An Integrated Network of Roadside Sensors and Vehicles for Driving Safety: Concept, Design and Experiments

Hua Qin, Zi Li, Yanfei Wang, Xuejia Lu and Wensheng Zhang
Department of Computer Science
Iowa State University, USA
Emails: {qinhua,zili,yanfei,xuejialu,wzhang}@cs.iastate.edu

Guiling Wang
Department of Computer Science
New Jersey Institute of Technology, USA
Email: gwang@njit.edu

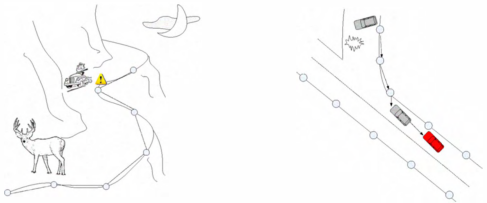
***Abstract*—One major goal of the vehicular ad hoc network (VANET) is to improve driving safety. However, the VANET may not guarantee timely detection of dangerous road conditions or maintain communication connectivity when the network density is low (e.g., in rural highways), which may pose as a big threat to driving safety. Towards addressing the problem, we propose to integrate the VANET with the inexpensive wireless sensor network (WSN). That is, sensor nodes are deployed along the roadside to sense road conditions, and to buffer and deliver information about dangerous conditions to vehicles regardless of the density or connectivity of the VANET. Along with the concept of VANET-WSN integration, new challenges arise and should be addressed. In this paper, we investigate these challenges and propose schemes for effective and efficient vehicle-sensor and sensor-sensor interactions. Prototype of the designed system has been implemented and tested in the field. Extensive simulations have also been conducted to evaluate the designed schemes. The results demonstrate various design tradeoffs, and indicate that satisfactory safety and energy efficiency can be achieved simultaneously when system parameters are appropriately chosen.**

***Keywords*-Vehicular Ad Hoc Networks, Wireless Sensor Networks, Integration**

I. Introduction

Driving is an indispensable part of the life of many people. The past years have witnessed substantial efforts on improving *driving safety*. Among them, the most prominent technological one might be the emerging vehicular ad hoc network (VANET) and the safe driving-targeted applications built atop the VANET. The VANET is composed of highly-mobile vehicles and sparsely-deployed roadside stations, each equipped with wireless communication devices and optionally with sensing devices. Wireless communication can be conducted between vehicles and/or between vehicles and roadside stations. On top of the VANET, applications have been developed to collect, process, share and deliver real-time information about road conditions.

These systems sometimes help in accident prevention, but they are not always effective since the underlying VANET does not provide guaranteed real-time detection of road conditions or communication connectivity. Firstly, the VANET only opportunistically monitors road conditions. That is, only when there exists a vehicle or a roadside station detecting or being notified of some conditions, can the information be shared within the VANET. Secondly, the VANET can be disconnected due to high mobility and unpredictable movements of vehicles and the sparse deployment of roadside stations. If the VANET is disconnected, critical information about road conditions known by one partition of the VANET cannot be shared timely with vehicles that need to know it but are in other partitions.



(a) (b)

Figure 1. Examples for VANET-WSN Integration

Deploying more roadside stations appears to be a solution. This, however, may significantly increase the investment cost; also, lack of power supply is a big obstacle to do so in rural areas. Wide area wireless networks (such as cellular networks) could be used to connect disconnected segments of the VANET. This approach may achieve communication connectivity, but it does not solve the problem of lacking guaranteed real-time sensing of road conditions.

Towards addressing the above problems, we propose to integrate the VANET with the WSN to provide timely detection of road conditions and to help connect partitioned segments of the VANET. Wireless sensor nodes, for example, MicaZ motes [1], are much *cheaper* than roadside stations. Besides, some inexpensive and low-power sensing modules, for example, the WiEye passive infrared sensors [2], have been commercialized and can be installed on the motes to sense road conditions with low cost.

These sensor nodes can be deployed along roadside [3]–[5] with higher density than current roadside stations to form a connected network together with the VANET. The sensor nodes can sense the road conditions, collect and process the sensing data to find out information useful for safe driving, and deliver the information to vehicles

that need it. The sensor nodes also can buffer the safety-related information generated by vehicles, and forward the information to vehicles in different partitions of the VANET.

Following are some examples showing that deploying WSNs can greatly help in preventing road accidents:

- **Example I.** Deploying WSN along rural roads can help prevent vehicle-animal collision accidents. As shown in Fig. 1 (a), the WSN nodes can detect a deer roaming on the road and propagates the information within the nearby area. Approaching vehicles will get the warning beforehand. The advantage brought by the deployment of WSN is significant. It may help to avoid 1.5 million vehicle-deer collisions happening every year (according to auto insurer State Farm) which result in about 150 deaths and $1.1 billion losses [6].
- **Example II.** Fig. 1 (b) shows that, bad road conditions (e.g., slippery surface) detected by an isolated vehicle can be told to nearby roadside WSN nodes, and the WSN nodes can then collaborate with each other to propagate the information to other vehicles approaching this dangerous area. Note that, this cannot be accomplished if only VANET can be used since the VANET is not connected.

