Data Element requested by one application is contained within a Data Frame requested by another application, it is only included as part of the Data Frame. This is illustrated with the Data Element *DE:Acceleration* and *DE:IVWWarningVehPos* as well as *DE:PositionShort*in Table 2.

Since the applications have registered data at three different frequencies, the MD can construct and use three different messages, which we will call Msg10Hz, Msg5Hz and Msg3Hz. They will be transmitted alternatively so as to meet the overall data requirements, while minimizing bandwidth usage. It should be stressed the naming of these three messages is only for discussion purposes and should not be interpreted as message IDs, which are common in fixed message set compositions. In fact, the MD does not require any message ID, just Data Element (and Data Frame) identifiers. The contents of each of these messages is indicated in the final 3 columns of Table 2. Note that the Data Elements in the 10Hz message also appear in the 5Hz and 3Hz message. The MD will then send the messages in the sequence illustrated in Table 3 and satisfy the frequency requirements in Table 2. Since the MD only sends one message at a time and does not duplicate Data Elements in different messages the overall communication channel utilization, as well as b_E, is reduced.

Time (sec)	Message Sent
0.0	Msg3Hz
0.1	Msg10Hz
0.2	Msg5Hz (a)
0.3	Msg10Hz
0.4	Msg3Hz (b)
0.5	Msg10Hz
0.6	Msg5Hz
0.7	Msg10Hz
0.8	Msg5Hz
0.9	Msg3Hz

Table 3: Time line indicating the sending procedure for the Message Dispatcher to send the messages described in Table 2. Note (a) The Msg10Hz is not sent since it is contained into Msg5Hz. (b) The Msg5Hz and Msg10Hz are not sent since they are both contained into Msg3Hz.

4.4 Implementation

The Message Dispatcher has been implemented by the Toyota Technical Center in two Toyota Prius. Each vehicle is retrofitted with a Linux-based miniature PC, an OBD-II vehicle interface, a DENSO prototype DSRC radio, and a commercial DGPS unit (See Figure 6(f)).

The MD implementation conducted by TTC uses a callback mechanism to interface with the various applications. Upon initialization, each application registers with the MD those Data Elements it will **provide** to surrounding vehicles and infrastructure and those it will **receive** from the MD. Depending on the application, certain Data Elements may be both provided and received. During Data Element registration, the application will supply the MD with the callback method that should be invoked by the MD when it is time to send the Data Element across the communication channel or when it has received a new value for the Data Element over the communication channel.

At the end of the registration process, the application can specify the frequency at which certain Data Elements should be provided and inform the MD to begin a periodic transmission of these Data Elements. The Message Dispatcher will then be responsible for the message composition detailed in Section 4.3, using the provider callback method specified by the application to get the Data Element values. Conversely, the transmission of other registered Data Elements may be event-driven. In this case, when the event is triggered by the application logic, the application is responsible for invoking a "Send Now" API in the MD. This API will use provider callback methods immediately to get the necessary Data Element values to construct the message. With both periodic and event-driven communications possible, the MD includes an internal scheduler to decide when to send periodic Data Elements. When interrupted by an event-triggered transmission that included some periodic Data Elements, the MD scheduler will resynchronize future periodic transmission times to the time of the event, thus insuring that the average transmission frequency matches the period specified by the application.

The Toyota Technical Center has successfully implemented the MD described above in approximately 1000 lines of code. It has been extensively tested while running the EBW and IVW applications simultaneously. For these applications, the traffic light and two vehicles each run their own MD.

For the EBW application, both vehicles periodically provide the Heart-Beat Message (HBM) shown in Table 2. However, when an emergency braking event is identified, the braking vehicle will append the additional Data Elements shown in Column 3 of Table 2. Subsequently, both vehicles request to listen for the Data Elements of Column 3, and perform a relevancy check defined in its associated callback method only when *AH:EBWBreadcrumb* is heard.

