Using Ad-hoc Inter-vehicle Networks For Regional Alerts

Qixiang Sun and Hector Garcia-Molina
Computer Science Department
Stanford University
{qsun, hector}@cs.stanford.edu

Abstract

Ad-hoc inter-vehicle networks will soon be a reality as cars become equipped with wireless communication system. One use of an inter-vehicle network is to propagate alerts such as accidents and road conditions within a region. Unlike previous work in the area that focuses on instantaneous *delivery of an alert to all reachable cars, this work studies the problem where an alert needs to be maintained for a* duration *of time. In other words, we must also notify cars that become reachable after the alert begins. Maintaining an alert for a duration is important because other cars can then take precautions or change their travel path to avoid the condition. Moreover, we do not require the original initiator of an alert to be stationary and constantly repeating the alert. In this paper, we formally define the problem and its correctness. We provide an efficient protocol that minimizes the number of broadcasts needed for maintaining a regional alert over a period of time, and we evaluate our protocol through simulation.*

1 Introduction

In recent years, car manufacturers like BMW, Daimler-Chrysler, and Toyota have included global positioning system (GPS), map service, and IEEE 802.11 wireless communication system in their upcoming commercial vehicle designs. Thus the future of an ad-hoc inter-vehicle network will soon be upon us. From consumers' perspective, we want these new high-tech additions in our cars to improve our driving safety and experience.

In this paper, we focus on one such application: a regional alert system (RAS) that warns us about road and traffic conditions ahead of us. For example, consider the scenario depicted in Figure 1. Suppose car X has just driven over a bridge and discovered a patch of black ice on the surface. Then X should automatically notify other cars via wireless communication so that they are aware of the condition before moving within the *safety* radius. Moreover,

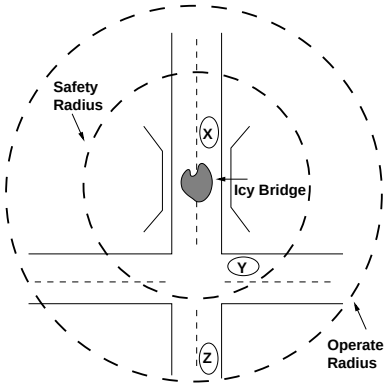

Figure 1. A scenario of regional alert.

we want this icy-bridge alert to remain in effect so that new cars, e.g. car Z, are also notified before entering the safety radius. Thus even when X leaves the region, someone else, e.g. car Y in Figure 1, should continue to propagate the alert. Of course the alert is not propagated "infinitely far." In Figure 1, there is an operating radius beyond which no cars will disseminate the alert.

Informally, the regional alert problem can be stated as follows: given an alert with a location, a time duration, and the safety and operating radius, if feasible, all cars traveling through the alert region during the time of the alert should be notified before breaching the safety radius. Precise description of the problem and assumptions are given in Section 3. As seen from the example in Figure 1, RAS is useful for disseminating information like road conditions, accidents, congestion, road repairs, detours, etc.. The key characteristics of a RAS are:

1. No association between senders and an alert. An alert is associated with a location rather than a particular sender or car. There does not exist an "owner" of an alert. There is, however, an originator of an alert who first detects and propagates the alert condition.
2. No stationary "repeater" at the origin of the alert. In other words, the originator of an alert does not remain at the site of the alert to continuously relay the alert.

Unlike accidents where a disabled car may function as a repeater, road condition alerts originate from passing cars, thus unreasonable to assume a repeater at the origin.

3. No pre-determined set of receivers. Receiving cars are determined by their location with respect to an alert. In other words, the set of receivers is highly dynamic.
4. A time duration for the alert. When an alert occurs, instantaneous delivery to cars in the affected region is not sufficient. One must continuously inform other cars coming into the region.
5. Many cars are expected to enter and leave the alert region during the alert duration.

These characteristics require a solution that is more than just the traditional flooding or store-and-forward scheme in ad-hoc and mobile networking. Any RAS solution must address both the geographical constraint and the time duration constraint of an alert. Instead of the traditional problem of routing a message *instantly* via an ad-hoc network to a *specific* client or group of clients, RAS must route an alert to *all* clients in a *region* for a *duration*, even if the underlying ad-hoc network changes as cars enter and leave the region.

In this paper, we study how to build such a regional alert system by only relaying alerts between cars using wireless communication, i.e., an ad-hoc inter-vehicle networks. We also answer the question on whether we can guarantee if an alert can be propagated to "all" affected cars. We choose this ad-hoc approach because cars will be equipped for both sending and receiving data, thus making it easy and cheap to deploy an inter-vehicle solution. [1]

One simple solution for building a regional alert system is to have vehicles that know about an alert periodically rebroadcast while the alert is still active. Although this solution can provide all the desired functionality of a RAS, the operating overhead is high because many periodic broadcasts are wasted in that they do not reach any new cars. Thus we want a solution that minimizes the number of broadcasts needed in maintaining the alerts.

Our approach, the *Bidirectional Perimeter-based Propagation* (BiPP), provides an elegant solution for building RAS using ad-hoc inter-vehicle networks by exploiting one crucial observation — cars can only enter the alert region if they cross the boundary or the "perimeter" of the alert region. This perimeter is typically maintained between the safety and the operating radii. Thus instead of having periodic broadcasts *throughout* the entire region, it is sufficient to broadcast "near the perimeter." Since cars must travel on existing roads, broadcasting "near the perimeter" means broadcasting at locations where roads cross the perimeter. For reasonably sized alert region, one might expect only a few broadcasting locations near the the perimeter.

