Efficient Message Composition and Coding for Cooperative Vehicular Safety Applications

C. L. Robinson, *Student Member, IEEE,* D. Caveney, *Member, IEEE,* L. Caminiti,
G. Baliga, K. Laberteaux *Member, IEEE,* and P. R. Kumar, *Fellow, IEEE.*

(Invited Paper)

Abstract—**Wireless inter-vehicular communication will enable a broad range of applications in the future. However, multiple vehicles, multiple diverse applications, multiple vehicle manufacturers and the mobile vehicular environment raise several issues related to utilizing the wireless medium effectively.**

In this paper we present a method for efficiently communicating vehicle data amongst neighbouring vehicles, primarily for safety applications. The "Message Dispatcher" (MD) coordinates communication between applications and the wireless channel at the application level. It addresses technical implementation issues, business deployment considerations and issues of extensibility and system architecture. The Message Dispatcher concept has become an integral part of the Society of Automotive Engineers safety message standardization effort.

We shall describe the Message Dispatcher and present results illustrating its utility. We also describe a deployment in several vehicles at the Toyota Technical Center in Ann Arbor, MI. Then, using data collected from the vehicles, we investigate a Predictive Coding method for data transmission using the Message Dispatcher. We show that this scheme can reduce wireless channel utilization and bandwidth requirements by over 80% compared to regular transmission methods. Several insights for future wireless channel usage optimization are provided.

I. I

TO enable proliferation of wireless communication between vehicles we must consider methods to more efficiently use the scarce and shared wireless medium. Competition for access to the wireless channel will only increase as more vehicles become equipped with the required technology. New and evolving applications will also drive this demand and generate new data and transmission requirements. This paper describes a method for efficiently creating message packets for wireless vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. It addresses several implementation issues, restrictions and objectives, and describes some preliminary solutions to these problems. The general approach is to reduce channel load by exploiting similarities in transmitted data, as well as efficient inclusion of individual data elements.

We focus predominantly on intervehicular safety applications. In this domain, Dedicated Short Range Communications (DSRC) [8]–[10] is the leading wireless technology under consideration [1]. Significant progress has been made in standardizing the lower layer protocols for DSRC [8], [10]. Safety applications and related technologies are being examined by industry/government consortiums, such as the Crash Avoidance Metrics Partnership (CAMP) [17], [18], the Car2Car Communications Consortium [19] and the Advanced Safety Vehicle (ASV) Project [20].

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 standard SAE J2735, Dedicated Short Range Communication (DSRC) Message Set Dictionary [11] defines several static message structures, as well as a method to efficiently and dynamically create messages. This flexible method was contributed to SAE by the authors of this paper in the form of the Message Dispatcher which is fully described, and extended upon, below.

As we shall describe in Section II, the automotive applications that appear to have the greatest safety potential rely heavily on single-hop broadcast communication with nearby vehicles and infrastructure. Further, it appears likely that implementation of vehicular safety applications will be executed quasi-autonomously by each manufacturer. The eventual vehicular wireless communication 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. This 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.,

This work has been supported by a Toyota research contract at the University of Illinois at Urbana Champaign.

C. L. Robinson is with the Department of Industrial and Enterprise Systems Engineering and P.R. Kumar is with the Electrical and Computer Engineering Department at the University of Illinois at Urbana Champaign, IL, 61801 USA e-mail: clrobnsn—prkumar@uiuc.edu.

D. Caveney. L. Caminiti and K. Laberteaux are at the Toyota Technical Center in Ann Arbor, MI. email lorenzo.caminiti—derek.caveney—ken.laberteaux@tema.toyota.com.

G. Baliga is with Google Inc. gibaliga@gmail.com.

windscreen wiper status) need not be sent frequently[1].

The Message Dispatcher (MD) addresses this Data Element coordination issue. Briefly, the MD sits between the applications and the lower-layer protocols, to coordinate the data requirements of each application. Its goal is to reduce the redundancy of the broadcast data. We shall focus on two approaches to achieve this in this paper.

First, duplicate data elements from various applications are combined into a single message and only sent once. Further efficiencies can be obtained by coordinating transmissions so as to meet minimum transmission frequency or latency requirements. We present some primitive results showing that the MD approach greatly reduces the expected channel usage.

Second, using ideas from Predictive Coding [21], one can use a model for state estimation, and an update scheme which transmits data elements only when the model estimate is inaccurate. We also examine the predictive coding method whereby small state 'corrections' requiring fewer bits are transmitted, as compared to a full data element. We shall demonstrate that these methods have the desirable effect of reducing channel utilization, thereby mitigating channel congestion and thus data loss and delay. We shall quantify the reduction in this paper.

The Message Dispatcher has been successfully implemented in a testbed for cooperative vehicle safety demonstrations at the Toyota Technical Center (TTC), confirming its feasibility and flexibility. Using actual vehicle data collected from the testbed for multiple drivers driving multiple routes on public roads, we show that using predictive coding can reduce channel usage by over 80%.

The remainder of this paper is organized as follows. In Section II, the unique characteristics of the Safety Vehicular Ad-hoc NETworks (SVANET) are described. This includes a description of some of the the current trends in safety applications and their Data Element requirements. The section also identifies several ideal features of a SVANET system. Section III outlines the Message Dispatcher architecture. Section IV describes the MD implementation and vehicle testbed at TTC. In Section VI we describe the predictive coding approach and based on the testbed data we analyze the reduction in channel load. Section VII evaluates the MD against the goals outlined in Section II. Conclusions as well as potential extensions are made in Section VIII.

Some of this work has previously been presented in [16]. The extensions in this paper are sections related to the predictive coding, as well as the collection and analysis of the testbed data sets.

II. P F

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.

A. Vehicular Safety Applications and Data Requirements

Significant efforts, involving the vehicle industry and government agencies mentioned in Section I, 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 [1], [2] shown in Table I along with their *proposed* communication requirements[2].

