STEID: A Protocol for Emergency Information Dissemination in Vehicular Networks*

Josiane Nzouonta and Cristian Borcea
Department of Computer Science
New Jersey Institute of Technology
{jn62, borcea}@njit.edu

***Abstract*— This paper presents STEID, a spatio-temporal emergency information dissemination protocol for vehicular networks. Its main goal is to quickly disseminate traffic alerts about accidents or congestions to every car that passes through an emergency zone during the lifetime of the emergency. To achieve this goal, we propose a hybrid network architecture consisting of WiFi clusters connected through proxy servers and cellular links. STEID executes on top of this architecture and ensures the emergency message delivery to all the intended vehicles in a short time interval. Simulations conducted with hundreds of vehicles moving at speeds between 45mph and 70mph demonstrate this fact. We also compare STEID to a solution using only cellular communication with a central server for information dissemination. The results indicate that the average delay for STEID is between two and four times less than the one for cellular-only approach for regular highway speeds, while the traffic load imposed on the cellular network decreases by more than 65%.**

I. Introduction

In recent years, most new vehicles come equipped with GPS receivers and navigation systems. Car makers such as Ford, GM, and BMW have already announced efforts to include significant computing power inside their cars [2], [13]. We envision that in the near future the number of vehicles equipped with computing technologies and wireless network interfaces will increase dramatically. These vehicles will be capable to run network protocols that will exchange messages for safer and more fluid traffic on the roads.

Avoiding accidents and traffic jams are two main immediate benefits of vehicular networks. For instance, most drivers would like to receive real-time alerts about accidents happening at a short distance in front of their vehicles at night, in the fog, or heavy rain. This type of accidents could lead to collision chains involving tens of vehicles. Although not life-threatening, traffic jams are huge nuisances. The ability to receive an alert about a potential traffic jam would allow drivers to take alternate routes, saving both time and fuel.

The problem addressed in this paper is: How to ensure that a traffic emergency message is disseminated with high probability and low delay to all the potentially affected vehicles? If we assume that participating vehicles (not all moving vehicles) are equipped with an embedded computer, a GPS receiver, navigation software, and one or multiple wireless network interfaces, the problem can be reduced to designing a reliable and efficient safety information dissemination protocol in vehicular networks.

To understand the requirements for such a protocol, let us consider the scenario illustrated in Figure 1. When an accident or other event (e.g., road work) susceptible to lead to traffic congestion happens, a traffic alert message is generated. Such a message is either triggered automatically by a car involved in an accident (e.g., a sensor for inflated airbags could generate an event propagated to the embedded computer through the on-board diagnostic system interface), by a person (driver, road worker, or police officer) or by a system capable of detecting/predicting traffic jams. Such an alert is characterized by:

- the *location* of the emergency, defined as a pair of road and relative position of the emergency on that road.
- the emergency *zone* or alert region, defined as all the roads leading to the emergency location.
- the *lifetime* of the emergency, defined as the time needed to return to regular traffic conditions after an emergency.

These characteristics are determined by applications running on vehicles. Let τ_{max} be the *maximum safest delay* in delivering the message to a vehicle. This value could be the average time it would take a vehicle to move from the entry of the alert zone to the nearest exit (to avoid traffic congestion). If the dissemination protocol ensures that every vehicle already in the zone or every new vehicle entering the zone during the emergency lifetime receives the alert within τ_{max} seconds, we say that the protocol provides both spatial and temporal reliability as defined below.

Spatial Reliability: A protocol provides spatial reliability if the alert is delivered to all participating vehicles within the emergency zone.

Temporal Reliability: A protocol provides temporal reliability if a participating vehicle does not spend more than τ_{max} time within the emergency zone before receiving the alert.

For instance, spatial reliability means that all vehicles present in the alert region in Figure 1.a must receive the alert. Figure 1.b helps us to understand temporal reliability; vehicles G, I, and H, which entered the zone after the accident happened, must receive the alert before reaching the nearest exit.

This paper presents the Spatio-Temporal Information Dissemination protocol (STEID), a protocol that addresses both the spatial reliability and the temporal reliability requirements. STEID ensures timely delivery of alert messages to all the

*This work is supported in part by the NSF grant CNS-0520033.