Although intuitive, there are many challenges in creating a perimeter-based protocol. For example, the perimeter shrinks and grows dynamically as cars enter, move around, and leave the alert region. To illustrate, a car on the perimeter may move outside of the operating area, thus causing the perimeter to suddenly shrink. When such a shrinkage occurs, a car previously not on the perimeter has to, all of a sudden, detect that it is now on the perimeter and begin broadcasting. If one is not able to detect and cope with these changes correctly, there will be traffic patterns where cars are not notified about the alert.

In this paper, we describe our BiPP protocol and demonstrate that it is efficient and provably "correct." Our key contributions are

- A simplified model and a formal characterization of what it means to guarantee delivery of an alert to "all" affected cars in an alert region.
- BiPP, a protocol that uses cars traveling in opposite directions to reduce broadcasting overhead and guarantee alert delivery.
- A demonstration, via simulation, that our protocol has very low overhead in the number of broadcasts.

As a disclaimer, for this work we use a simple model that operates at the message level, i.e., we are not modeling packet level details like collisions, packet transmission time, or exponential back-off timers. Despite the simplistic model, our work still offers valuable insight into constructing a practical regional alert system.

The remainder of the paper is organized as follows. Section 2 provides a high level overview of the BiPP protocol and how it relates to other work. Section 3 gives our model and defines delivery correctness. Section 4 then describes in detail how BiPP operates. Section 5 shows some simulation results. We conclude in Section 6.

2 Overview

In this section, we informally describe BiPP through a few examples on a single two-way road. Consider the scenario depicted in Figure 2(a) where two cars U and V are moving towards the alert on the right. In this example, car U already knows about the alert, indicated by a rectangular box, while car V does not, indicated by a round oval.

In order to propagate the alert further to the left, car U has to periodically broadcast the alert, hoping that car V eventually is in communication range before V reaches the safety radius. In Figure 2, we use a shaded box to indicate that car U is broadcasting. Note that car U has to broadcast

[1] One can build a regional alert system using additional infrastructure like cellular towers. We do not suggest that an ad-hoc approach is better or worse than an infrastructure-based approach. While an ad-hoc approach has many technical challenges, an infrastructure-based approach has to deal with standardization, deployment, servicing, and pricing issues.

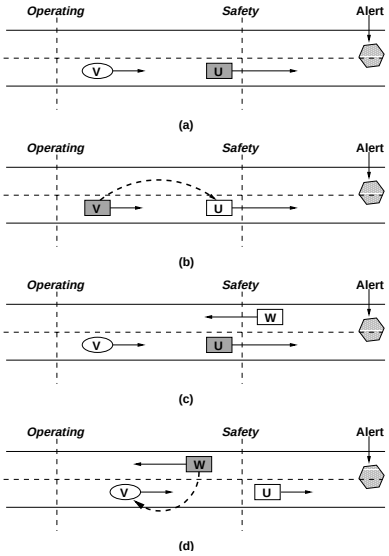

Figure 2. Example of alert propagation

"very frequently" because it does not know whether there is a car V behind it or when car V would be in communication range. If U broadcasts only once in a while, then it is possible that car V may creep into and out of communication range by quickly accelerating and then decelerating between successive broadcasts by U.

If car V is in range to receive U's broadcast, then V can realize, by consulting its GPS coordinate and maps, that it is further to the left of the alert than U. Therefore, V is more suited to propagate the alert to the left than U. As a result, V would begin to broadcast as shown in Figure 2(b). Now if car U receives V's broadcast, then by the same logic that V is more suited, car U will stop broadcasting, depicted by U changing from a shaded box to a clear box in Figure 2(b). Thus from that moment on, car V "takes over" the broadcasting responsibility from car U.

The examples in Figure 2(a) and 2(b) illustrate a fundamental limitation on how well we can propagate an alert if there are no traffic in the opposite direction. When cars U and V are out of communication range, commonly known as *fragmentation*, it is impossible to propagate an alert. On the other hand, if they are in range, then only the left-most car, car V in this example, will actively broadcast the alert. Car V is said to be *on the perimeter* and is responsible for propagating the alert further.

When there is traffic in the opposite direction, as in Figure 2(c), BiPP takes advantage of the traffic to alleviate the fragmentation problem discussed previously. Moreover, the periodic broadcast can be much less frequent without sacrificing guarantees on reaching as many cars as feasible. To illustrate, consider car W in Figure 2(c). Initially, car W is not broadcasting because U is further to the left. When W eventually "passes" U as in Figure 2(d), car W takes over the broadcasting responsibility. Obviously the fragmentation problem is solved because car V would eventually be notified by W when they "pass" each other.

Unlike U which has to broadcast frequently because another car may sneak into and out of communication range quickly, car W can be less aggressive in broadcasting, i.e., avoiding unnecessary broadcasts. For instance, to guarantee that V hears about the alert, W only has to broadcast frequent enough so that V does not move into W's range, continue to pass W, and leave W's range between W's successive broadcasts. This time interval is much larger than two cars traveling in the same direction that creep into each other's range momentarily; hence using cars in the opposite direction leads to a much more efficient protocol.

There are, however, many issues with cars traveling in the opposite direction. As alluded to in the introduction, when car W in Figure 2(d) eventually leaves the operating radius, car V has to "take over" the broadcasting. Moreover, car W is only useful because it was leaving the area. In Section 4, we give details on when and how we can effectively use cars in opposite direction while guaranteeing an alert is propagated to all "reachable" cars. We also discuss how intersections are handled.