We highlight that these requirements have only been proposed. There has, to the authors' knowledge, been no thorough investigation or deployment of these systems on which these requirements are based. This highlights the requirement that any communication protocol be sufficiently flexible to enable changing requirements. The salient features of the table are that the 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, some 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., latitude and longitude position, heading, acceleration (with varying precision: 4bit, 8bit, 16 bit), headlight status and brake status). Of these Data Elements, thirty of the most frequently used elements are selected as the basis of a "common message set". The common message set is intended to provide a standardized set of messages which vehicles could use to communicate. The messages were all of fixed size, structure and contents. However, since very few of the intended applications have actually been implemented or fully developed, the exact usage characteristics of the safety messages were in flux at the time they were being defined and standardized. This made the process difficult, as these choices are likely to be refined over time. It is however clear from Table I 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 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. This observation motivated the work in this paper.

B. Broadcast Characteristics

As illustrated in Table I, safety messages tend to be locally broadcast with a maximum transmission range

[1] This statement ignores a potential requirement that new neighbours of a vehicle be promptly updated with the vehicles current state. This can be addressed in our proposal, as described in Section VI-C.

[2] Similar deliberations are underway in Europe and Asia, with similar results.

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 I
E - NHTSA VSCC [2]. N 1-50
H 50-300 . P2P ‘P - -P ’, P2M ‘P - -M ’, I2V
‘I - -V ’ V2I ‘V - -I ’.

.

of 300 meters; messages sent by a vehicle will contain Data Elements useful to multiple vehicles in the nearby vicinity. As DSRC radios are required to communicate at least 300 meters, we assume that safety messages broadcast their messages in a single hop[3].

As a result of the highly dynamic vehicular environment, it is likely that the nearby neighbors of a vehicle will change frequently. It is likely unnecessary, and more difficult, to maintain an updated topology of 1-3 hop neighbors. Again, 1-hop broadcasts seem appropriate. Further, coordinating transmission between vehicles will be difficult. Packet interference and loss seem likely. We approach this issue by attempting to reduce channel load and use the channel infrequently.

C. 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 a 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 with multiple safety applications. However, creating and testing these safety applications is an ongoing effort. As such, a scheme must be backward compatible as well as future proof to newly-defined, evolving, or upgraded applications.

Flexibility: It seems likely that in the heterogeneous marketplace for vehicles, different vehicles will be running different subsets of safety applications. A scheme should be sufficiently flexible to account for this, as well as the other requirements mentioned 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 constraints on the information they are able to communicate.

Unified 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.

Layered Architecture: By providing a layered architecture that abstracts the message sending interface from the application designer, a separation of concerns for the application designer is achieved. This enables easier, faster and more modular development.

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

Information Rate: Some Data Elements, such as headlight status, change infrequently. Thus, a solution should distinguish these properties and transmit information only when it is appropriate.

Recognize vehicle capabilities: Not all vehicles will be able (or willing) to measure and transmit certain pieces of information. 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.

[3] 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.

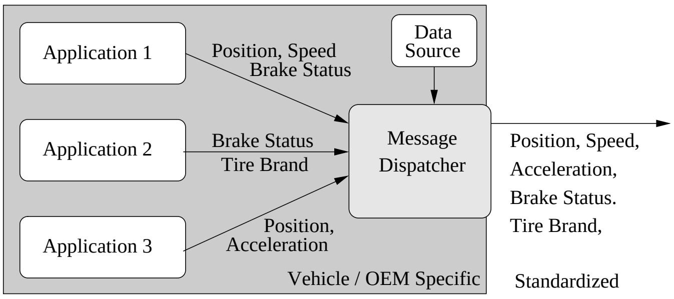


Fig. 1. The Message Dispatcher assimilates data requirements from all the on-board applications and compiles a single message using a dictionary of defined Data Elements and standardized message construction guidelines. Some of the data may be obtained directly by the MD from an onboard data source (e.g., CAN bus).

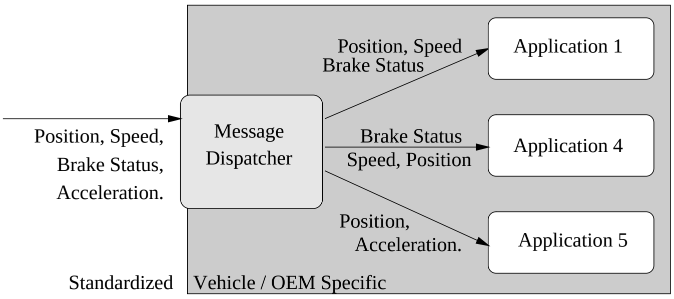


Fig. 2. A receiving Message Dispatcher is responsible for separating and disseminating Data Elements from the received message to all on-board applications, as well as managing data requirements for surrounding vehicles.

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

III. M D

The basic architectural concept of the Message Dispatcher (MD) 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 application layer and the communication stack.

Safety applications will register or send Data Elements to be broadcasted to the MD. The MD then summarizes these Data Elements across applications and creates a single packet comprising the minimum set of the Data Elements to be transmitted (See Figure 1). In some implementations the MD might collect data from other data sources within the vehicle (e.g., through the CAN bus). The MD would also consider data requirements of other surrounding vehicles or roadside units, as described in Section III-C. This combined message 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 Dictionary (Section III-A). Second, the specification of how these elements should be combined into a message (Section III-B).

A. Data Element Dictionary

This section describes how Data Elements are identified and formatted. The section also describes the authors' proposal for adding new elements to the Data Element dictionary. The SAE standard [11] identifies over 70 data elements in its "Data Element Dictionary". Each element in the dictionary is defined using the fields indicated in the example in Table II.

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 II
A L D E .

Using this data dictionary, a message can be constructed by creating a string of unique identifiers followed by the value of the Data Element. Further, this unique ID overhead can be reduced when related Data Elements are grouped into a "Data Frame". For example, latitude is frequently updated and transmitted together with longitude. Overhead is reduced by formatting latitude and longitude into a single "position" Data Frame with a single ID. Each Data Frame consists of Data Elements in a specified order and thus their unique ID's are not required within the Data Frame. 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, which is discussed in Section III-B.