For the IVW application, the traffic light periodically provides phase, timing, and map Data Elements to its MD, while each vehicle requests to listen for these Data Elements. When a vehicle's MD receives these Data Elements, it supplies them to the IVW application through the associated callback method. In this manner, the IVW application performs a relevancy check on the message from the traffic light immediately upon hearing from the MD that new information is available. In addition, the traffic light and both vehicles all request *IVWWarning* elements. The transmission of these elements from vehicles is event driven and triggered only when intersection violations are imminent. However, it is during these events that the MD combines information from both the EBW and IVW applications into the compact message as mentioned in Section 4.3.

4.5 Analysis

A crude evaluation of the performance of the MD for the two-application TTC implementation is now given. Using the Data Element sizes specified in SAE J2735 [11], the Heart-Beat Message is 25.5 bytes, Emergency Brake Warning message is 155.5 bytes, and the Intersection Violation Warning message is 46.75 bytes.These message sizes are calculated without including Data Frame headers, Data Element identifiers, or any tagging schema. A Common Message Set that incorporates all the Data Elements necessary for the HBM, EBW, and IVW is 176.75 bytes.

Assume this Common Message Set is periodically transmitted with the highest frequency in Table 2, 10Hz, in order to meet the requirements of the IVW application. For one vehicle, this requires a channel usage of 14.1kbps, as shown in Table 4. Conversely, a HBM being sent out by the MD at the 3Hz frequency specified in Table 2 requires only 0.6kbps. This is a reduction of 95% of channel load. The MD channel usage rises to 3.7kbps when IVW events occur (initiating a 10Hz transmission) and to 6.2kbps when EBW events (initiating a 5Hz transmission) occur. If both IVW and EBW events occur simultaneously on one vehicle, then the channel usage of the MD will be equal to that from the Common Message Set. However, this channel loading under the Message Dispatcher assumes that the elements for either implemented application are sent continuously. Since the MD can dynamically manage message contents depending on requirements, the full EBW/IVW message need only be sent in EBW/IVW instances which we assume to occur with the % frequencies shown in the first column of Table 4. Thus, the expected channel loading can be easily computed and is shown in column 3 to be far less than the peak channel loading.

Message Type & (Use Freq).	CMS	MD	
	Bandwidth		E[Bandwidth]
Heart Beat (100%)	14.1	0.6	0.6
EBW (2%)	14.1	6.2	0.71
IVW (4%)	14.1	3.7	0.72
EBW & IVW (3%)	14.1	14.1	1.01

Table 4: Comparison of bandwidth requirements (in kbps) under the Message Dispatcher (MD) and Common Message Set (CMS) architectures. The 4^{th} column represents the expected bandwidth usage assuming that the frequency of usage in the first column.

Further, it is significant to notice that additional saving will be achieved in overall bandwidth usage when there are multiple vehicles present. This is because when using the MD only a limited number of vehicles will need to transmit a EBW or IVW message, while the remaining vehicles continue to transmit the heart beat message.

Although these results depend on several simplifying assumptions it is clear that with a maximum DSRC channel capacity of 27Mbps the reduction of channel loading possible by employing the MD is relevant.

5. ARCHITECTURE ANALYSIS

In this section we describe how the goals identified in the Section 2.3 are met by MD architecture.

As shown in Figure 1, each application will register with the Message Dispatcher and stipulate its data requirements. Thus, the interface requirements for applications are specified in a single location, thus any upgrade or modification to the interface requirements are well localized (*single interface*). The Message Dispatcher is then responsible for generating a combination message that satisfies the requirements of applications that have registered. Applications do not need to consider how Data Elements are shared between vehicles, achieving an effective *separation of concern*. When the message format or protocol changes, only the MD implementation must change, and not the application. The separation of the application requirements from the channel and message construction yields an important abstraction for the both application and communication designers. Specifically, without modifying the applications, algorithms for avoiding channel overload, secure data transmission and low latency message delivery can be implemented in the Message Dispatcher enabling *lower bandwidth usage* as well as managing the $b_E = \frac{\text{Bit Rate}}{\text{Entropy Rate}}$ ratio as required. Lastly, the messages composed do not have to contain redundant information that is not available and thus can *recognize vehicle capabilities*.