To realize the proposed VANET-WSN system, several important issues should be investigated. Firstly, the system should be *viable* in the real scenarios. The impacts of interference, noises and other environmental factors on system performance should be investigated. Secondly, the system should be *scalable*, considering the large scale of highway system in the world. As the scale of deployment increases, the difficulty in deploying and maintaining the system should not increase much, and the quality of service and the energy efficiency of the system should remain stable. Thirdly, the system should be *flexible* to changes in the real world. WSN nodes may fail or lose time synchronization, the highways may be extended or reshaped, and traffic pattern may change from time to time. It is desired that the deployment and the working parameters of VANET-WSN system can be adjusted with low overhead as the above changes happen. Fourthly, *energy efficiency* should be maximized for the roadside WSN. Although WSN nodes can be deployed and redeployed by humans and their batteries can be replaced manually when necessary, it is still important to minimize the energy consumption and maximize the network life time to reduce energy and maintenance costs. Finally, satisfactory *quality of service* should be attained. Dangerous road conditions should be detected and the information about the dangers should be delivered to related vehicles in a timely fashion to ensure driving safety.

Towards tacking the above issues, this paper makes the following major contributions:

- We adopt the idea of group-based modular design to achieve *scalability* and *flexibility*. In our design, the roadside WSN is made up of sensor groups. Each group works autonomously and asynchronously, and neighboring groups interact with each other through a gateway node shared by them. Deployment or redeployment of a group does not affect others; topology and working parameter adjustments conducted within each group do not affect others, either.
- The objectives of *energy efficiency* and *quality of service* are achieved by (i) an event-driven duty cycle scheduling strategy which leverages the VANET to minimize energy consumption in the WSN, and (ii) low-contention and low-delay communication protocols which ensure contention-less intra-group transmission and can reduce inter-group contentions with certain coordination costs.
- A prototype of our system has been implemented and tested in the field to study the *viability* of the system. Based on realistic vehicle traffic traces and roadside sensor-to-sensor communication traces, extensive simulations have also been conducted. The results demonstrate various design tradeoffs, and indicate that desired quality of service and energy efficiency can be achieved simultaneously with appropriately chosen system parameters.

To the best of our knowledge, this is the first work that proposes, implements and evaluates an integrated VANET-WSN system for driving safety.

In the rest of the paper, Section II presents an overview of our proposed system, which is followed by the design details in Section III. Section IV and Section V report our implementation and simulation results. Finally, Section VI concludes the paper.

II. System Overview

A. Network Deployment

The proposed system consists of vehicle nodes and sensor nodes. Each vehicle node has two communication interfaces: a WiFi (IEEE 802.11) interface for communication with other vehicle nodes; and a ZigBee (IEEE 802.15.4) interface for communication with roadside sensor nodes. In our prototype, each vehicle node is an onboard laptop with an embedded WiFi card and an attached Telosb mote [1].

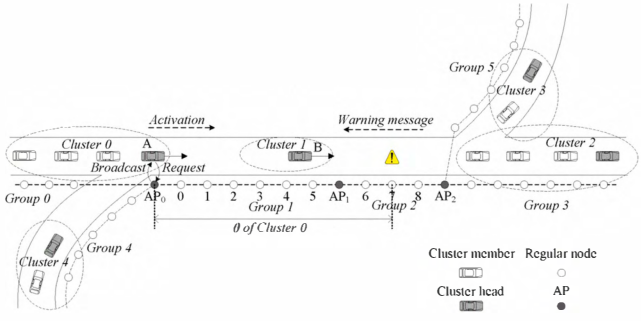


Figure 2. Network Deployment

Each sensor node has a ZigBee interface used to communicate with other sensor nodes and with vehicle nodes,

and in our prototype, each sensor node is a Telosb mote. Sensor nodes are also mounted with sensors which are used to sense road conditions.

As illustrated in Fig. 2, sensor nodes are deployed along one side of the highway. We consider only one-way highways, though the system can be extended to two-way roads. The sensor nodes form a connected network. According to their roles, sensor nodes have two types: the *regular sensor node* and the *access point sensor node* (called *AP* thereafter), and can sense and relay messages, while APs have extra responsibilities of discovering and communicating with vehicles, and managing the network. APs are much fewer than regular nodes. Regular nodes that are deployed between two adjacent APs form a *group*. As shown in Fig. 2, one highway may merge into another one, two highways may be connected with a ramp, and one highway may branch into two or more highways; hence, the roadside sensor network is not linear. In our design, the node connected with three or more linear segments must be an AP.

In practice, some roads (e.g., in mountain areas) may be more prone to safety-related events than others; hence, sensor nodes may only be deployed along the roads with high risks. This way, deployed sensor nodes do not form a single connected network, but multiple disconnected networks. Our design is flexible and is applicable to such deployment due to the modularity approach adopted.

B. Duty Cycle Scheduling and Message Forwarding

A connected partition of vehicular nodes on a highway forms a cluster. Cluster formation has been widely studied and is beyond the scope of this paper. Each cluster maintains a *cluster head*, a node which is running at the front of the cluster. It is responsible for communicating with roadside sensors on behalf of the whole cluster. As shown in Fig. 2, there are five clusters, where cluster 2 and cluster 3 are connected but they are on different highways.

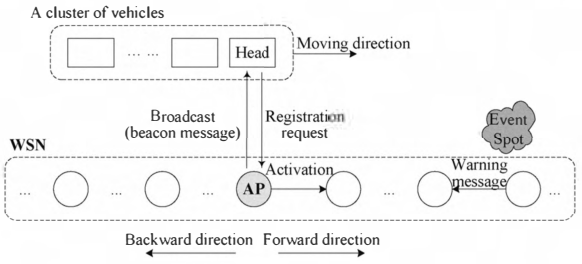


Figure 3. Big Picture of the Integrated VANET-WSN System

As illustrated in Fig. 3, each AP periodically broadcasts a beacon message. If the AP has buffered some safety-related information that its nearby vehicles should be aware of, it will piggyback these messages in its beacon message. When a passing cluster head hears the message, it sends its registration request to the AP.