B. 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 (either in a Frame or as an individual Data Element) elements in a particular message 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

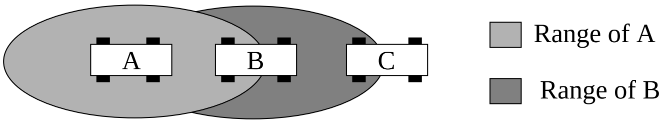


Fig. 3. Vehicle A sends a message that the Message Dispatcher on vehicle B matches on subsequent transmissions. Although Vehicle C is beyond the range of interest of Vehicle A, it too begins to match the message resulting in a "racing" condition.

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.

The third section is reserved for the inclusion of ad-hoc or newly defined terms by using a lightweight labeling scheme. The schema defines an escape character which indicates the start 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. Using the tag the Message Dispatcher would poll the subscribed applications for knowledge of the incoming Data Element.

C. What and When to Send

Determining what elements are required to be sent by a vehicle in order to satisfy *surrounding* vehicles is a problem in SVANET. Within CAMP or SAE there is still no way to obtain specific information from a newly encountered vehicle. Sending a large packet, comprising all defined Data Elements, at the maximum required rate among all elements is very inefficient. While not the main contribution of this paper, we propose two possible solutions easily implemented under the MD architecture:

- *Match Received Message*: If a Message Dispatcher receives a message with a Data Element it is not currently transmitting, it should include its own version of that data element in the following transmition. In this way it 'matches' the incoming message. To avoid a "racing" situation shown in Figure 3, the original sender includes a Data Element indicating whether its message contents should be matched.
- *Request Data Elements*: Define and send a Data Element that specifically requests certain elements. This solution would be useful in probe type applications where a roadside unit requests information from passing vehicles.

IV. E A

This section illustrates how the MD concept can be leveraged to efficiently construct a messages for multiple safety applications. It further describes our MD implemention. The two implemented safety applications, the Emergency Brake Warning (EBW) application and the Intersection Violation Warning (IVW) application, are described (along with their data requirements) in the following subsections. It is likely that all DSRC-equipped vehicles will send out a Heart-Beat Message (HBM) comprising a minimal set of Data Elements and Frames, as shown in Column 2 of Table III. We will look at ways to reduce the heart beat message channel load in a later section.

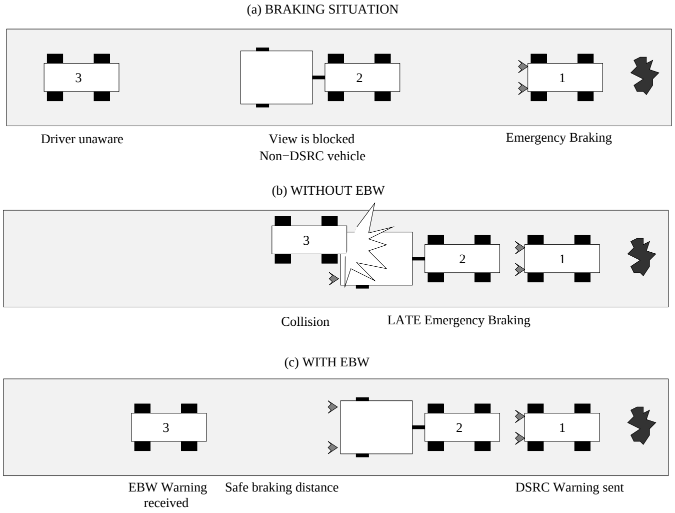


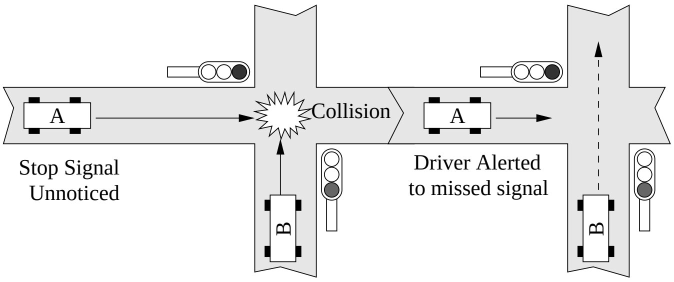
Fig. 4. Brake lights are often difficult to see if there is a blocking vehicle. EBW provides a brake warning by using wireless communication to allow safe stopping.

A. 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 Figure 9. *Method of Operation:* Consider Figure 4. When vehicle 1 performs severe-braking, a request is passed to the MD to begin transmitting the Data Elements in Column 3 of Table III at the indicated frequency. Note the data with ID *AH:EBWBreadcrumb*, which represents a sampled-path history of the vehicle, has not been defined in the Data Dictionary. It is appended to the message using the schema described in Section III-B. On reception, vehicle 3 determines if the braking car is in its forward path using the *AH:EBWBreadcrumb* data element. The vehicle can then take appropriate action. The path history is only used when a severe-braking event occurs and need not be sent until it is needed. A series of pictures illustrating the functionality of EBW are included in Figure 9.

B. Intersection Violation Warning

The Intersection Violation Warning (IVW) application warns the driver if violating a red light seems imminent. See Figure 5. Vehicles approaching the intersection are also warned if an approaching vehicle has issued a warning. As shown in column 4 in Table III, many of the required data elements are similar to the EBW application, although they may have different frequency requirements. Some data elements are unique to the IVW application and have not yet been defined in the Data Dictionary (e.g. *AH:IVWMap*).



(a) No IVW

(b) With IVW

Fig. 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: Roadside units transmit traffic light information including its location, light status, time till color change, dimensions of intersection (called data element *AH:IVWMap*), etc. The IVW application registers with the MD to receive all incoming IVW related Data Elements. Vehicles then determine if a signal violation is imminent. If so, the driver is alerted and a message is sent to the traffic light and surrounding vehicles indicating that a violation is likely. Thus, the message dispatcher only sends IVW data when triggered by a violation event.

C. Message Composition

Consider a vehicle is speeding toward a red light. The driver has been alerted to a potential violation and is braking sharply. Thus, both the IVW and EBW systems are active. Table III lists a *subset* of the Data Elements and transmit frequencies which the Message Dispatcher must satisfy.

The Message Dispatcher combines the required Data Frames and Data Elements into a minimal set of messages. Duplicates are ignored, such as with *DE:Acceleration* and *DE:IVWWarningVehPos* in Table III. Another example is data element *DF:PositionShort* which is not included in outgoing messages as it is a subset of the information already included in *DF:PositionLong*.