2.1 Related Work

The three most relevant papers on disseminating alerts are Role-based Multicast (RBM)[4], TRADE [6], and Inter-Vehicle Geocast (IVG)[1, 2]. Our work differ from this previous work in three important aspects:

1. we do not assume a stationary repeater at the alert location and handle a time duration for an alert,
2. we use cars leaving the alert area to efficiently disseminate an alert,
3. we guarantee to propagate an alert to all "reachable" cars.

RBM, TRADE, and IVG only use cars moving towards the alert, thus suffering from the fragmentation problem mentioned previously. The three schemes differ in how they address the fragmentation problem. In RBM, they delay relaying broadcasts, as opposed to flooding immediately after the alert begins. They also use a time-to-live counter for their alerts rather than an active time duration for an alert. TRADE and IVG use a similar technique of maintaining broadcasts near the perimeter to address the problem. They do not, however, have a clean notion of safety radius and operating radius.

Aside from propagating an alert as "far" as possible, there is also the issue of multiple cars in close vicinity receiving the same broadcast and rebroadcasting simultaneously, i.e., a broadcast storm problem[13]. To solve this simultaneous rebroadcasting problem, the Distance Delayed

Time (DDT) [6] mechanism is used. In DDT, after receiving a broadcast from a sender, one set a time-out before rebroadcasting that is inversely proportional to the distance to the sender. In other words, farther away cars will rebroadcast first, thus suppressing nearby cars from rebroadcasting at all. This DDT technique can also be used in our work, although we do not address it specifically.

Maintaining alerts is also similar to various flavors of ad-hoc multicast [3, 10, 11, 17, 12, 9] because one can treat all cars needing an alert as a multicast group. Most of these multicasts, however, build a tree and rely on the traditional unicast routing[15, 8, 14]. For cars on the road where the ad-hoc network is never stable, a different type of routing technique, like interest-based, is more appropriate. For example, content-based multicast (CBM)[18] and direction diffusion [7] both use application-level semantics (or interests) in the routing. Although we focus on the application level, other work such as CarTalk [5] address technical issues at the physical, data-link, and network layers.

3 Model, Assumptions, and Definitions

We discretize time and location to create a simple model for RAS. For simplicity, we will focus on handling a *single* active alert for the remainder of the paper. As a result, our model contains the following components:

1. A global map known by every vehicle. We discretize the map on a 2D grid, and model it as a graph $G = (V, E)$. The set of nodes V are all locations with integer coordinates on the grid. These nodes are then connected with edges that represent road connectivity. For simplicity, cars can only reside at these node locations and move between connected nodes. Figure 3 shows an example of two parallel roads and one intersecting perpendicular road. The distance $D(x, y)$ between two points x and y on this map G is then simply the hop count (number of edges) in the shortest path from node x to node y in G.
2. Cars and their trajectory. We model each car's trajectory as a set of pairs $\langle location, time \rangle$. To model car's movement, at each time step, a car may either stay at its current location or move to an adjacent grid points.
3. A single alert (as a simplification for ease of discussion). We represent the alert as a tuple of the form $\langle location, start\_time, duration, safety, operate \rangle$. The *start_time* and *duration* fields indicate when the alert is active and for how long. The *safety* field gives the desired radius of the alert. More specifically, when a car moves to within the *safety* distance of an alert, it should have "heard" about this alert. The *operate* radius specifies when a car will stop participating in disseminating the alert.

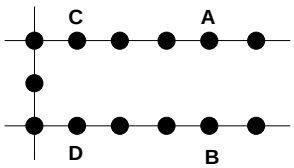

Figure 3. Example map and communication.

4. There is a source car S who initiates the alert. For example, if the alert is an accident, the source car can either be the car(s) involved in the accident or a car passing by at the time of the accident.

For wireless communication, we make the following simplifying assumptions:

- Two cars can communicate wirelessly if their hop count distance on the map G is less than or equal to some communication range W. Note, this assumption disallows two cars on two unconnected parallel roads from communicating.[2] To illustrate, consider the road map depicted in Figure 3. When the communication range W is 4 grid points, despite the fact that A and B are located only 2 grid points apart, they can not communicate with each other because there is no path of at most 4 hops between them on the map. On the other hand, C and D can communicate with each other.
- All cars broadcast omni-directionally and have the same communication range W.
- A car can broadcast up to one message per time step.
- Each car, in a single time step, receives all messages broadcasted within range.
- We *do not* model message loss; i.e., we do not model signal interference, retransmissions, etc..
- No implicit message acknowledgment of wireless broadcasts. In other words, a car will not know if its broadcast is received by anyone.

3.1 Reachability and On-time

In a regional alert system (RAS), there are two important concepts: *reachability* and *on-time*. We give informal definitions here. For a rigorous treatment based on our model, see the extended technical report[16].

Informally, for a given alert A, a car X is *reachable*, subject to the operating radius constraint, if there exists a "path of cars" over time that can relay the alert A from its originator to X. For example, consider the case in Figure

[2] This restriction is not as severe as one may think. In practice, there are usually structures between parallel roads that interfere with or prevent communication between parallel roads. We make this simplification to avoid the complexity caused by "cross-communication" between two parallel roads in formal analysis.