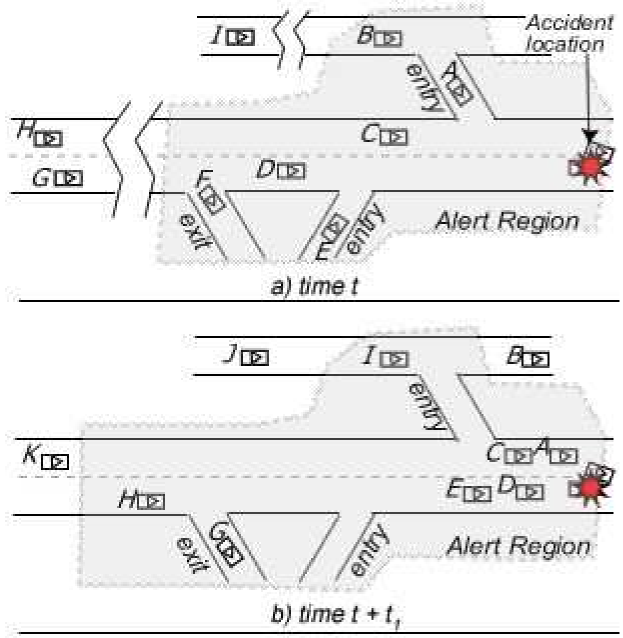

Fig. 1. An Accident occurs on the road. Which vehicles should receive an alert about the accident? How fast should they receive it? For how long should the alert remain active?

intended recipients in the alert zone throughout the lifetime of the emergency. STEID executes on top of a hybrid vehicular network architecture consisting of IEEE 802.11 clusters connected through proxy servers and cellular links. Additionally, it involves a data synchronization mechanism between nearby vehicles to improve the dissemination reliability. Simulations conducted with hundreds of vehicles moving at speeds between 45mph and 70mph demonstrate that STEID achieves spatial reliability and temporal reliability. We compare STEID to a solution using only cellular communication with a central server for information dissemination. The results indicate that STEID delivers the messages faster, while reducing the traffic load on the cellular network by more than 65%.

The rest of this paper is organized as follows. Section II presents the existing solutions for emergency dissemination in vehicular networks. Section III describes STEID, the proposed emergency information dissemination protocol. In Section IV, we show through simulation that STEID delivers the emergency messages to all the intended cars in a very short time interval. The paper concludes in Section V.

II. Background

This section provides an overview of existing solutions for disseminating safety-related information in vehicular networks. We classify these solutions by the method of handling message propagation.

A. Vehicle to Vehicle Propagation

In this approach, vehicles form ad hoc networks using short-range wireless network interfaces. One of the problems associated with this solution is broadcast storm, which can decrease significantly the message delivery rate. Various heuristics have been proposed to coordinate the rebroadcasting of the message. RBM [5] advocates broadcasting the alert only when the vehicle comes in the transmission range of a new neighbor. Additionally, a maximum number of hops is appended to the message to prevent it from being forwarded indefinitely. DDT [11] delays the rebroadcasting for a time inversely proportional to the distance from the sending vehicle in order to avoid some retransmissions. IVG [1] limits rebroadcasting to a given region. Other solutions to improve the delivery rate and the message delay include [14] and [6].

These solutions improve the transmission success rate when the vehicular network is not partitioned. However, if we consider the road structure presented in figure 1, we observe that network partitions prevent vehicles C and E from communicating with H and I. Thus, G, H, and I do not receive the alert before reaching the exit road if the propagation protocol uses only vehicle to vehicle communication.

B. Vehicle to Roadside Base-Station Propagation

In [10], [4] roadside base stations are used to bridge network partitions in vehicular networks. A car already informed of an accident forwards the alert when passing by a roadside base-station. Subsequently, the base-station forwards the message to other base-stations located in the alert zone. Each of the informed stations periodically broadcasts the alert to inform passing vehicles. As the authors pointed out, this method is heavily dependent on the distance from the accident site to the closest base station. Although this approach could be useful for metropolitan scenarios, it may prove costly for a general use as it would require the installation of base-stations on all roads.