Since the applications have registered data at different frequencies, the MD constructs three different messages, called Msg10Hz, Msg5Hz and Msg3Hz. The message contents are described in the final 3 columns of Table III. Note that the Data Elements in the 10Hz message also appear in the 5Hz and 3Hz message. To meet the frequency requirements and minimize the utilized bandwidth, the MD will send the messages in the sequence illustrated in Table IV.

D. Implementation

The Message Dispatcher has been implemented by the Toyota Technical Center in two Toyota Prius cars. 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 9(f)).

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 IV

T M
D T III. N ( ) T
M 10H M 5H . ( ) T
M 5H M 10H
M 3H .

The MD implementation uses a callback mechanism to interface with applications. Upon initialization, applications register with the MD those Data Elements it will **provide** and those it wants to **receive** from the MD. Data Elements may be both provided and received. During Data Element registration, the application supplies the MD with the callback method to be invoked when it is time to send the Data Element, or when the MD has received an updated Data Element over the channel.

At the end of the registration process, the application specifies the frequency of transmission and informs the MD to begin a periodic transmission of these Data Elements. The MD is then responsible for the message composition detailed in Section IV-C, using the provided callback method 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. 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, the MD scheduler may reschedule future periodic transmission times.

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, a traffic light and two vehicles each run their own MD.

E. Analysis

A basic evaluation of the performance of the MD for the two-applications in the 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

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 III

A D E (DE) F (DF) (E B W (EBW)
I V W (IVW)), H B M (HBM). C 5
M D
. C 6, 7 8 3, 5 10H S IV-C.

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 V

C ( ) M
D (MD) C M S (CMS) . T
4th
.

bytes.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 that this Common Message Set (CMS) is periodically transmitted with the highest frequency in Table III, 10Hz, in order to meet the requirements of all the applications. For one vehicle, this requires a channel usage of 14.1kbps, as shown in Table V. Conversely, a HBM being sent out by the MD at the 3Hz frequency specified in Table III requires only 0.6kbps. This is a reduction of 95% of channel load when neither of the applications are required to transmit. Channel usage rises to 3.7kbps when IVW events occur (initiating a 10Hz transmission) and to 6.2kbps when EBW events occur (initiating a 5Hz transmission). If both IVW and EBW events occur on one vehicle, the channel usage is equal to that of the CMS. However, this assumes that the elements for either application are sent continuously. Since the MD can dynamically manage message contents, the full EBW/IVW message need only be sent in EBW/IVW instances, which we assume to occur with the (overly generous) percentage frequencies shown in the Column 1 of Table V. Thus, the expected channel load, shown in Column 3, is far less than the peak channel load.

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. 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 load possible by employing the MD is relevant.

In Section VI we shall present a further method for reducing the channel load which can readily be implemented in the Message Dispatcher framework.

V. D S

We have described our MD deployment and vehicle testbed in Section IV-D. Using the testbed we collected several data sets under a variety of driving conditions. We have choosen three sets for further analysis. The sets were all recorded in and around Ann Arbor, MI. The trajectories are shown in Figure 6, and described in Table V. The data can be downloaded at [22]. Each data set contains samples of vehicle position (latitude and longitude), speed and acceleration (lateral and longitudinal). Other data such as brake status, brake pressure and steering wheel angle were also recorded.

Parameter	Data Set A	Data Set B	Data Set C
Environment	Urban	Urban	Highway
Sample Freq.	5Hz	5Hz	5Hz
Duration	8min 24sec	10min 34sec	6min 17sec
Length	6.1km	7.6km	9.1km
# of Stops	2	2	0
# Turns	7	5	0

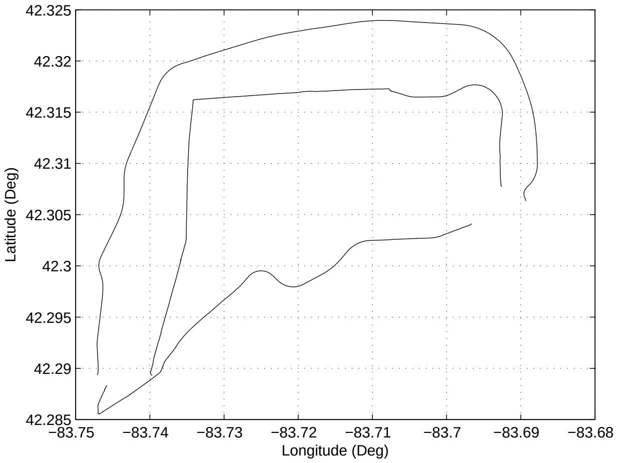


Fig. 6. Traces of the three recorded data sets. The lower trajectory represents Data Set A, the middle Data Set B and the upper trajectory is the Highway Data Set C.

VI. P C

The objective of this section is to show, using real world data, that predictive coding can significantly reduce the channel load in vehicular safety applications.

The analysis thus far has basically been focused on two concepts:

1) Avoid duplication of data in multiple messages.
2) Compose sequences of messages so as to only send data at the minimum required update rate.

However, neither of these notions considers the uncertainty or variability associated with the underlying data. For example, analysis of the data sets reveals that brake status changes infrequently. When it does change, the status is often maintained. So, compared to vehicle longitude which changes rapidly and continuously when a vehicle is moving, routine brake status can be transmitted less frequently. Only status changes need be transmitted. Of course, upon a change, it can and should be transmitted immediately. There is a difference between nominal state update transmission, and rapid reaction to changes.

Indeed, we can extend this notion to encompass data which change frequently, but which can be easily predicted. For example, given a particular vehicle position, velocity and steering angle, the future trajectory of the vehicle can be predicted. Hence, to extend the example above, the longitude may not need to be transmitted frequently either, if it can be predicted well. E.g. A stationary vehicles position can easily be predicted.

The main idea presented in this section is to transmit data only when the error in estimating the state is 'sufficiently large'. We shall characterize the model used for the state estimation as well as the notion of 'sufficiently large error'. This idea is similar to conserving bandwidth while using computational power [15]. We also consider the case when the error is small only the least significant bits, or some smaller correction update is sent, rather than a complete data element with full resolution.