In response to the request, the AP activates sensor nodes that are within a certain number (denoted as θ, a system parameter) of hops along the moving direction of this cluster (called *forward direction* hereafter), if these nodes have not been activated yet. By this, these waken-up sensor nodes will be able to proactively monitor the conditions of the roads. To save energy, a roadside sensor is active only when there is a vehicle cluster approaching to its sensing range within θ hops. If there is no vehicle approaching this sensor, the sensor does not need to work.

If a dangerous condition is detected (e.g., a deer is roaming on the road), the detecting sensor node will generate a warning message and propagate it along the direction opposite to the moving direction of the vehicles (called *backward direction* hereafter) until the message reaches the heads of all incoming clusters that requested the activation of the sensor nodes. Then, the warning message can be propagated within the clusters of vehicles by using a certain data dissemination protocol such as [7]–[9]. This way, VANET nodes are leveraged whenever possible to reduce the workload of the roadside WSN to save its energy.

III. Detailed Design

The above description on duty cycle scheduling and warning message propagation remains high-level. To realize these functionalities, practical and efficient protocols should be designed for scheduling duty cycles of sensors and for propagating activation messages in forward direction or warning messages in the backward direction. Since the duty cycle scheduling and message propagation are not independent of each other, we will study them together.

One big challenge in designing these protocols is that, the forward and the backward propagations take place on the same communication channel, and hence they should be scheduled appropriately to avoid or reduce collisions to minimize both propagation delay and energy consumption. Although CSMA/CA-based protocols are commonly used in wireless ad hoc and sensor networks, TDMA-based protocols are preferred in our system for following reasons:

- As opposed to CSMA-based MAC protocols commonly used in wireless networks, sensor nodes in the proposed system often have very little data to transmit (packets are generated only when cluster heads pass APs or events are detected by sensors). Meanwhile, once there is data to transmit, the data should be transmitted in a timely fashion to guarantee quality of service. If CSMA/CA is adopted, time and energy may be wasted for long idle listening, medium contention, etc.
- To improve network throughput and support real-time data delivery in WSN, TDMA-based MAC protocols [10]–[12] have been proposed recently. Although they can achieve real-time transmission, their different application scenarios (e.g., high data rate, special network structure, etc.) make them unsuitable for our proposed system. Moreover, these protocols are for unidirectional communication,

while our system requires bidirectional. Thus, designing a new TDMA-based protocol becomes necessary.

- Further, the special network topology in the proposed system can facilitate the application of TDMA-based protocol. Each sensor has a limited number of neighboring nodes, which are pre-determined, making the assignment of the time slots for transmission easier. By carefully assigning the transmission slot, we can avoid or greatly mitigate the hidden terminal problem that is hard to be solved by using CSMA/CA-based protocols.

However, TDMA-based protocols require time synchronization among nodes, which is hard to accomplish in large-scale systems. To accommodate bidirectional communication in a single channel and achieve scalability, we adopt the idea of modularity: sensor nodes are divided into groups; within each group, duty cycles of nodes and bidirectional propagations are scheduled to achieve both contention-less communication and energy efficiency; inter-group communication is handled by APs shared by different groups.

In this section, we first present our proposed intra-group and inter-group scheduling schemes. Then, we discuss the choice of system parameters.

A. Intra-group Scheduling

Sensors in the same group are time synchronized (See Section IV). The time is divided into slots of fixed length. During each slot, a packet can be sent from a sensor to its neighbors. We call the length of a slot a *packet time (denoted as τ)*. A certain number of slots form a period, and the length of a period is denoted as p. Protocols for duty cycle scheduling and medium access control (MAC) are presented in the following such that, every c_f period(s), a packet can be propagated hop by hop from the most back sensor of the group to the most front sensor along the forward direction, and every c_b period(s), a packet can be propagated hop by hop from the most front sensor to the most back sensor along the backward direction. Here, we call c_f the *forward interval* and call c_b the *backward interval.*

1) Scheduling for Forward or Backward Propagation: Without loss of generality, let us consider the example shown in Fig. 4, where circles $A, B, \cdots, E$ represent sensors in the same group, A is the most back sensor and E is the most front sensor. We want to schedule the duty cycles of these nodes and their communication behaviors such that a packet can be forwarded from A to E hop by hop.

Taking into account the unique characteristics of the network topology, we adopt the following methods to design forwarding propagation protocol which has no contention, high energy efficiency and low propagation delay:

Firstly, TDMA-based access control is adopted to eliminate contention. For each sensor, a certain number of slots are reserved for it for sending or receiving. The reservation of slots follows the following rule: During the slot reserved for node X for sending, none of its one-hop and two-hop neighbors is allowed to send packets. For the example in Fig. 4, during the slots for sensor C to send, sensors A, B, D and E are not allowed to transmit. This way, contention (even the hidden terminal problem) can be eliminated.

Secondly, the broadcast nature of transmission is leveraged to speed up packet propagation and reduce acknowledgement overhead. Specifically, after a node has received a data packet from its previous hop, it forwards the packet to the next hop immediately in the next slot. Due to the broadcast nature of transmission, the data packet can also reach the previous hop, serving as the acknowledgement. If the packet cannot reach the previous hop due to errors in the channel, the packet that has arrived at the next hop can be propagated further without waiting for the acknowledgement packet being successfully sent to the previous hop.