2(c). Suppose car U is the originator of an alert. Now even if car U and V are never in communication range, car V is still *reachable* because there exists a "path" from U to V, namely $\mathcal{P} \equiv U \rightarrow W \rightarrow V$. In this path $\mathcal{P}$, cars U and W are in range of each other at some point in time. Later on, as shown in Figure 2(d), cars W and V are also in range. Notice two important points: 1) the existence of a path in reachability does not imply that any implementation of RAS must route the alert along this path, 2) even if all successive pairs of cars in this path $\mathcal{P}$ are not in range of each other simultaneously, over time by relaying the alert along the path $\mathcal{P}$, the alert can reach car U.

The notion of *on-time* captures "when" is a car notified about an alert. For a RAS to be useful, we must notify cars before they breach the safety radius. Suppose a car X breaches the safety radius of an alert A at time t, then we call the delivery of an alert A to X *on-time* if car X receives a broadcast about A before time t.

3.2 Correctness and Problem Definition

With the notion of reachability and on-time, we can discuss the meaning of implementing a RAS *correctly*. Again we only give the informal definition here.

Definition 1. *(*Correctness*) Given a set of cars $\mathcal{V}$, an alert A, the originator S of A, and the safety and operating radius, an implementation of a RAS $\mathcal{R}$ is* correct *if for every car $X \in \mathcal{V}$ such that there is a reachable path from S to X before X first crosses the safety radius, then $\mathcal{R}$ delivers the alert A to X on-time.*

Note that the correctness only says to deliver an alert on-time, not as soon as possible. Thus an implementation can delay propagating an alert if it is more "efficient" and does not violate the on-time criterion.

Problem Definition: Devise a distributed protocol that *correctly* implements a regional alert system while minimizes the number of broadcasts.

---

4 Details of Our Protocol BiPP

We describe BiPP in the context of a single two-way road first. Section 4.5 sketches the necessary modifications to handle intersecting roads. To succinctly explain BiPP, we introduce the notion of *inbound* and *outbound* cars.

Definition 2. *A car is* inbound *with respect to an alert A if it is moving toward the alert. Otherwise, it is* outbound.[3]

Figure 4 illustrates our classification of inbound and outbound cars on a two-way road. Cars in the clear area are inbound; cars in the shaded area are outbound. As we will see

[3]Because the way BiPP handles intersection described later in Section 4.5, this definition extends naturally to intersecting and parallel roads.

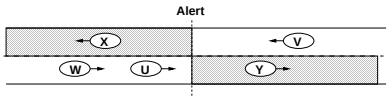

Figure 4. Inbound and outbound.

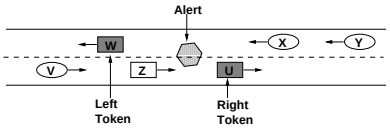

Figure 5. Perimeter Tokens.

shortly, our protocol uses cars differently based on whether a car is inbound or outbound.

4.1 Perimeter Tokens

For a single two-way road, BiPP maintains two types of perimeter *token*s, namely *left* and *right* tokens, as shown in Figure 5. In this figure, if a car knows about the alert, we use a square box; otherwise, we use an oval. Cars holding tokens are represented by shading the corresponding box. In the example, car U holds a *right* token. Car W holds a *left* token. BiPP uses tokens in two ways:

1. A car with a token knows the alert and broadcasts periodically (see below) to disseminate the alert.
2. (*Invariant*) A car between any pair of right and left tokens knows about the alert. (In Figure 5, car Z is between the left token W and the right token U. Thus Z must know the alert, as indicated by the square box.)

BiPP efficiently maintains these tokens beyond the safety radius (if feasible), thus notifying all cars before they breach the safety radius. As illustrated in the Overview (Section 2), a left token is passed to a car that is further to the left; a right token is passed to a car that is further to the right. Although we only have two types of tokens, there can be multiple "active" tokens of the same type. For example, car U in Figure 5 holds a right token. When U broadcasts the alert, cars X and Y both receive the alert. Without any global coordination, both X and Y believe they should "become" the holder of a right token. As a result, all three cars U, X, and Y now hold a right token. Eventually when Y broadcasts, cars U and X will drop their right tokens.

4.2 Passing Tokens

Efficient passing of the tokens is the key in BiPP. The two types of token are passed in a similar manner. Here, we describe how a right token is passed among cars. There are two scenarios to consider depending on where the token is: 1) token is within the safety radius, and 2) token is beyond

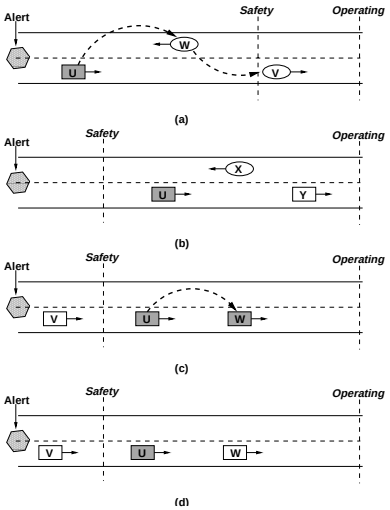

Figure 6. Token Passing.

the safety radius. Figure 6(a) and 6(b) depict the two cases. Token passing in the two cases is different.

In case 1 where the token is inside the safety radius, the token holder, say car X, must broadcast *every time step* (round) to propagate the alert as quickly to the right as possible, regardless whether car X is an inbound or an outbound car. If X does not broadcast every round, then a car that is about to enter the safety radius may not receive the alert on-time. Therefore, the token passing is simply based on the relative location of the cars. Whenever a car X receives a broadcast from a sender Y that is to the left of X, car X creates a right token for itself and begins broadcasting. When Y receives X's broadcast, it will drop its right token. For the example in Figure 6(a), the token will pass from car U to W and then to V.