These type of transmission policies represent Predictive Coding (PC), which we describe in the following section.

A. Predictive Coding

Linear Predictive Coding (LPC) [21] is commonly used in the transmission, reproduction and generation of human voice over low data rate channels. It is a method of encoding signals in which the value of the signal at each sample time is predicted as a linear function of the past values of the signal [14]. Predictive Coding does not require the use of a linear prediction function.

The design of a Predictive Coding scheme can be divided into two parts. The first is to determine an appropriate model to be used to predict the signal which is being sampled. In voice applications, a linear model of a particular order is assumed. The model parameters are obtained using a least squares or auto-regressive fit on a window of sampled data. The model parameters (i.e., the coefficients of the characteristic polynomial and the model gain) are then transmitted across the channel. In general, the order of the model is chosen so as to be able to accurately reproduce the signal given a particular model input. This is the second part of LPC design: specifying the model input. In voice applications, the input is characterized by the voice pitch frequency. An appropriate pulse train and white noise signal is used as a model input.

Our approach differs from LPC in the following ways. First, our model is not linear. Second, we apriori define a Newtonian model to represent the behaviour of the vehicle. The model is described in Section VI-B. By defining the model apriori, we have specified the model coefficients. Thus, in our update scheme, we shall only transmit state updates, and not the model coefficients. Third, we do not characterize the input function at every time instant, but rather assume a zero-order hold for intermediate values of the state which are not modeled. Finally, we do not choose a fixed sampling and transmit frequency. Instead we shall dynamically choose a state update frequency which ensures that the state estimate is always within some tolerable error bound.

B. System Model

In this section we describe a simple first order model used for state prediction. We use the discrete time index k. Transmission of a particular data element does not occur at fixed time intervals, but will occur at the discretized time instants. Define the time between transmission instants as Δ_k. We define the following velocity estimate update:

$$\hat{v}_k = \begin{cases} \hat{v}_{k-1} + \hat{a}_k \Delta_k & \text{when } v_k \text{ not transmitted,} \\ v_k & \text{when } v_k \text{ is transmitted.} \end{cases} \quad (1)$$

where $\hat{v}(k)$ and $\hat{a}(k)$ represent *estimates* of the velocity and acceleration at time k. The term v_k represents the *sampled* vehicle velocity. The estimate of the distance, $\hat{D}_k$, moved by the vehicle in time Δ_k, is computed using

$$\hat{D}_k = \hat{v}_k \Delta_k + 0.5 \hat{a}_k \Delta_k^2. \tag{2}$$

The estimate of the vehicle heading, $\hat{\Phi}$, is updated using a non-slip tri-cycle model:

$$\hat{\Phi}_k = \hat{\Phi}_{k-1} - \frac{\hat{v}_k \Delta_k}{l} \tan \hat{\Psi}_{k-1}, \tag{3}$$

where Ψ is the angle of the front wheels and l is the wheel base of the vehicle. The angle of the front wheels is linearly related to the steering wheel angle, which we are able to measure. The distance moved, 2, is resolved into a displacement using the most recent heading estimate. A longitude and latitude update can then be performed using this displacement. Estimates for any other data element not included in the model are given by a zero order hold of the most recently transmitted value. For example, the acceleration estimate is updated as follows:

$$\hat{a}_k = \begin{cases} \hat{a}_{k-1} & \text{when not braking,} \\ a_k & \text{when } a_k \text{ transmitted.} \end{cases} \tag{4}$$

Whenever a state observation is transmitted it replaces the state estimate at that time, as illustrated by the velocity update in (1).

Our estimation model represents a Kalman Filter update scheme with noiseless observations. There is scope for improvement in this regard by deploying a full Kalman filter, and using a higher dimensional model for state prediction and incorporating models for other parameters. This is beyond our purpose here, which is simply to demonstrate that using predictive coding yields a significant transmission data rate reduction for vehicular communication.

C. Tolerable Error

In Table III we have presented transmission frequencies for various data elements in two example applications, as well as a 'heart beat' message. We refer to these fixed frequency requirements as the 'regular transmission scheme'. To the best of the authors' knowledge, selection of these frequency values in practice has been based on three requirements:

1) To provide neighbouring vehicles with a sufficiently accurate estimate of a vehicle's current state. That is, to maintain the state estimation error within some 'tolerable error'.
2) Ensuring vehicles entering an area receive a timely introduction from their new neighbours.
3) Ensure that neighbours are quickly updated about a state transition.

Data Element	E[Error]
Year	0
Month	0
Day	0
Hour	0
Minute	3.33E-003
Seconds	0.2
mSeconds	398
Speed (m/s)	0.0716
Longitude (Deg)	2.54E-005
Latitude (Deg)	1.65E-005
Heading (Deg)	2.42
Long. Accel. (m/s^2)	0.0769
Lat. Accel. (m/s^2)	0.0687
Yaw Rate (m/s)	0.248
Steering Wheel Angle (Deg)	1.04
Brake Switch	5.70E-003
Brake Torque (Nm)	5.89

TABLE VI

E

. A . F ,

1.9 .

In the Predictive Coding scheme we propose here, there is no requirement that data elements be transmitted at a particular frequency. Hence, the state estimation error incurred between successive transmissions may be very large. To address the first item listed above, we shall use the state estimation error between successive sample times (under the regular transmission scheme), to define the tolerable error. This is the error measured at time $k+1$ when using a zero-order hold of the state at time k to predict the state at time $k+1$. Computing the expected value of this error yields what we call the 'Expected Error', shown in Table VI. The time between transmissions is 0.2 seconds (5Hz transmission frequency). We use this error value in the following section as an indicator as to when a data element should be transmitted. This approach also gives an indication of when to transmitt data so as to satisfy the last requirement in the list above.

To address the second requirement achieved by a regular transmission frequency, that is that new neighbouring vehicles are informed of the state, we stipulate that a data element must be transmitted at least as frequently as some lower bound. The frequency is chosen to be small compared to the regular transmit frequency, e.g., 0.25Hz. We shall also investigate the effect of not having this requirement at all.