Thirdly, reserved retransmission slots can be dynamically shared among sensors in the same group. For reliability, retransmission slots are reserved for sensors. However, the quality of different links may not be the same and may change dynamically. For example, sometimes the link between sensors A and B may be better than the link between D and E, and vice versa in other time. Considering this, our design can enable sensors to dynamically share a certain total number of retransmission slots.

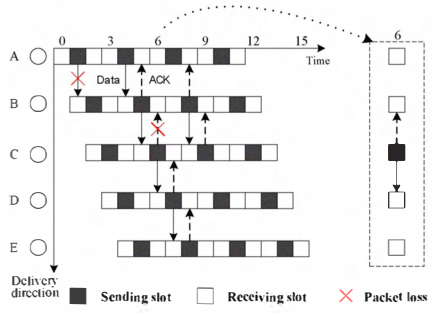


Figure 4. Example of Intra-group Scheduling for Forward Propagation

The forward scheduling is detailed as follows.

- *Reservation of slots*: The most back sensor is assigned with $3(r+1)$ sequential slots, where r is the system parameter specifying the maximum times to retransmit a packet by all nodes in the group, which we call *retransmission quota* hereafter. Without loss of generality, we call the first slot assigned to the node as slot 0, and the remaining slots are called slot $1, 2, \cdots, 3(r+1)-1$, respectively. Slots $3i+1$ $(i = 0, \cdots, r)$ are reserved for sending while others are reserved for receiving. If we use R to represent a slot for receiving and S to represent a slot for sending, all these slots can be represented as a sequence of $r+1$ RSR's. For each of the remaining sensors in the group, it is also assigned with $3(r+1)$ sequential slots of the same sensing/receiving pattern, except that its first slot is one slot later than that of its previous hop. In Fig. 4, the scheme for slot reservation is shown for a group composed of 5 nodes and $r = 3$.

- *Sending of a packet*: If a sensor has a packet to propagate, it will send it out at the first sending slot. If it overhears the forwarding of this packet or receives an acknowledgement in the next slot, which is reserved for receiving, from the next hop, the transmission is successful. Otherwise, it will retransmit the packet in the next sending slot. The procedure continues until the transmission is successful or the slots reserved for sending have been used up. In the case that the reserved slots have been used up, the packet can be transmitted in the next reserved propagation time (i.e., nearly $c_f \cdot p$ time later).
- *Receiving/forward of a packet*: If a sensor does not have any packet to send, it will listen in the first slot. If it does not hear anything from its previous hop, it can go to sleep in the following two slots since it can be predicted that it will not have any sending or receiving in the next two slots. If it receives a packet from its previous hop, it will transmit the packet to next hop immediately in the next slot, which is a slot reserved for sending. Then the follow-up procedure for checking if the packet has been successfully sent and retransmitting the packet is the same as in the part of *Sending of a packet*. Note that, if its forwarding is not overheard by its previous hop, the previous hop node may resend the packet. In this case, this forwarding node should be able to identify the duplication; then, it will send a dedicated acknowledgement packet to its previous hop in the next sending slot.

Note that, if multiple sensors in the group have packets to send, these packets can all be propagated except that, some sensor in the middle may have multiple packets to send/forward. In this case, it can merge these packets into one if possible, or send these packets one by one.

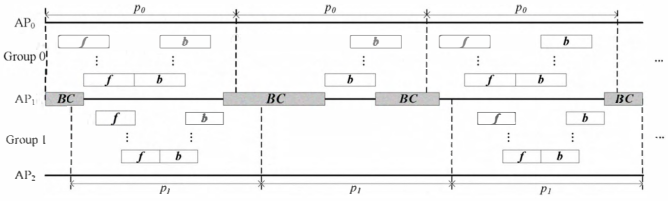


Figure 5. Forward and Backward Propagation Scheduling in a System with Multiple Groups (Group 0: $c_b = 1$ and $c_f = 2$; Group 1: $c_b = c_f = 2$; two groups have the same period $p_0 = p_1$; BC: slots that the AP_1 can use to broadcast beacon messages)

2) Example: An example for packet sending and forwarding is also shown in Fig. 4, which is explained as follows. Sensor A wants to send a packet to sensor E. It starts the transmission at its first available sending slot, slot 1. However, this packet gets lost. Hence, A will not receive the acknowledgement from B during the following receiving slot, so it retransmits the packet in the next available sending slot, i.e., slot 4, and it succeeds. Upon receiving the new packet, B immediately forwards the packet, which serves as both data packet to downstream node (C) and acknowledgement to upstream node (A). This sending packet has been received by C but is not acknowledged successfully. Thus, B assumes that C has not received the packet and retransmits that packet. This retransmission packet can be overheard by A and C. A simply ignores it while C attempts to resend the acknowledgement. Due to the good link quality in following propagation, the packet can eventually reach E even before every packet has been successfully acknowledged.

With the same idea, the protocol for backward propagation can also be designed similarly. Fig. 5 has shown examples of backward propagations next to forward propagations. A sensor can turn off its radio during the slots that are not reserved for sending or receiving.

B. Inter-group Scheduling

The scheduling of each group is made independently. When two groups are connected together at an AP, an issue arises: how can the AP successfully pass packets from one group to another with low delay? To address this issue, the AP needs to cooperate with its neighboring regular nodes (called *boundary nodes*, for example, nodes 5 and 6 are boundary nodes of AP_1 in Fig. 2) as follows.