C. Server-Based Propagation

In [7], each car communicates with a server which forwards the emergency information to all vehicles in the alert zone. The server can be reached either through ad hoc multi-hop communication or through satellite communication as in [3]. We note that when the server is accessed through multi-hop communication, the method is similar to the vehicle to roadside base station propagation and therefore presents the same limitations. To have each car communicating by satellite with a server would be costly and not scalable. The same problems appear in a cellular-only approach, such as the one used for comparison in this paper, where the vehicles exchange alerts by connecting to an Internet server over cellular links.

III. STEID Design

STEID is a spatio-temporal information dissemination protocol that leverages existing approaches to create a solution that addresses the issues mentioned in the previous section. This protocol satisfies both the spatial and temporal reliability requirements by ensuring the delivery of traffic alerts in a short time, to all vehicles that pass through an emergency zone during the lifetime of an emergency.

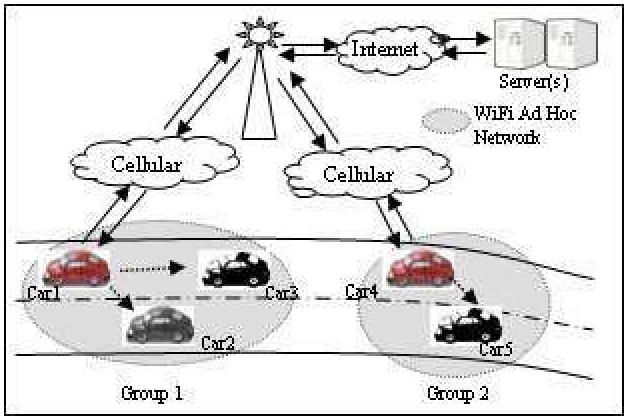

Fig. 2. Hybrid Vehicular Network Architecture

A. Hybrid Network Architecture

STEID works over a hybrid vehicular network architecture, as shown in Figure 2. This architecture consists of WiFi clusters connected through proxy servers and cellular links. In this paper, a cluster is a connected network partition formed in ad hoc fashion. Note that vehicles moving on the same road and in the same direction can maintain a cluster structure for relatively long periods of time despite mobility within the cluster. In practice, each vehicle features two wireless network interfaces: short range (e.g., 802.11a, DSRC, 802.11n) and cellular (e.g., GPRS, EVDO). The general idea of this architecture is to use short range communication to transfer the bulk of data within clusters in an inexpensive and scalable way, while cellular communication is used to improve communication reliability when the network is partitioned in multiple clusters. Data transfers between clusters are mediated by one or multiple proxy servers. Each cluster elects a *cluster head* as the point of contact for communication with a proxy server.

1) Cluster Management: To form a cluster, each car on the road periodically advertises its presence by broadcasting a *Hello* message using the WiFi interface. This message includes the current road, relative position, and moving direction of the sender. Upon receiving a Hello message, each car adds an entry for the sender in its one-hop neighbor list if the sender is on the same road and moving in the same direction. This entry is purged, unless refreshed periodically.

After the neighbor discovery phase, a loosely coupled cluster is formed, and this cluster proceeds to elect a cluster head. The election is based on the relative location of the vehicles in the cluster. Due to space limitations, we skip this simple distributed algorithm. The only task of a cluster head is to provide a communication end-point for inter-cluster communication. Once elected, it determines the proxy server responsible for its region and registers with this server: it sends its address and position on the road, and it receives a unique *cluster-id*. The cluster head propagates this cluster-id and the server address to all nodes in the cluster using a cluster update message. If a vehicle does not receive an update from its cluster head for two update periods, it will trigger a new cluster head election.

2) Inter-Cluster Communication: A cluster head acts as a gateway that receives messages from other clusters (through proxy servers) and sends them to vehicles within its cluster. However, it does not act as a gateway for uplink traffic. Individual nodes connect to their associated server when they need to send data to other clusters (the address of the server is shared across the cluster). Thus, the cluster head does not become a bottleneck for uplink traffic.

Each proxy server maintains a dynamic structure of its registered clusters. If the cluster head does not update its position for a fixed time period, its entry is removed. Therefore, cluster heads send periodic keep-alive messages to update their position with the servers. For improved scalability, a cluster head downloads all the emergency messages active in its region after sending the keep-alive messages (Section III-B explains how the emergency messages are uploaded at the proxy servers). This operation results in improved scalability as TCP connections are not continuously maintained between servers and cluster heads.