D. Predictive Coding Transmission Policies

We now describe five transmission policies for reducing the number of bits transmitted over the channel:

1) Data elements are transmitted at the transmit frequency specified in Table III. This is a reference policy.
2) Only transmit data elements when their estimation error exceeds the tolerable error bound in Table VI. A zero-order hold is used in between transmissions, i.e., the value of a data element is held constant until a new value is received. There is no predictive coding used. When an update is required the full data element is sent. Each data

element must be sent at a frequency of at least 0.25Hz.

3) Same as policy 2) except that the model described in Section VI-B is used to predict the state in between sample transmissions. I.e., Predictive Coding. When an update is required, the full data element is sent. Each data element must be sent at a frequency of at least 0.25Hz.
4) Same as in Policy 3), except that, when possible, a smaller state 'correction' is sent instead of the full data element. For example, when the error is sufficiently small, instead of transmitting the 4 byte Longitude data element, we transmit a 1 byte correction. A *complete* data element must be sent at a frequency of at least 0.25Hz.
5) Same as policy 4), except that there is no minimum update frequency requirement.

In all the policies, whenever a data element is sent the time in millseconds must also be sent. We have used the data element definitions from the SAE standard [11]. For our purposes here we shall consider how these policies reduce the channel loading caused by a Heart Beat message transmitted at 5Hz, and containing all of the data listed in the left column of Table VI. Note that the data elements related to time (e.g., Year, hour, etc.) are included as time-stamp for the other Data Elements in the packet, and is not for clock synchronization.

E. Predictive Coding Results

We have summarized the results of the predictive coding analysis in several ways. Table VII shows the average send frequency of each data element under the five policies. Several observations can be made. First, data elements that are highly predictable (e.g., year, month, and hour) do not need to be sent regularly. Hence, under policies 3 and 4 their send frequency is almost the minimum update frequency. Under policy 5 there is no minimum update frequency, and hence the required transmission frequency approaches zero (they must be sent at least once to start). Also note that since longitude and latitude change frequently while moving, they are required to be updated most frequently under policy 2. They can, however, be predicted fairly well, and hence under the remaining policies their frequency of transmission drops considerably. We also note, that data elements which are not predicted (e.g. Yaw rate) maintain a fairly constant transmit frequency across all policies, as expected. A more descriptive model would reduce their transmit frequency. Finally, since we assumed that *msecond* data element is sent with any and every transmission, its high frequency of transmission (almost 5Hz) indicates that some sort of data is sent at almost every transmission opportunity. Thus, there may be additional bandwidth savings realized by adding a data element to a packet already scheduled for transmission if it prevents the additional data element being sent in its own message later. This is especially applicable when there is significant message overhead.

Data Element	Average Transmission Frequency (Hz)			
	Policy 1	Policy 2	Policy 3, 4	Policy 5
Year	5	0.25	0.25	0
Month	5	0.25	0.25	0
Day	5	0.25	0.25	0
Hour	5	0.25	0.25	0
Minute	5	0.25	0.25	0.02
mSecond.	5	4.86	4.49	4.35
Speed (m/s)	5	2.01	1.76	1.76
Longitude	5	2.90	0.26	0.07
Latitude	5	2.97	0.27	0.1
Heading	5	0.69	0.57	0.48
Accel. x (m/s^2)	5	1.83	1.83	1.8
Accel. y (m/s^2)	5	1.95	1.95	1.93
Yaw Rate	5	1.39	1.39	1.36
Steering Wheel	5	1.59	1.59	1.56
Brake Switch	5	0.27	0.27	0.04
Brake Torque	5	0.69	0.69	0.48

TABLE VII

S

.

Data Set	Policy #2	Policy #3	Policy #4	Policy #5
A	62%	77%	81%	84%
B	63%	77%	80%	83%
C	57%	78%	82%	84%

TABLE VIII

P

P #1.

Table VIII represents the percentage reduction in the transmitted rate for the four policies in comparison to the reference policy. Since highway driving has less abrupt changes in acceleration and direction as compared to urban driving, we can expect the state prediction for highway driving to be somewhat better than for urban driving. This is reflected by the larger reduction in message size associated with the highway driving data set (Data Set C). Note that this is not the case under policy 2 because messages need to frequently include position updates, which change frequently during highway travel. The results also indicate that policy 5) does not perform that much better than policies 3 and 4. This is because, as shown in Table VII, some sort of message is sent at almost every opportunity.

Finally, Figure 8 presents a rolling average of the length of the messages transmitted along the trajectory followed in Data Set A (shown in Figure 7). Inspection of the data elements contained in the message during the 'peaks' indicate that they are correlated to changes in acceleration, such as when braking or accelerating before and after a stop. Acceleration is not modelled in our predictive model. Incorporating it indicates a future direction for extending the predictive coding method.

These results lead us to a few conclusions. First, predictive coding is highly effective at reducing the channel usage for vehicular safety applications. Even a simple state triggered transmission policy (such as the zero-order hold in policy 2) is significantly better than a