Case 2 is different because outside of the safety radius, there is less urgency to propagate the token to the right, hence more room for optimization. As argued in the Overview (Section 2), an outbound car is more efficient to carry a token because it broadcasts less frequently. (The exact amount of delay between successive broadcasts is given in the next section.) With the exception of one case, it can be shown that it is not necessary for an inbound car that receives an alert while outside of the safety radius to create a token for itself.

The exception case corresponds to when an outbound car with a token leaves the operating radius, in which case we will permanently lose the token. When this exception occurs, the only solution is for "some" inbound car to create a new right token. Note that if we always maintain the right token with the right-most outbound car which is very close to the operating radius, then this exception case will occur very frequently. Thus there will be a constant juggling of tokens between inbound and outbound cars.

BiPP minimizes the occurrence of this exception case by using a somewhat counter-intuitive approach — instead of propagating the token as far to the right as possible, we maintain the token on an outbound car that is *just beyond the safety radius*. Figure 6(c) illustrates this concept. In this example, both outbound cars U and W have a right token; and BiPP will maintain the token with car U. To keep the token at U and drop the token at W, note first that both U and W will be broadcasting periodically because they have a token. When car W receives a broadcast from U (which includes U's current location), by consulting its own map and location and the alert radii, car W drops its right token because U is closer to the safety radius, resulting in Figure 6(d). Note that the right token is actually being passed to the left in this case. To facilitate this token passing in the opposite direction, in BiPP an outbound car automatically generates a new token when it crosses the safety radius. For the example in Figure 6(d), when car V is eventually beyond the safety radius, it will create a token for itself and start broadcasting. Car V's broadcast in turn will cause U to drop its token.

Our approach of maintaining the token just beyond the safety radius alleviates but does not completely eliminate the exception case where some inbound car has to create a new token. BiPP handles the new token creation on inbound cars by having "inactive" tokens with a "timeout." In other words, an inactive token becomes an active token after a pre-specified time delay. For instance, when an inbound car X receives a broadcast from an outbound car Y, car X will create an inactive token with a time delay that lower bounds the amount of time for Y to leave the operating radius. The detail of inactive tokens is a special case of suppressing unnecessary broadcasts which we describe next.

4.3 Suppression

We use suppression as an optimization for reducing unnecessary broadcasts without explicit coordination. Suppression occurs in two cases: 1) an inbound car with an inactive token, and 2) an outbound car broadcasting infrequently. To implement suppression, each car maintains a suppression counter for each token that it has. Recall that a car with a token is responsible for broadcasting the alert at every time step. The suppression counter is then simply a mechanism for delaying the broadcasts. More specifically, at every time step (round), the counter is decremented. When the counter reaches 0, the car broadcasts and resets the counter if appropriate. The two types of suppression use the counter differently.

Inbound Suppression: Inbound suppression is a fail-safe mechanism for regenerating a token if "all" outbound cars left the operating radius. Therefore, when an inbound car X receives a broadcast from an outbound car Y, car X creates an inactive token. The suppression counter for the in-

active token is determined by how far from the operating radius Y is. If Y is at a distance d away, then the suppress counter for the inactive token is set to d. The counter is then decremented by 1 each round. When the counter expires, the inactive token becomes active. Note that while we are decrementing the counter, if X receives another broadcast from an outbound car, the counter is reseted according to the new position data.

Outbound Suppression: After an outbound car X broadcasts, it is not necessary for X to broadcast again at the next time step; instead X can delay for a period of time before the next broadcast. The exact delay period depends on the communication range and how fast X is moving. Specifically, it is unnecessary to broadcast as long as an inbound car Y (currently just beyond the communication range) can not move into communication range, pass car X, and then move out of range or breach the safety radius. Since it is impossible to tell without communication whether such a car like Y exists or how fast Y is traveling, BiPP makes a pessimistic assumption that car Y exists and is moving at the maximum speed of one position per time step.

Under this pessimistic assumption, if an outbound car X's distance to the safety radius is s and the wireless range is W, then X can safely use a suppression count of $C = W + \min\{W - 1, s\}$. The logic behind C is that if X is stationary, then it takes an inbound car Y at least W time steps to reach X's position from beyond the communication range and at least $\min\{W - 1, s\}$ before it leaves X's range or breaches the safety. Now if X is also moving, then X and Y may get out of range of each other faster. To account for this, suppression counter for an outbound car is updated as follows. If X does not move in the current time step, the suppression counter is decremented by 1; otherwise, the suppression counter is decremented by 2. It can be shown that X and Y do not miss each other using the above suppression counter update.

4.4 Protocol

With the key ideas of BiPP described, we now give some lower level details. To implement BiPP, each car needs to maintain the following local state variables:

1. *r_token* and *l_token*: boolean variables for whether the car has the right or left token.

2. *r_supp* and *l_supp*: suppression counter for the tokens. If the counter is greater than 0, then the token is temporarily inactive.

3. *my_alert*: the content of the alert if any. This variable is unset if the car does not know about the alert.

4. *c_loc*: the car's current GPS location on the map.