The AP needs to know the schedules of its boundary nodes. For this sake, the boundary nodes periodically tell the AP their schedules, by either explicitly sending the schedule or implicitly piggybacking it in the data packet. Knowing the schedules of its boundary nodes, the AP should be active when any of its boundary nodes is active. This way, packets sending to the AP will not be missed if no collision occurs.

The AP also follows the protocol below to ferry packets between groups:

(i) Initially, the AP is in the *idle* state. Suppose the AP is connected with multiple groups, we call a group connects to it on the backward direction as its *upstream* group and a group connects to it on forward direction as its *downstream* group. For example, in Fig. 2, Group 1 is a upstream group of AP_1 while Group 2 is a downstream group of AP_1. When the AP receives a forward (backward) packet from its upstream (downstream) group, AP is bound to delivering the forward (backward) packet and hence sets itself to the *forward (backward) state*.

(ii) The AP in *forward (backward) state* is dedicated to delivering the forward (backward) packet. Any incoming backward (forward) packet will be just buffered and not acknowledged.

(iii) The AP will make an attempt to send the forward (backward) packet to the downstream (upstream) group if (a) any boundary node is in its forward (backward) receiving slots and (b) the last attempt was two time slots away from the current attempt to ensure AP to have enough time to get the possible acknowledgement.

(iv) Step (iii) is repeated until the AP has got an acknowledgement from its downstream (upstream) group. Then,

AP will check its buffers to see if there is any packet ready to be delivered. If so, step (iii) and (iv) will be repeated; otherwise, the AP goes back to step (i).

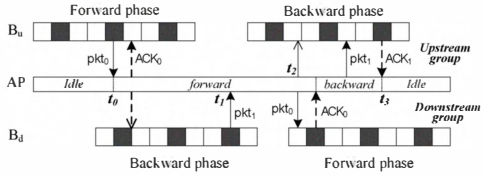


Figure 6. Example of Scheduling for Inter-group Communication

1) Example: Fig. 6 shows an example. B_u and B_d are two boundary nodes of the AP. At time t_0, B_u sends a forward packet (pkt_0) to the AP. Since the AP is in the idle state, it switches to the forward state and sends out the packet acting as acknowledgement. This packet can be also received by B_d as well. However, since B_d is now in backward phase, it will just buffer this packet without acknowledging it. Suppose at time t_1 B_d wants to send a backward packet (pkt_1) to AP. Since AP is now in the forward state, it will also just buffer this packet without acknowledging it. After some time, the forward phase of B_d becomes available. Then, the AP sends pkt_0 at B_d's first available receiving slot. Note that, even if this packet can not be successfully transmitted to B_d, B_d can still send out the acknowledgement for the previously buffered packet. Upon receiving the acknowledgement from downstream group, the inter-group delivery of that packet is accomplished. At that time, the AP checks its buffer and finds the backward pkt_1 is there to be delivered. Then, it changes to backward state and starts another inter-group delivery.

2) Resolution for Extreme Collisions: In some extreme case, data packets from two boundary nodes may arrive at the AP simultaneously and the schedules of these two boundary nodes match exactly. Then, the AP may never receive the data packet from either node because collision always exists. In this case, the above scheme fails. Thus, we propose the *yield* mechanism to deal with this situation. The basic idea is to let one boundary node yields to the other when they do not receive the acknowledgement from the AP for a certain number of times (which indicates the possible occurrence of collisions). At this time, one boundary node will start resending the packet once every two sending slots, while the other remains the same. Note that the working of this mechanism can be coordinated by the AP.

3) Broadcast of AP Beacon Messages: When none of the boundary nodes is in their reserved slots for backward or forward propagation slots, shown as "BC" blocks in Fig. 5, the AP can use some slot to broadcast beacon messages such that passing cluster heads can discover and contact with the AP, as described in Section II.B. The interval between two consecutive beacon messages should be short enough to ensure that a passing cluster head cannot miss it during its stay within the transmission range of the AP. During the time that the AP does not broadcast beacon messages and none of its boundary nodes is active in their forward/backward propagation, the AP can go to sleep to save energy.

C. Discussion on System Parameters

In this section, we show the relation between various system parameters by presenting our derived equations. Due to space limit, we will not show the detailed derivation.

1) System Parameters n and r: Based on the probability model for estimating the packet loss between two nodes in [13], we can derive the following inequality by requiring the expected number of *retransmissions* that a packet needs to cross a group should be no greater than the retransmission quota r. n is the number of sensors in a group and p_i is the packet loss ratio of node i in the group.

$$\sum_{i=1}^{n} \frac{p_i}{1-p_i} \leq r \tag{1}$$

If we assume that each sensor node has the uniform packet loss ratio, say $\bar{p}$, then from the Equation (1) we can get the lower bound of r.

$$r = \frac{n\bar{p}}{1-\bar{p}} \tag{2}$$

2) System Parameter c_f and c_b: Here, we only show the impact of c_f on the delay of forward propagation, and the impact of c_b on backward propagation is similar. The propagation delay within a group includes two parts: the intra-group delay (among regular nodes) and inter-group delay (at AP). Considering the *worst* case that the forward phase of the downstream boundary node is just over when the packet reaches the AP, the propagation speed within a group (including one AP), denoted by v_p, is

$$v_p = \frac{(n+1)I}{(3r+n+2)\tau + c_f p}, \tag{3}$$

where τ is the length of a slot, p is the length of a period (defined before) and I is the distance between two neighboring sensors. From this equation, we can see that by changing c_f we can dynamically control the propagation speed and further satisfy the delay requirement.