The ability of the Message Dispatcher to evolve as application requirements change by using the flexible terminal character with post-fixed identifier scheme is of great benefit. Firstly, if a particular application requires a unique set of infrequently used Data Elements it is trivial to combine these into a message which is sent only when required and thus achieves *flexibility*.

The problem of when an application requires a Data Element that is not yet defined is described in Section 3.3 and illustrates the *extensibility* of the architecture. The MD can also transfer unique messages containing application specific data not defined in the standard and hence it effectively *exploits the communication ability*. Finally, since the applications have been successfully separated from the communication protocol, companies are able to develop their own products and *enable product differentiation*.

Finally, this architecture enables several other interesting considerations. For example, the MD could dynamically create messages based on channel loading conditions, vehicle traffic congestion conditions, Data Element or application priority, low latency requirements, as well as modulate transmission power based on Data Element requirements, etc.

6. CONCLUSION AND EXTENSIONS

This paper starts in Section 1 with a brief review of the current developments and motivations of industry and government agency efforts to leverage VANET to create a new class of vehicular safety applications. Specific applications and their data and architectural requirements were described in Sections 2. It was noted that many of the data requirements for various vehicular safety applications either partially or completely overlap. Consequently, in Section 3 the Message Dispatcher concept is introduced which leverages the Safety VANET environment and significantly improves efficiency of the wireless DSRC channel. It is then shown in Section 5 that this architecture meets several technical and business objectives and also enables other useful functionality. A specific implementation example from the Toyota Technical Center is described in Section 4.

In addition to being adopted by the relevant SAE standard, the Message Dispatcher has received wide appreciation among vehicle manufacturers. An upcoming three year project, funded by US NHTSA, will demonstrated the MD in a large-scale (approximately 100 car) demonstration project.

There are also several interesting research avenues opened and enabled by this architecture. For instance, channel loading can now explicitly be managed through dynamic message construction, packet collision avoidance algorithms can be implemented where particular Data Elements are retransmitted in the case of loss, MD's can cooperate to manage channel usage by considering application requirements as well as present collective data delivery information to applications, priority information such as low latency or application dependencies can be accounted for, filtering or other modifications to the raw incoming data can be performed and even multi-hop information passing schemes can be implemented.

7. ACKNOWLEDGMENTS

The authors wish to thank Professor P.R. Kumar and Dr. Girish Baliga at the Coordinated Science Lab at the University of Illinois at Urbana Champaign for fruitful discussions on the architecture, implementation, and standardization of the Message Dispatcher.

From the Toyota Technical Center, the authors wish to thank Mike Samples and Mike James for their influential discussions on architecture abstractions, and Hideki Hada, Jeff Rogers, and Hemant Kowshik for the contributions in TTC vehicle builds and demonstrations. Several other TTC staff made substantial efforts in procuring and managing assets need for our intelligent vehicle fleet.

The authors also wish to thank Tom Schaffnit for his support and guidance in bringing the Message Dispatcher to the SAE standards process.

8. REFERENCES

[1] A. Carter, "The status of vehicle-to-vehicle communication as a means of improving crash prevention performance," Tech. Rep. 05-0264, NHTSA, 2005. http://www-nrd.nhtsa.dot.gov/pdf/nrd-01/esv/esv19/05-0264-W.pdf.

[2] "Vehicle safety communications project-final report," Tech. Rep. HS 810 591, USDOT, April 2006. http://www-nrd.nhtsa.dot.gov/departments/nrd-12/pubs_rev.html.

[3] "Traffic safety facts," Tech. Rep. Report DOT HS 809 767, NHTSA, http://www-nrd.nhtsa.dot.gov, 2003.

[4] "U.S. Transp. Sec. Mineta announces opening of crash preventing intelligent intersection test facility." Press Release, 23 June 2003. http://www.its.dot.gov/press/fhw2003.htm.