$send\_right()$:

```
1: if r_token and r_supp ≤ 0 then
2:     broadcast()
3:     reset(r_supp)
4: else
5:     decrement_counter(r_supp)
6: end if
7: update c_loc and inbound
8: if not inbound and just passed safety then
9:     r_token = true, r_supp = 0
10: end if
```

$recv()$:

```
1: update r_token, r_token, r_supp, r_supp, my_alert
```

Figure 7. Pseudo-code

5. *inbound*: boolean variable for whether the car is inbound or outbound. This variable is undefined if the car does not know about the alert.

When a car broadcasts, the message format of the broadcast is as follows:

$$\langle alert, b\_loc, b\_type, token\_held \rangle$$

The $alert$, b_loc, and b_type fields contain the actual alert information , current sender location, and which token caused the broadcast, respectively. The $token\_held$ field clarifies which tokens the sender has. Even though the sender may only be broadcasting because of a right token, it may have an inactive left token. The presence of an inactive token in our message has two uses: 1) if appropriate, the receiver can use this information to remove its own token without needing the sender to waste another broadcast when the inactive token becomes active, 2) error checking to detect anomalies.

With these details, the protocol can be described in pseudo-code as in Figure 7. The protocol has two components: a sending module and a receiving module. The sending module is executed once per time step (round) to determine whether the car should broadcast this round. The receiving module is executed once per broadcast to update its alert, token holding, and suppression counters. For brevity, we only give right-code related to the right token.

Due to space constraint, we do not give the exact rules for updating the tokens and suppression counters. See the extended technical report [16] for all the rules. Here, we informally describe one such rule for illustration. For example, suppose two cars both hold a right token and are traveling towards the right. If both cars are outbound and beyond safety radius, then the car closer to the operating radius, i.e., farther away from the safety radius, should lose its token. Otherwise, the car farther away from the operating radius should lose its token. (Note that farther away from the operating radius is *not* equivalent to closer to the safety radius.) This rule precisely allows us to keep the token, if feasible, with an outbound car just beyond the safety radius.

Although our protocol does not always use the optimal (i.e., minimum) number of broadcasts, we can, however, give a strong statement on its correctness.

Theorem 3. *BiPP* correctly *implements a regional alert system as defined in Section 3.*

We do not give the proof here. In Section 5, we experimentally verify this theorem, i.e., BiPP reaches the same cars as the naive solution where every car that knows the alert broadcasts constantly.

4.5 Intersection

So far we have only described BiPP in the context of a single two-way road. BiPP handles intersections by dividing intersecting roads into four road segments and handle each segment individually as a single two-way road. For a road segment R that does not contain the alert, BiPP creates a "virtual" alert location on R and adjusts the safety and operating radii accordingly. Specifically,

- For a road segment R that does not contain the alert, if one can reach the alert from either end points of R via other routes without leaving the operating radius, the alert location on R is set to be the middle of the segment. Otherwise, the alert location on R is the end point that can reach the alert.
- For a segment R that does not contain the alert or part of the alert safety radius, the safety radius is set to 0, and the operating radius is set to the whole segment.

Aside from changing the alert location and radii for different road segments, cars approaching and crossing the intersection must take additional actions. Specifically,

- When a car is moving into communication range of an intersection, if it knows about the alert, regardless whether it currently holds a token or not, it must broadcast periodically (with appropriate suppression).
- When a car that knows the alert "enters" a new road segment R (i.e., moving from one side of the intersection to another), it creates a new token of the appropriate type depending on the the "virtual" alert location on R. (This information can be derived locally from the map and the actual alert location.)

It can be shown that the above two modifications are necessary to ensure correctness. Due to space constraint, see the extended technical report [16] for more detail.

5 Evaluation

We evaluate BiPP against two protocols: (1) the naive protocol that always broadcasts, and (2) the IVG protocol that only uses inbound cars for disseminating the alert.

5.1 Simulation Setup

Our goal is not to model some specific road or scenario, but rather to construct a simple synthetic environment that makes it possible to quantify the difference between schemes. Thus we use a simple round-based simulation. For the simulation, we use a single two-way road represented by a linear chain of 99 nodes (road positions). There are no limits on how many cars can be at a single node simultaneously.

During each round, a car may move into an adjacent node or stay at its current node. The trajectory of a car is generated so that the car moves continuously from one end of the linear chain to the other, i.e., there are no U-turns in the movement of the cars. We also use one of eight different speeds for each car, chosen randomly. The different speeds are emulated as follows. For the slowest speed, a car moves to the adjacent node in one round and pauses (i.e., does not move) in the next round. This move and pause sequence is then repeated until the car reaches the other end. For the next slowest speed, a car moves for two rounds and then pauses. Similarly, for the higher speeds, a car moves for k rounds and then pauses.

After all the cars have moved in a round, any car can broadcast a message regardless whether it moved or not. All the broadcasts by different cars occur at the same time. We do not simulate collisions between multiple broadcasts or message losses. All broadcasts are delivered to cars within the communication radius. Unless stated otherwise, we use a broadcast radius of 10 nodes. For some experiments, we do vary the broadcast radius from 4 to 20. Upon receiving a broadcast, each car is given the opportunity to process the message and adjust its state. We do not, however, allow a car to change its own broadcast after hearing other car's message. If a car receives multiple broadcasts in a single round, the order of message arrival is arbitrary, i.e., we do not guarantee messages from the closest car arrives first.

In our simulation, we use a single alert event. The alert is generated by one of the cars, chosen at random. This chosen car will initiate the alert as it passes through the middle of road, i.e., when it reaches node 50 on our 99-nodes two-way road. The duration of the alert is also chosen randomly between 500 to 1000 rounds. In our evaluation, we vary the safety radius from 10 nodes to 40 nodes. Our operating area for the cars include the entire road, i.e., cars will participate in disseminating the alert until it leaves the road.