IV. Prototyping and Field Tests

We implement the proposed system and test it in the field. In the implementation, three components, namely, AP, regular node and vehicle node, have been prototyped. The vehicle node is implemented atop a laptop injected with a Telosb mote. AP and regular nodes are implemented atop Telosb motes with WiEye passive infrared sensors mounted. The Telosb motes run TinyOS-2.1.0.

For TDMA-based protocols to work, time synchronization is a prerequisite. To realize time synchronization, we use two interfaces provided by TinyOS-2.1.0 library: *TimeSyncAMSend* and *TimeSyncPacket*, which provide the primitives to synchronize a group of nodes through exchanging packets.

A. Field Tests

The major purposes of field tests are two folds. The first is to test if the proposed system works in field, and the second is to find out the impact of real environmental factors on the proposed system, especially on the communication of the system. Hence, we conduct two sets of field tests in a large open parking lot: One set of experiments are to test how the whole system works. The test is conducted in two scenarios: there exist intensive WiFi traffics nearby and there does not. Another set of experiments are conducted to measure the impacts of environmental conditions on communication between two Telosb motes when the interference level varies. Here, we elaborate the findings and results from the first set of experiments, while the results from the second set of experiments are used as inputs to our simulation which is discussed in Section V.C.

Two groups of Telosb motes (including totally 9 motes) are deployed along the roadside in a large open parking lot. The motes cover the length of 480 meters, the inter-mote distance is 60 meters. A vehicle repeatedly runs along the motes. Whenever the vehicle enters the road from one end and is discovered by an AP, the AP will wake up all the rest motes to start sensing. Warning messages are generated by the AP located at the other end of the road at a constant frequency, and the messages are propagated to the AP who discovers the vehicle and then is delivered to the vehicle.

Other experimental parameters are as follows. Transmission range is 100 meters. AP broadcasts beacon message every 10 seconds and an event is generated every 20 seconds. Retransmission quota (r) is fixed at 3 while the number of hops to activate (θ) is set to 8. Vehicle speed is about 20miles/hour. Each test lasts for 20 minutes.

B. System Performance with Interference

As WiFi communication is expected to co-exist with the proposed system, we first test the working and performance of the system when WiFi communication exists. For this sake, two laptops equipped with WiFi cards are put near each of the APs, respectively, to serve as interferers. To make the interference strong, about 10Mbps traffic is exchanged between the them, and the traffic is generated by using LAN Traffic V2 [14]. In the experiment, WiFi communication uses channel 6 (the default channel) and ZigBee uses channel 26. For comparison, we also conduct experiment for the situations of no WiFi traffic.

In these experiments, we set the forward/backward interval (c_f/c_b) and group size (n) to 3. We measured the average per-hop delay for the forward/backward message propagation, which are denoted as $D_{Forward}$ and $D_{Backward}$ respectively. The results are shown in the table below. From the results, we can see that the average delay measured with interference is slightly (i.e., between 5% and 9%) higher than that without interference for both forward and backward message propagation. Note that, the simulated interference traffic is intensive. This indicates that the impact of interference on propagation delay is not significant.

$n = c_f = c_b = 3$	With Interference	No interference
$D_{Forward}$(ms)	54.07	**51.06**
$D_{Backward}$(ms)	95.99	**88.31**

C. System Performance with Varying Parameters

Since the impact of interference from WiFi traffic is insignificant, we conduct more extensive experiments without the interference. In the experiments, we vary the system parameters (i.e., c_f, c_b and n) and measure the propagation delay. The results are as follows.

$D_{Forward}$(ms)	$c_f = c_b = 2$	$c_f = c_b = 3$	$c_f = c_b = 4$
$n = 3$	63.04	**51.06**	59.57
$n = 4$	205.19	75.73	157.25
$n = 5$	85.76	122.73	102.65

$D_{Backward}$(ms)	$c_f = c_b = 2$	$c_f = c_b = 3$	$c_f = c_b = 4$
$n = 3$	111.69	**88.31**	97.30
$n = 4$	540.59	105.39	189.08
$n = 5$	293.40	327.29	424.17

As we can see the largest forward propagation delay is about 205ms per hop, which means the speed for propagating activation messages from an AP which detects an incoming cluster of vehicles to other sensors that should be activated is about 293m/s, i.e., 659miles/h, which is much faster than the speed of a vehicle. The largest backward delay is about 540 ms per hop, which means the speed to propagate a warning message to related vehicles is about 111m/s, i.e., 250miles/h, which is also much faster than the speed of a vehicle.

We can also see that the backward delay is higher than the forward delay. The reason is found to be that, the forward phases of boundary nodes happen to have a better match than their backward phases in our experiments. Consequently, each forward packet arriving at APs can be relayed to the downstream group immediately, while some backward packets have to wait for the next available backward phase of the downstream group. Besides, we can see that the propagation delay goes up as the forward/backward interval increases most of the time. Occasionally, it varies. By analyzing the collected data at each node, we find that the variations are caused by the random packet loss, which affects the delays.