[5] J. Paniati, "Intelligent safety efforts in America." Presentation, November 2003. http://www.its.dot.gov/speeches/madridvii2003.ppt.

[6] CARE, "Community road accident database," 2004. http://europa.eu.int/comm/transport/care/.

[7] Toyota, "Toyota safety: Toward realizing zero fatalities and accidents." Presentation, 2004. http://www.toyota.co.jp/en/safety presen/index.html.

[8] "Standard specification for telecommunications and information exchange between roadside and vehicle system - 5GHz band dedicated short range communications (DSRC) medium access control (MAC) and physical layer (PHY) specifications," Tech. Rep. ASTM E2213-03, ASTM, September 2003.

[9] "Report and order," Tech. Rep. FCC 03-324, USFCC, December 2003.

[10] "IEEE DSRC standards group meetings." Meeting Minutes. www.leearmstrong.com/DSRC/DSRCHomeset.htm.

[11] "Dedicated short range message set (DSRC) dictionary," Tech. Rep. Standard J2735, SAE, 2006.

[12] V. Kawadia and P. Kumar, "Principles and protocols for power control in ad hoc networks," in *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 76–88, January 2005.

[13] Y.-B. Ko and N. Vaidya, "Geocasting in mobile ad hoc networks: Location-based multicast algorithms," in *WMCSA*, (New Orleans), 1999.

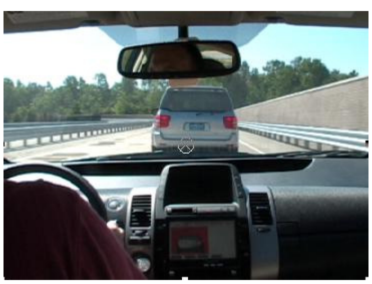
Application	Comm. type	Freq.	Latency	Data Transmitted	Range
Traffic Signal Violation	I2V One-way, P2M	10 Hz	100msec	Signal Status, Timing, Surface Heading, Light Posn.,Weather,	250m
Curve Speed Warning	I2V One-way , P2M	1 Hz	1000msec	Curve Location, Curvature, Speed Limit, Bank, Surface	200m
Emergency Brake Lights	Vehicle to Vehicle Two-way, P2M	10 Hz	100msec	Position, Deceleration Heading, Velocity,	200m
Pre-Crash Sensing	Vehicle to Vehicle Two-way, P2P	50 Hz	20msec	Vehicle Type, Yaw Rate, Position, Heading, Accel.	50m
Collision Warning	Vehicle to Vehicle One-way, P2M	10 Hz	100msec	Vehicle Type, Position, Heading Velocity, Acceleration, Yaw Rate	150m
Left Turn Assist	I2V and V2I One-way, P2M	10 Hz	100msec	Signal Status, Timing, Posn. Direction, Road Geom., Vel. Heading	300m
Lane Change Warning	Vehicle to Vehicle One-way, P2M.	10 Hz	100msec	Position, Heading, Velocity Accel., Turn Signal Status,	150m
Stop Sign Assist	I2V and V2I One-way	10 Hz	100msec	Position, Velocity Heading, Warning.	300m

Table 5: Eight high-priority vehicular safety applications as chosen by NHTSA and VSCC [2]. Note that communication frequency ranges from 1-50 Hz and maximum communication range span 50-300 meters. P2M represents 'Point-to-Multipoint', I2V represents 'Infrastructure-to-Vehicle' and V2I represents 'Vehicle-to-Infrastructure'.

(a) Three cars in transit. View of front vehicle obscured.

(b) Middle vehicle swerves to avoid braking front vehicle.

(c) Emergency braking required to stop last vehicle.



(d) With EBW, driver is alerted before seeing front vehicle and starts to brake.

(e) Safe stoping distance achieved.