B. Emergency Dissemination

Once generated, an emergency message is broadcasted over the WiFi interface for dissemination within the local cluster and unicasted to the server over the cellular link. The servers can be programmed to inform the police each time they receive an alert about a new accident. Every vehicle keeps a list containing all the active emergency messages. When a vehicle receives a message, it checks its list to see whether to store or discard this message. If this is a new message, the vehicle stores it and computes the time, inversely proportional to the distance to the sender, when it should rebroadcast the message.

The server maintains a similar emergency messages list. As mentioned above, the cluster heads synchronize their lists with the one maintained at the server after each position update with the server. Once a new message is downloaded from the server, the cluster head starts its dissemination within the cluster. The server as well as the vehicles purge the active emergency lists periodically to ensure that only active emergencies are kept.

STEID uses a field in the header of the periodic Hello messages to hold the number of active emergencies known by each vehicle. This prevents the alert from continuously being re-broadcasted by each node, but rather to be unicasted on demand when two neighboring vehicles synchronize their emergencies lists. This mechanism is described in more details below.

C. Alert Synchronization

Since IEEE 802.11 is subject to contentions and fading, which may prevent a car from receiving a message broadcasted in the cluster, we optimized STEID to deliver, with high probability, the emergency messages to every node that spent at most three consecutive update periods (i.e., periods between Hello messages) in the emergency zone. Our protocol adds two more pieces of information in the Hello messages: the count of active alerts in the region, and the position of the closest active alert. Using these data, we designed an algorithm for alert synchronization among neighbor nodes (Algorithm 1).

Algorithm 1 Synchronization algorithm

1: $Missed\_updates \Leftarrow 0$
2: SUB on_Hello_Send
3: broadcast $Hello$ message
4: $Missed\_updates \Leftarrow Missed\_updates + 1$
5: **if** $Missed\_updates > 2$ **then**
6: start cluster head election
7: **end if**
8: SUB $on\_Hello\_Receive$
9: **if** sender moving in same direction **then**
10: update sender information in neighbor list
11: **if** timestamp higher than current timestamp **then**
12: $Missed\_updates \Leftarrow 0$
13: **end if**
14: **end if**
15: **if** different active emergency count or different nearest active emergency position **then**
16: synchronize emergency list with sender
17: **end if**

The procedure on_Hello_Send (lines 2-7) runs every update period, whereas $on\_Hello\_Receive$ (lines 8-17) runs each time a new $Hello$ message is received.

Upon synchronizing with the server, the cluster head receives an active emergency message for the area and rebroadcasts it over WiFi. Let us assume that due to high packet loss, one node does not receive the message. There are two cases that have to be considered. First, the node continues to broadcast its position (line 3), but does not receive a message with a newer server timestamp from a neighbor. The variable $Missed\_updates$ is used to count the number of update periods elapsed since receiving the last message with a higher server timestamp. After two missed updates, the node starts a new cluster election (lines 5-6). During this process, it will receive the potentially missed alert message from the server.

Second, if the node receives a Hello message with a newer server timestamp from a node moving in the same direction, it resets the $Missed\_updates$ variable (lines 9-14). Then, independent of the direction of movement of the source node, it verifies if it has to synchronize its alert list with the source node. The synchronization (lines 15-16) happens in two cases: (i) the nodes have different counts for the number of active emergencies, or (ii) the nodes have the same count, but the position of the closest alert differs.

IV. Performance Evaluation

In this section, we present the evaluation of STEID using the network simulator NS-2 version 2.29 [12]. We also present a comparison between STEID and a simple alternative (called cellular-only) using only cellular communication between cars and a central server. Our goal is to show that compared with the cellular-only solution, STEID provides better responsiveness and scalability.