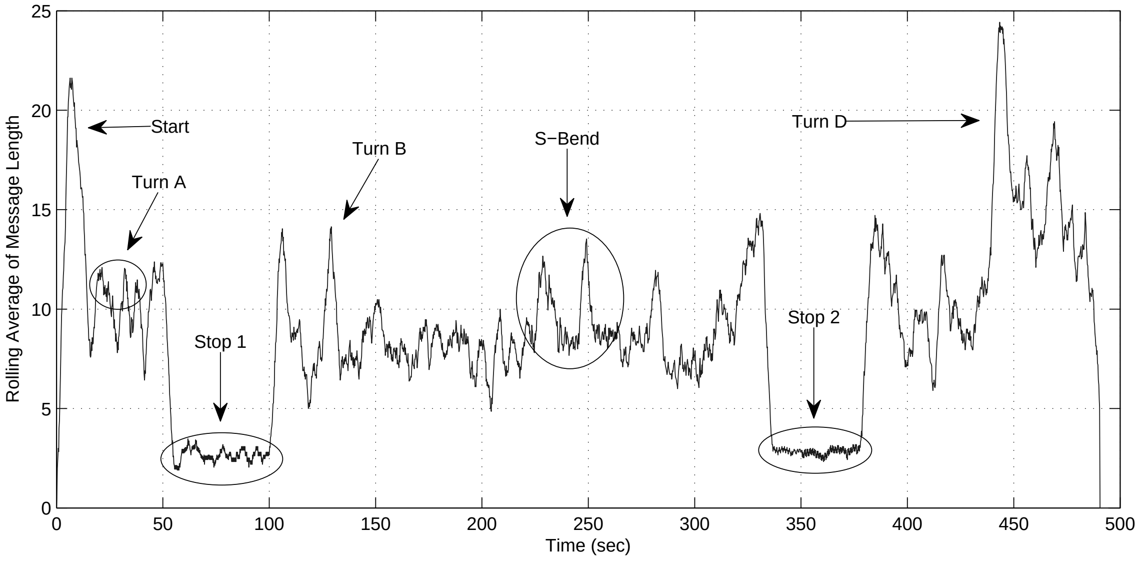


Fig. 8. Representation of the length of messages transmitted along the route shown in Figure 7. A rolling average of 20 messages is used and the length is defined as the number of characters transmitted in each message. The majority of the 'spikes' are caused by changes in acceleration, such as before and after a stop.

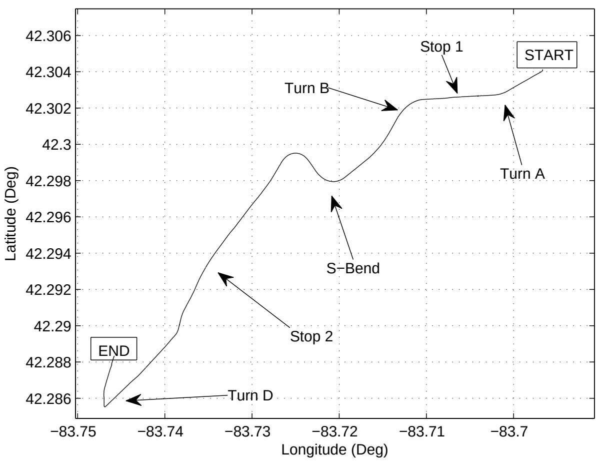


Fig. 7. The route corresponding to Data Set A. The points marked correspond to those in Figure 8.

regular high frequency transmission approach. Second, future message optimizations should focus on reducing the number of transmitted messages, as compared to trying to reduce the size of individual messages.

VII. A A

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

Each application stipulates its data requirements to a single location, the Message Dispatcher. Thus, any upgrades or modifications to the interface requirements are well defined and localized (i.e., the *single interface* goal is met). The Message Dispatcher generates a combination message satisfying application requirements. Applications do not consider how Data Elements are shared between vehicles, achieving an effective *separation of concerns*. When the message format or protocol changes, only the MD implementation must change, and not the applications. This is an important abstraction for both the application and communication layer designers. Algorithms for avoiding channel overload, secure data transmission, and low latency message delivery can be implemented in the Message Dispatcher enabling *lower bandwidth usage*. Messages composed do not have to contain redundant information and thus can *recognize vehicle capabilities*.

The Message Dispatcher to evolve as application requirements change bBy using the flexible terminal character with post-fixed identifier scheme. If a a unique or new set of Data Elements are required, these can be incorporated only when required. This illustrates *flexibility*, *future proof* and *extensibility* of the architecture. This enables the MD to effectively *exploit the communication ability*. Finally, since the applications have been successfully separated from the communication protocol, companies are able to develop their own products and thus *enable product differentiation*.

VIII. C E

This paper provides an account 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. Many of the application data requirements either partially or completely overlap. This motivated our introduction of the Message Dispatcher concept, which leverages the Safety VANET environment and significantly improves efficiency of the wireless DSRC channel. We have presented the Toyota Technical Center testbed which we used to collect real world data for analysis. We proposed a predictive coding scheme for reducing channel load and showed, based on our data, that using even a simple first order model can reduce channel load by over 80%. We discussed how the MD architecture meets several technical and business objectives and also enables other useful functionality.

In addition to being adopted by the relevant SAE standard, the Message Dispatcher has received wide appreciation among vehicle manufacturers. A current three year CAMP project, funded by NHTSA, will demonstrate the MD in a demonstration project involving 5 automobile manufacturers.

There are several interesting research avenues opened and enabled by this architecture. For instance, channel loading can now explicitly be managed through dynamic message construction (e.g., based on vehicle traffic), packet collision avoidance algorithms can be implemented (e.g. transmit power modulation), specific 'important' data elements could be retransmitted in the case of loss, cooperation between MD could be enabled to manage channel usage, enabling data delivery notifications, giving priorities to information (e.g., low latency or application dependencies), filtering or other modifications to the raw incoming data can be performed and even multi-hop information passing schemes can be implemented. Extensions to our predictive coding approach are also possible.

IX. A

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 needed for our testbed.

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

R

[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 Motor Corp., "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.

[14] American National Standards Inc., "ATIS Telecom Glossary 2000," at http://www.atis.org/tg2k/t1g2k.html.

[15] J. K. Yook, D. M. Tilbury, N. R. Soparkar, "Trading Computation for Bandwidth: Reducing communication in Distributed Control Systems Using State Estimators" in *IEEE Transactions on Control Systems Technology*, vol. 10, no. 4, pp. 503-518, 2002.

[16] C. L. Robinson, L. Caminiti, D. Cavney and K. Laberteaux, "Efficient coordination and transmission of data for vehicular safety applications", in *The Third ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 10-19, 2006.