V. Simulation

NS2-based simulation has been conducted to evaluate our design. We evaluate the impacts of system parameters and environmental factors on system performance. The system parameters include *group size* and *forward/backward interval*. The performance metrics include *energy consumption* (the average energy consumption per hour of all APs and regular nodes) and *propagation delay* (the time from when the event occurs to when the cluster head receives the warning message, which is normalized as delay per hop).

We conduct both *theoretical evaluation* and *empirical evaluation*. For the theoretical evaluation, we vary the value of system parameters to evaluate our system performance.

For the empirical evaluation, we follow the empirical traffic data to generate traffic and use the packet transmission traces collected from field experiments.

A. Setup

The following table shows the parameters fixed in the simulation. We simulate a highway with more than 200 sensor nodes deployed along one side. Based on field experimental result, we set the packet time (i.e., length of a slot in the proposed protocols) to 25ms. We assume $c_f = c_b$. In addition, since retransmission quota is decided by group size and packet loss ratio as shown in Equation 2, we do not explicitly consider it in our simulation.

Road length	18900m × 20m
Number of hops to activate (θ)	50
AP Beacon interval	600ms
Sensor transmission range	100m
Inter-node distance (I)	90m
Vehicle transmission range	250m
Slot length (τ)	25ms
Average packet loss ratio at sensor and vehicle	15%
Interval between two events	6 minutes
Simulated time	1 hour

B. Theoretical Evaluation

Since the system performances are associated with three different parameters, we evaluate each performance metric by varying one parameter while fixing the other two. Besides, the arrival rate of the clusters is set to be 2 cluster per minute and the average speed of the vehicles is 30m/s (67.5miles/hour).

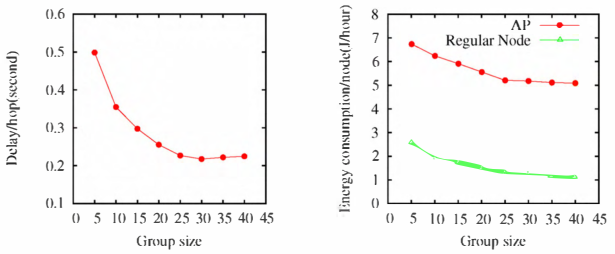


Figure 7. Impacts of n on System Performance

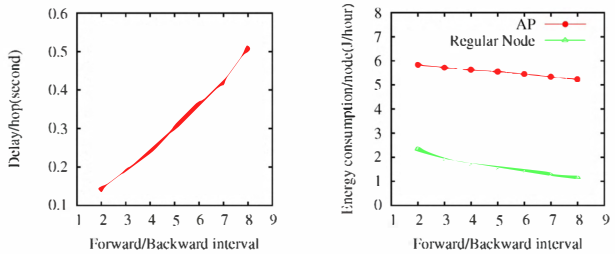


Figure 8. Impacts of c_f and c_b on System Performance

1) Group size n: Fix $c_f = c_b = 5$. In Fig. 7, we can see that the impact of group size on delay becomes insignificant as it goes up. This is because, by combining Equation (2) with (3), we know that v_p converges to a constant value as n approaches infinity and c_f (or c_b) and t are fixed. For the energy consumption on sensor side, we can see that a larger group consumes less energy per node. This is because forming a large group can reduce the number of boundary nodes and APs, which consume more energy than regular nodes, in a given area.

However, we can not conclude from the above results that, the larger the group size is the better performance we can achieve. The major problem of forming large group is that the message propagation delay at APs becomes larger as the group size increases. This means that the activation process becomes slow; hence, a vehicle may move ahead of the activation message, which is not desired for safety. Thus, an appropriate group size should be around 25.

2) Forward/backward Interval c_f and c_b: Fix $n = 20$. It is obvious that by using larger forward/backward interval the sensor nodes can get a larger fraction of time to sleep. Therefore, the delay increases and energy decreases accordingly, which is approximately linear, as shown in Fig. 8. Since group size is often pre-defined according to the road topology and packet loss ratio is related to environmental conditions, the forward/backward interval should be the key factor affecting our system performance.

C. Empirical Evaluation

In order for the simulation to better reflect the real-world traffic, we use the empirical vehicle traffic data, measured on I-80 highway in California in [15], to generate traffic for our simulation. Also in field experiments, we log the packet transmission of the sensor nodes under different traffic scenarios. These logs are transformed into the packet loss traces and then fed into our simulator to determine the reception and dropping of the incoming packets, serving as a realistic emulation of *packet loss ratio* in our system.

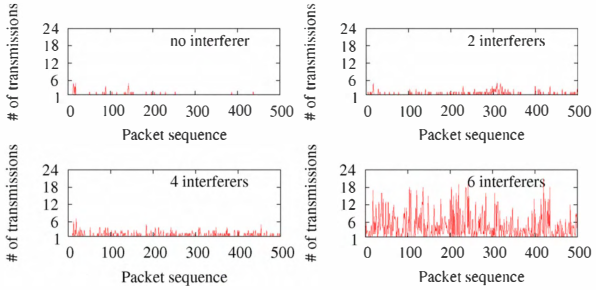


Figure 9. Outdoor Experiment: Packet Transmission Traces

1) Traffic generation: In our simulation, the vehicle clusters are generated following the three traffic categories proposed in [15]: *night traffic* with very low traffic volume and high speed (1 am - 3 am), *free-flow traffic* with moderate traffic volume and high speed (10 am - 12 pm) and *rush-hour traffic* with low speed and very high traffic volume (3 pm - 5 pm).