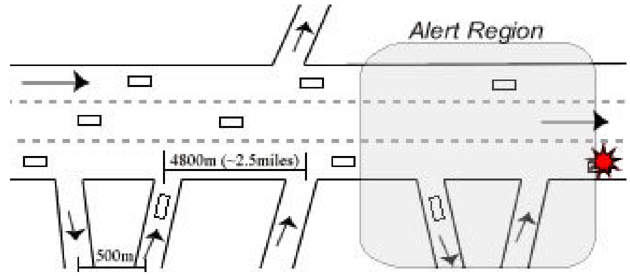

Fig. 3. Simulation Road Structure

A. Simulation Methodology

We evaluate the performances of both protocols on a sparse 3-lanes highway of 15km (Figure 3). In this scenario all vehicles move in the same direction. Highway entry-ramps and exit-ramps are evenly distributed on the road. Cars are inserted during the simulation at a rate of 9 vehicles/km/lane for the highway, and 6 vehicles/km/lane for entry ramps. When approaching an exit-ramp, a node has a probability of 0.3 to take the exit. We implemented the Gipps car-following and lane-changing model [8], [9] to adjust each car's speed with its predecessor and determine the logic followed by a car when changing lanes. This model provides the ability for a smooth transition between acceleration and deceleration of speed for each car. At each moment the maximum safest speed of a vehicle is given by:

$$V_n(t+T) = b_n * T + [b_n^2 * T^2 - b_n * (2 * (X_{n-1}(t) - S_{n-1} - X_n(t)) - V_n(t) * T - (V_{n-1}^2(t)/b'))]^{\frac{1}{2}}$$

where n = current vehicle, $n-1$ = preceding vehicle
T = time interval ($= 1sec$)
$V_n(t)$ = speed of vehicle n at time t
b_n = maximum braking car n is willing to do ($= -2.5m/s^2$)
$X_{n-1}(t)$ = location of the front bumper of car $n-1$ at time t
S_{n-1} = length of vehicle $n-1$
b' = estimate of maximum braking car $n-1$ is willing to do

We used the IEEE 802.11 Distributed Coordination Function (DCF) protocol at the MAC layer, a data rate of 11Mbps, a transmission range of 250m, and the Shadowing radio propagation model. This propagation model accounts for multipath propagation effects by introducing unpredictability in data transmission of nodes whose distance is close to the maximum transmission range. We simulated the cellular link at each node through a secondary 802.11 interface with parameters comparable to those of GPRS (e.g., round-trip time of 700ms).

For each simulation, the speeds of the cars are initially uniformly distributed within $[v_{avg} - \gamma,\ v_{avg} + \gamma]$, where $\gamma = 5m/s$ (~11.5mph). Each simulation runs for 1000s, and the emergency message is generated after a 300s warm-up period. The emergency lifetime is 600s and its zone is an area stretching 5600m from the accident, with the nearest exit being at 4000m from the accident (Figure 3). The metrics used in the simulation are:

Message reception delay. This metric measures the protocol responsiveness (i.e., the amount of time a car has to wait in the alert zone before receiving the alert message).

Timely delivery rate. This metric quantifies the number of

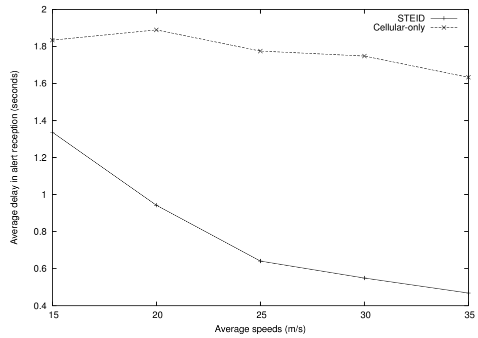

Fig. 4. Average time spent by cars in the zone before receiving the alert

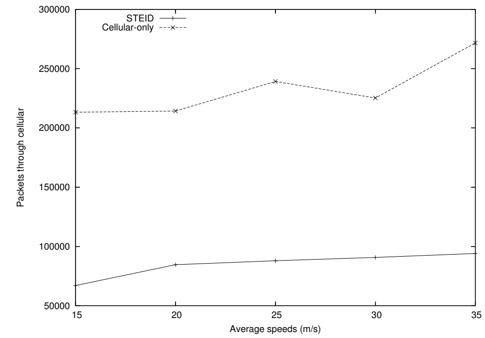

Fig. 5. Traffic load on cellular network

cars which received the message within $\tau_{max} = 10sec$ of entering the emergency zone as a percentage of the total number of cars entering the zone for the lifetime of the emergency ($10sec$ corresponds to three times the update period in the implementation of STEID).

Cellular network load. Given that cellular communication incurs cost and has a limited bandwidth compared to WiFi, this metric is used to measure the number of packets transferred through cellular network for each solution.