[17] www-nrd.nhtsa.dot.gov/pdf/nrd-12/CAMP3/images/CAMP-IVIThirdAnnualReport.pdf

[18] www-nrd.nhtsa.dot.gov/pdf/nrd-12/CAMP3/images/CAMP-IVIThirdAnnualReport.pdf

[19] http://www.car-2-car.org/

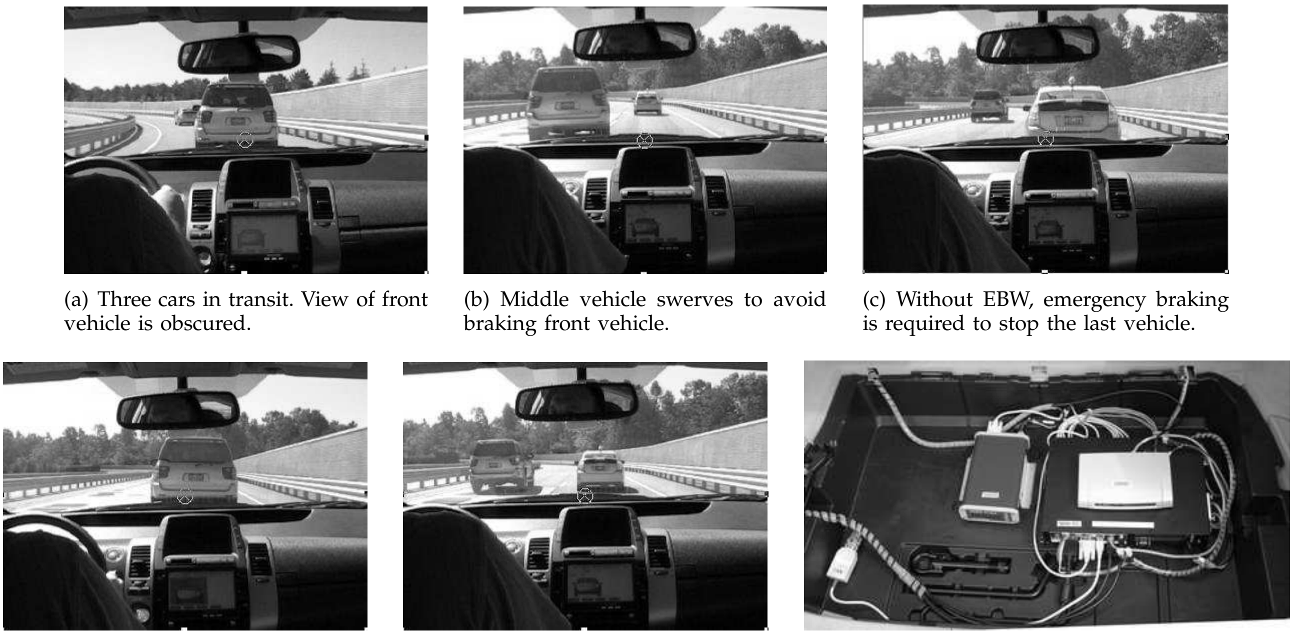
[20] http://www.mlit.go.jp/

[21] P. Elias, "Predictive coding I", in *IEEE Transactions on Information Theory,*, vol. IT-1, no. 1, pp/ 16-24, March 1955.

[22] http://decision.csl.uiuc.edu/ testbed/TTCUIUCData.zip

Craig Robinson received a B.Sc. E. degree from the University of the Witwatersrand, South Africa in 2000. He was awarded a Fulbright Scholarhip in 2001 under which he completed his M.Sc. at the University of Illinois at Urbana Champaign (UIUC) in 2003. He is currently completing his Ph.D. at UIUC in the Industrial and Enterprise Systems Engineering Department. His research interests include performance in networked control systems, estimation and control in wireless communication environments and designing system architecture for such systems.

D. Caveney received his B.Sc.E degree in Applied Mathematics from Queen's University in Kingston, Ontario, Canada in 1999 and his M.Sc. and Ph.D. degrees in Mechanical Engineering from the University of California, Berkeley, in 2001 and 2004, respectively. He then acted as a Visiting Postdoctoral Scholar with the Center for Collaborative Control of Unmanned Vehicles at the University of California, Berkeley for 2004 to 2005. Since 2005, Dr. Caveney has been a Senior Research Scientist with the Toyota Technical Center in Ann Arbor, Michigan, where his interests include cooperative control of vehicles for safety and mobility.



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

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

(c) Without EBW, emergency braking is required to stop the last vehicle.

(d) With EBW, driver is alerted as the braking occurs and starts to brake.

(e) Safe stopping distance achieved.

(f) System hardware

Fig. 9. Illustration of the functionality of the Emergency Brake Warning (EBW) system. Figures (a), (b) and (c) represent behaviour without the EBW system. Three vehicles are traveling at high speed are 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 and a potential collision by the rear vehicle in Figure (c). Alternatively, with EBW 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.

Lorenzo Caminiti is a Sr. Engineer at the Toyota Technical Center. His current research interests include the creation and validation of advanced vehicular active safety systems. He received his masters in engineering from the University of Rome. He has extended experience in software design and development for embedded and networked systems.

Girish Baliga developed a middleware framework for networked control systems under the guidance of Prof. P. R. Kumar for his PhD in Computer Science at the University of Illinois at Urbana-Champaign. He also received an MS in Computer Science and an MS in Mathematics from the University of Illinois, and a B. Tech in Computer Engineering from the National Institute of Technology, Surathkal, India. He is currently a Software Engineer in the System Infrastructure Group at Google, Inc.

Ken Laberteaux is a Senior Principal Research Engineer for the Toyota Technical Center in Ann Arbor, MI. Dr. Laberteaux's research focus is information-rich vehicular safety systems, focusing on architecture and protocol design for vehicle-to-vehicle and vehicle-to-roadside wireless communication. He is one of the founders and two-year (2004, 2005) General Co-Chair of the highly-selective, international Vehicular Adhoc Networks (VANET) workshop. Dr. Laberteaux serves as the technical lead for communications of the multi-year, multi-million dollar Vehicle Safety Communications-Applications collaboration project between the US Government and several automotive companies.

P. R. Kumar is the Franklin W. Woeltge Professor of Electrical and Computer Engineering at the University of Illinois, Urbana-Champaign. He is a Fellow of the IEEE, a recipient of the Donald P. Eckman Award of the American Automatic Control Council, the IEEE Field Award in Control Systems, the IEEE Communications Society Fred W. Ellersick Prize, and a member of the National Academy of Engineering, U.S.A. He has worked on problems in game theory, adaptive control, stochastic systems, simulated annealing, neural networks, machine learning, queuing networks, manufacturing systems, scheduling, and wafer fabrication plants. His current research interests are in wireless networks, sensor networks, and the convergence of control, communication and computation.