2) Packet transmission traces: To emulate the road-side interference, we deploy some sensor nodes in the middle of two *regular nodes* to act as the interferers by randomly

broadcasting messages (20 packets/sec on average). The reason that we choose sensors, rather than WiFi devices, as interferers is to make the interference more intensive since they share the same channel with roadside sensors. The distance between two nodes is nearly 100m. According to different traffic categories, different numbers of interferers are employed: 2 for night traffic, 4 for free-flow and 6 for rush-hour. For comparison, we also tested no interferer scenario. The number of transmissions for a serial of packets (called *packet trace*) are logged, as shown in Fig. 9.

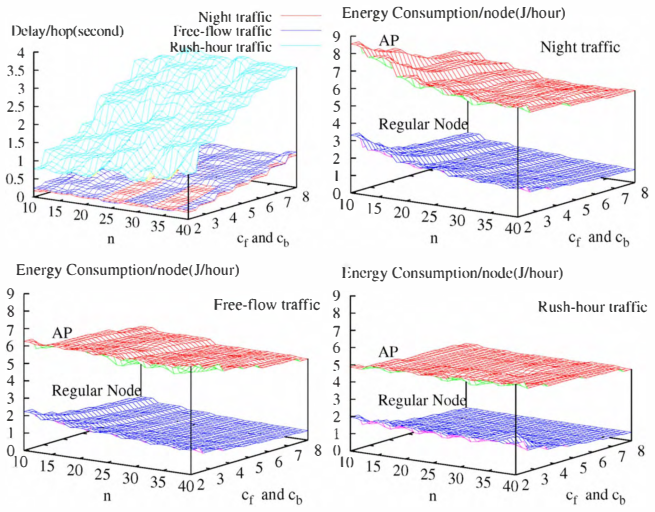


Figure 10. Impacts of n, c_f and c_b on System Performance

3) Results: Based on the empirical traffic generation and collected packet transmission traces, simulations are conducted. The first figure in Fig. 10 shows the propagation delay under three traffic categories. We see that forward/backward interval is the dominant factor that affects the delay, especially when group size is large. The delay in free-flow traffic is slightly higher than that in night traffic. However, the delay in rush-hour is much higher than the other two cases due to severe interference. Actually, this does *not* lead to a low system performance, since according to our proposed system model only cluster head interacts with the WSN. As shown before, cluster arrival rate in rush-hour traffic is less than 1, which means the VANET in rush-hour traffic is almost always connected. The warning propagation can be always conducted via the VANET rather than the WSN.

Therefore, *the utilization of sensor nodes is inversely proportional to the traffic (or cluster) density. The energy consumption of both APs and regular nodes in the rush-hour traffic scenario is the lowest while that in the night traffic scenario is the highest.*

VI. Conclusion

An integrated VANET-WSN system was proposed in this paper to overcome the inherent limitations of pure VANET-based system. Protocols were designed for efficient vehicle-sensor and sensor-sensor interactions. Prototype of the system has been implemented and tested in the field to verify its feasibility. The simulation results indicate that, with appropriately chosen system parameters, satisfactory safety and energy efficiency can be achieved simultaneously.

Acknowledgements

This work was partially supported by the National Science Foundation under Grands CNS-0834593 and CNS-0834585.

References

[1] CROSSBOW TECHNOLOGY INC., "WSN," *http://www.xbow.com/*, web Link.

[2] E. LLC, "Wieye - sensor board for wireless surveillance applications," *http://www.easysen.com/WiEye.htm*, web Link.

[3] E. Weingatner and F. Kargl, "A Prototype Study on Hybrid Sensor-Vehicular Networks," *GI/ITG KuVS Fachgesprach "Wireless Sensor Networks"*, 2007.

[4] J. Bohli, A. Hessler, O. Ugus, and D. Westhoff, "A secure and resilient wsn roadside architecture for intelligent transport systems," *WiSec'08*.

[5] U. Lee, B. Zhou, M. Gerla, E. Magistretti, P. Bellavista, and A. Corradi, "Mobeyes: smart mobs for urban monitoring with a vehicular sensor network," *Wireless Communications, IEEE*, Oct. 2006.

[6] State Farm, "State Farm Statistics," *http://www.philadelphia-accident-lawyers.com/auto-car-accidents/deer-road-safety.html*, 2005, web Link.

[7] I. Leontiadis, P. Costa, and C. Mascolo, "Persistent Content-based Information Dissemination in Hybrid Vehicular Networks," *PerCom '09*.

[8] P. Costa, D. Frey, M. Migliavacca, , and L. Mottola, "Towards Lightweight Information Dissemination in Inter-Vehicular Networks," *VANET '06*.

[9] O. Tonguz, N. Wisitpongphan, F. Bai, P. Mudalige, and V. Sadekar, "Broadcasting in VANET," *INFOCOM '08*.

[10] W. Z. Song, R. Huang, B. Shirazi, and R. LaHusen, "TreeMAC: Localized TDMA MAC Protocol for Real-time High-data-rate Sensor Networks," *PerCom '09*.

[11] G. S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo, "Funneling-mac: A localized, sink-oriented mac for boosting fidelity in sensor networks," *SenSys '06*.

[12] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: a hybrid mac for wireless sensor networks," *SenSys '05*.

[13] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "High-throughput path metric for multi-hop wireless routing," *MobiCom '03*.

[14] Packet Data Systems Ltd, *www.pds-test.co.uk*, web Link.

[15] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, "Routing in Sparse Vehicular Ad Hoc Wireless Networks," *JSAC*, 2007.