B. Simulation Results

Figure 4 compares the average delay in receiving the message during the entire lifetime of the alert for different average speeds. The average waiting time for STEID goes below 1s as STEID uses a hybrid WiFi-cellular architecture which allows cars to be informed even before entering the zone, if the car happens to belong to a cluster that has cars in the zone. STEID has a smaller average delay in reception because multiple nodes learn about an emergency from a broadcasted message. Furthermore, broadcasts can happen simultaneously in different clusters. The cellular-only solution, on the other hand, uses sequential unicasts, and these unicasts happen only after the nodes update their position from within the zone.

Similar to the cellular-only solution STEID achieves 100% delivery rate if we do not consider a time limit for receiving the message. When using the timely delivery rate metric (graph not included), we observed two situations for STEID. For speeds over 20m/s (∼45mph), our results indicate a 100% timely delivery rate. However, for lower speeds, a small number ($<$ 0.3%) of the cars did not receive the message within the $10s$ limit (these cars received the message in a time between 10s and 14s after entering the zone). The reason is the effect of large clusters (i.e., clusters that expand over a relatively long road stretch) coupled with the fact that the simulated road is a one-directional. For instance, it could happen that the cluster head enters the zone after our limit of 10s when the average speed is low. Vehicles in front of it that are already in the zone will receive the alert after the cluster head enters the zone. To solve this problem, we plan to investigate a solution that will limit the distance covered by a cluster.

Since cellular communication incurs a cost and its bandwidth is limited, we tried to limit the number of packets transferred through the cellular network. Figure 5 plots the traffic load on this network as function of average speed. We observe more than 65% less traffic for STEID. This result also translates into a lesser load on the server. The difference between the two protocols simply represents the part of the traffic moved on the IEEE 802.11 wireless network, which is free and has a significantly higher bandwidth.

V. Conclusions

This paper has presented STEID, a spatio-temporal emergency information dissemination protocol for vehicular networks. This protocol works over a hybrid network architecture composed of WiFi-based clusters inter-connected by cellular links and proxy-servers. STEID achieves over 99.7% timely delivery rate of alert messages for all the intended vehicles. In addition, the dissemination is between two and four times faster compared to a cellular-only protocol, while reducing the traffic load on the cellular network by more than 65%.

References

[1] A. Bachir, A. Benslimane. A multicast protocol in ad hoc networks inter-vehicle geocast. VTC Spring 2003.

[2] Cars get PCs despite safety concerns. http://www.post-gazette.com/pg/06026/644908.stm.

[3] T. Imielinski and J. C. Navas. Gps-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, 42(4):86–92, April-June 1999.

[4] K. Kutzner, J.-J. Tchouto, M. Bechler, L. Wolf, B. Bochow, and T. Luckenbach. Connecting vehicle scatternets by internet-connected gateways. Workshop on Multiradio Multimedia Communications MMC, 2003.

[5] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. MobiHOC, August 2000.

[6] T. Little and A. Agarwal. An information propagation scheme for vanets. ITSC 2005.

[7] C. Maihöfer, T. Leinmüller, and E. Schoch. Abiding Geocast: Time-stable Geocast for Ad Hoc Networks. In *Proc. VANET*, 2005.

[8] P. G. Gipps. A behavioural car-following model for computer simulation. Transportation Research B, 15: 105–111, 1981.

[9] P. G. Gipps. A model for the structure of lane-changing decisions. Transportation Research B, 20B(5):403-414, 1986.

[10] R. Mussa, J. Upchurch. Simulator Evaluation of Incident Detection using Vehicle to Roadside Communications (VRC) System. In *Procs. IEEE Intelligent Vehicles Symposium 2000*, 2000.

[11] M.-T. Sun and al. Gps-based message broadcast for adaptive inter-vehicle communications. VTC Fall 2000.

[12] The network simulator: NS-2. http://www.isi.edu/nsnam/ns.

[13] Wireless: The new backseat driver? http://news.com.com/2100-11389-5933641.html?tag=tb.

[14] Q. Xu, T. Mak, J. Ko, and R. Sengupta. Vehicle-to-Vehicle Safety Messaging in DSRC. In *Proc. VANET*, 2004.