We also use different car densities in our simulations. For each of our simulation, cars do not all enter the road at the same. Instead, we allow each car to enter the road at a random time chosen between rounds 1 and 1000. Thus, we control the density by having different number of cars in our simulation. As we will see in the later graphs, we vary the car density between 50 to 1000 cars for our simulations.

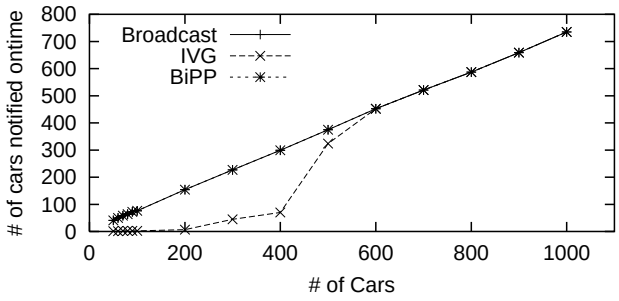

Figure 8. Number of cars notified on-time.

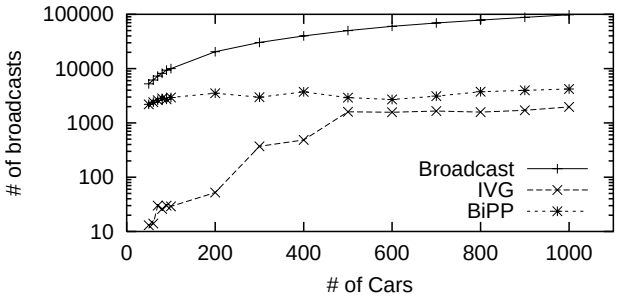

Figure 9. Overhead in number of broadcasts.

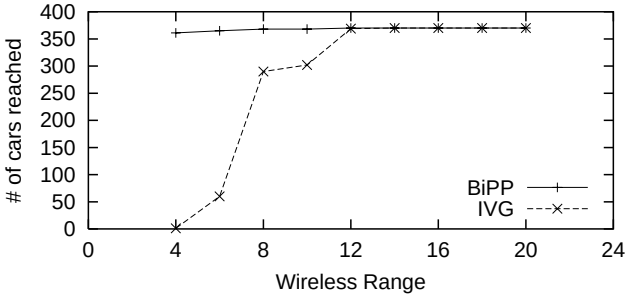

Figure 10. Reachability with Varying Range

5.2 Reachability

Unlike other protocols such as IVG, BiPP ensures that all vehicles receive the alert if feasible. To illustrate this property, we ran our simulations with each of the three protocols with safety radius 10 and different car densities. The result is shown in Figure 8. The x-axis gives the car density. The y-axis shows how many of these cars are notified before they breached the safety radius of the alert.

The Broadcast curve in Figure 8 is the baseline of comparison because it will reach all cars possible. Therefore, the BiPP curve which coincides with the Broadcast curve demonstrates that our algorithm does indeed reach all cars. However, the IVG curve is below the Broadcast curve for low density, showing that it fails to notify all cars. The reason is that if two inbound cars are not close enough, the alert message will not be propagated by IVG. BiPP, on the other hand, is able to overcome this difficulty by using outbound cars to carry the message. When the density is higher, say above 500 cars for this particular evaluation setup, IVG achieves the same. If we change the setup to use safety radius of 40, then even at high density, IVG does not reach all cars.

5.3 Overhead

The second objective of BiPP is to reduce the broadcast overhead as much as possible. We now show that BiPP gives significant reduction against the naive broadcast algorithm and is comparable against IVG in performance, while an giving extra correctness guarantee. Our simulation varies the car density and uses safety radius 40. The result is shown in Figure 9. The x-axis is the car density. The y-axis is the number of broadcasts, shown in log scale.

Naturally, the overhead of the broadcast protocol increases almost linearly as the number of cars increase. In contrast, IVG and BiPP are not very sensitive to car density. The reason IVG's overhead increases in the low density range (from 50 to 500) is purely because IVG stops prematurely when it cannot reach all the cars.

The important thing to note from Figure 9 is that when BiPP and IVG both reach the same number of cars (i.e., for car density > 500), BiPP's overhead is no worse than twice of IVG's overhead We cannot compare the two protocol for lower car densities because IVG stops broadcasting prematurely. Interpreting this observation differently, BiPP's performance penalty for guaranteeing to reach all cars is actually small. Even with this performance penalty, compared to the naive broadcast protocol at high density, BiPP's overhead is almost two orders of magnitude lower.

5.4 Varying Wireless Range

The communication range affects the number of cars reached and the overhead. To illustrate, we ran an experiment with 500 cars, a safety radius of 20, and different wireless ranges. Figures 10 and 11 show the result with BiPP and IVG. The x-axis gives the communication range W.

The number of cars reached is shown in Figure 10. Note that communication range has almost no effect on BiPP as we reach approximately the same number of cars for ranges of 4 and 20. Even though there are more car fragmentations with smaller ranges, our use of outbound cars can overcome most of these fragmentation. On the other hand, IVG which only uses inbound cars can not handle small communication ranges as evident from the shape decline in the number of cars reached with smaller ranges.