(f) System hardware

Figure 6: Figures (a), (b) and (c) represent behaviour without the EBW system. Three vehicles are traveling at high speed as shown in Figure (a). The front vehicle begins to brake sharply (Figure (b)) causing the middle vehicle to swerve at the last moment. The result is emergency braking by the tail vehicle in order to avoid a collision in Figure (c). Alternatively, in Figure (d), as soon as the front car begins braking an EBW is transmitted via DSRC to the rear vehicle where the driver is alerted through the LCD screen, as well as via an alarm over the audio system. Ample time is then available for the tail vehicle to stop as shown in Figure (e). Figure (f) shows the hardware deployed in the rear of a Toyota Prius. The CAN network is accessed on the left hand side, a GPS receiver is in the middle and the DSRC radio as well as the mini computer are on the right hand side.

#	Data Element	# of Bits	Signal Violation	Curve Warning	Emerg. Brake	Pre-Crash Sensing	Collision Warning	Turn Assistant	Lane Warning	Stop Sign Assist	# of uses
1	Acceleration	12	●	●	●	●	●	●	●	●	8
2	Acceleration Precision	3	●	●	●		●	●		●	6
3	Airbag Count	7					●				1
4	Ambient Air Temp.	8									0
5	Anti-Lock Brake State	2	●		●		●				3
6	Brake Applied Pressure	4	●	●	●		●	●		●	6
7	Brake Applied Status	4	●	●	●		●	●		●	6
8	Brake Boost Applied	1			●	●	●				1
9	Driving Wheel Angle	8		●							4
10	DSRC Message ID	3	●	●	●	●	●	●	●	●	8
11	Elevation Confidence	4		●							3
12	Elevation	20		●		●					2
13	Exterior Lights	3			●						4
14	Heading	16	●	●	●	●	●	●	●	●	8
15	Heading Precision	3	●	●	●	●	●	●	●	●	8
16	Headlights	2						●			1
17	Lateral Acceleration	12	●	●		●	●		●	●	6
18	Latitude of center of vehicle	32	●	●	●	●	●	●	●	●	8
19	Longitude of center of vehicle	32	●	●	●	●	●	●	●	●	8
20	Obstacle Direction	16		●		●	●				3
21	Obstacle Distance	10		●		●	●				3
22	Longitudinal Acceleration	12	●	●	●	●	●	●	●	●	8
23	Positioning Precision	4	●	●	●	●	●	●	●	●	8
24	Rain Sensor	3		●				●		●	3
25	Siren In Use	2	●								1
26	Speed	16	●	●		●	●	●	●	●	7
27	Speed Precision	3	●	●		●	●	●	●	●	7
28	Stability Control Status	3		●	●	●	●	●			5
29	Steering Wheel Angle	16	●	●			●		●		4
30	Steering Wheel Angle Precision	2	●	●			●		●		4
31	Steering Wheel Rate of Change	8		●		●	●	●	●		5
32	Sun Sensor	10									0
33	System Health	4				●					1
34	Throttle Position	8	●	●			●	●		●	5
35	Throttle Precision	3	●	●			●			●	4
36	Time Precision	4	●	●		●	●		●	●	6
37	Temporary ID	48					●		●		2
38	Traction Control State	2		●					●		2
39	Turn Signal/Hazard Signal	2	●				●	●	●	●	5
40	Two Byte Tag List	16									-
41	UTC Time	40	●			●	●	●	●		5
42	Vehicle Length	14				●	●	●	●	●	5
43	Vehicle Width	10				●	●		●		3
44	Vehicle Height	8		●					●		2
45	Vehicle Mass	8		●	●	●	●				5
46	Vehicle Type	7		●		●		●			3
47	Vertical Acceleration	8									0
48	Vertical Acceleration Threshold	4									0
49	Wiper Rate	8						●			1
50	Wiper Status	3						●			1
51	Yaw Rate	16		●							1
52	Yaw Rate Precision	3		●							1

Table 6: A subset of the more than 70 Data Elements included in the SAE Common Message Set [11], accompanied by the authors analysis of element usage in the eight NHTSA-defined safety applications [1], [2].