Despite the fact that smaller communication ranges in BiPP do not affect the number of cars reached, it does increase the broadcast overhead. Figure 11 shows that

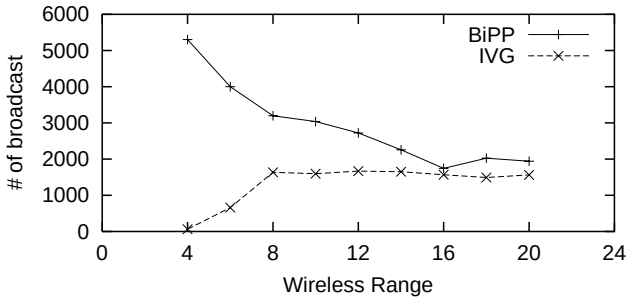

Figure 11. Overhead with Varying Range

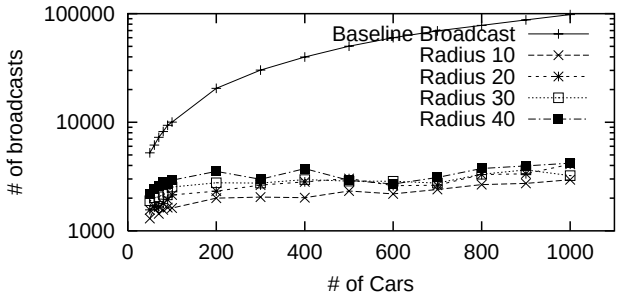

Figure 12. Varying safety radius

BiPP's overhead declines steadily with larger communication range. The same overhead reduction holds for IVG, though with noticeably less effect. (The initial increase for IVG was due to that fact that IVG is able to reach more cars with bigger communication range.)

5.5 Varying Safety Radius

The size of the safety radius also affects BiPP's overhead because we aggressively propagate an alert beyond the safety radius. However, once we begin to maintain the alert notification beyond the safety radius, having a larger safety radius has minimal impact. To illustrate this, we ran simulations with varying safety radius. The results are shown in Figure 12. Again, the x-axis is the car density. The y-axis is the number of broadcasts.

As predicted, larger safety radius incurs more overhead, as seen in the graph, to account for the extra initial cost of pushing the alert out of the safety radius. However, the gap in the overhead is constant for the various densities, which suggests that there are no additional costs once the initial alert has been propagated beyond the safety radius.

6 Concluding Remarks

This paper explores how to build an efficient regional alert system by using bidirectional traffic and maintaining a perimeter intelligently for a single alert. We demonstrated that our protocol BiPP is efficient in propagating an alert. For practical purposes, BiPP's overhead is independent of safety radius and car density. BiPP also performs superbly in notifying cars of the alert even when communication range is very small. Moreover, BiPP's overhead is within a small constant factor (typical within a factor of 2) of IVG's overhead while providing much stronger guarantees and tolerance for varying communication ranges.

References

[1] A. Bachir and A. Benslimane. A multicast protocol in ad-hoc networks: Inter-vehicles geocast. In *IEEE VTC Spring*, 2003.

[2] A. Benslimane. Optimized geocasting of alarm messages in vehicular ad-hoc networks. Technical report, Laboratoire d'Informatique d'Avignon LIA, 2004.

[3] E. Bommaiah, M. Liu, A. McAuley, and R. Talpade. Amroute: Ad hoc multicast routing protocol. Internet-Draft, draft-talpade-manet-amroute-00.txt, 2000.

[4] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Proceedings of MobiHOC*, 2000.

[5] L. Coletti, N. Riato, A. Capone, and L. Fratta. Architectural and technical aspects for ad hoc networks based on utra tdd for inter-vehicle communication. In *Proc. IST Mobile Summit*, 2003.

[6] M. Sun et. al. Gps-based message broadcast for adaptive inter-vehicle communications. In *IEEE VTC Fall*, 2000.

[7] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, 2000.

[8] David B Johnson and David A Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. 1996.

[9] Y. Ko and N. Vaidya. Geotora: A protocol for geocasting in mobile ad hoc networks. Technical report, Dept. of Computer Science, Texas A&M University, 2000.

[10] S. J. Lee, M. Gerla, and C. C. Chiang. Ad hoc wireless multicast with mobility prediction. In *Proc. IEEE ICCCN*, 1999.

[11] E. L. Madruga and J. J. Garcia-Luna-Aceves. Multicasting along meshes in ad hoc networks. In *Proc. IEEE ICC*, 1999.

[12] Julio C. Navas and Tomasz Imielinski. Geocast - geographic addressing and routing. In *Mobile Computing and Networking*, pages 66–76, 1997.

[13] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. P. Sheu. The broadcast storm problem in a mobile ad hoc network. In *MobiCom*, 1999.

[14] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. INFOCOM*, 1997.

[15] C. E. Perkins, E. M. Belding-Royer, and S. Das. Ad hoc on demand distance vector (aodv) routing. IETF RFC 3561, 2003.

[16] Q. Sun and H. Garcia-Molina. Using ad-hoc inter-vehicle networks for regional alerts (extended version). Technical report, Stanford University, 2004. Available upon request.

[17] C. W. Wu, Y. C. tay, and C. K. Toh. Ad hoc multicast routing protocol utilizating increased id-numbers (amris) functional specification. Internet-Draft, draft-ietf-manet-amris-spec-00.txt, 2000.

[18] H. Zhou and S. Singh. Content based multicast (cbm) in ad hoc networks. In *Proc. 1st Annual Workshop on Mobile and Adhoc Networkingand Computing*, 2000.