# 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 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

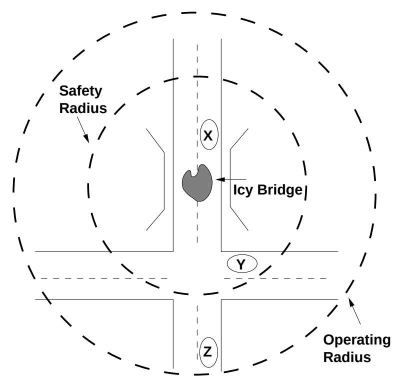


**Figure 1. A scenario of regional alert.**

the *safety* radius. Moreover, 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 is 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. The goal is to design a system that uses as few broadcasts as possible. Precise description of the problem and assumptions are given in Section 3 and Appendix A. 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 it is 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, i.e., 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. As far as we know, we are the first to study the problem of guaranteeing the delivery of the alert and the problem of maintaining an alert for a duration.

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]

[1] One can build a regional alert system using additional infrastructure like cellular towers. Although simplier than an ad-hoc solution, infrastructure-based solution has to deal with another problems such as standardization, deployment, servicing, and pricing.

One simple solution for building a regional alert system is to have vehicles that know about an alert "continuously" 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. One may argue that propagating one single alert does not generate much traffic even if broadcasting all the time, or perhaps the broadcast messages can be piggybacked on other traffic. However, consider an "emergency" scenario such as a snow storm in the New England area. The storm would cause many local alerts to be generated. The simple solution of broadcasting "continuously" by all cars is bad because the aggregate traffic is high and the interference among broadcasts becomes a serious issue. 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. The challenge of this approach is how to maintain the perimeter dynamically when cars enter, move around, and leave the alert region while ensuring to notify "all" cars.

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.
- Proof of correctness for BiPP.
- A demonstration, via simulation, that our protocol has very low overhead in the number of broadcasts.

As a clarification, for this work we use a simple model with assumptions including maximum speed for cars, GPS, and maps (Section 3. While one could argue that it may be more efficient to implement RAS within the lower-level MAC layer, we view RAS as an application-level protocol that should be implemented on top of a broadcast primitive. 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 "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.

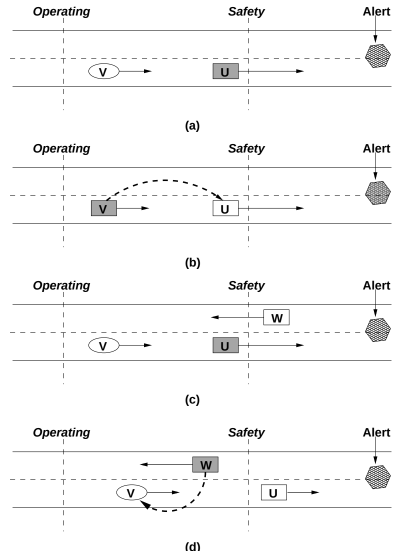


**Figure 2. Example of alert propagation**

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 in Appendix D.

### 2.1 Related Work

The three most relevant papers on disseminating alerts are Role-based Multicast (RBM)[4], TRADE [15], 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 earlier. The three schemes differ in how they address the fragmentation. 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[12]. To solve this simultaneous rebroadcasting problem, the Distance Delayed Time (DDT) [15] mechanism is used. In DDT, after receiving a broadcast from a sender, one sets 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. A similar technique based on prioritizing different types of messages with different delays is used for disseminating emergency messages in the Vehicular Collision Warning Communication (VCWC) protocol[17].

Maintaining alerts is also similar to various flavors of ad-hoc multicast [3, 9, 10, 16, 11, 8] 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[14, 7, 13]. 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 [6] 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 is as follows.

1. Communication and processing occur in synchronized rounds. We assume cars have GPS devices, thus they can achieve synchronized clocks. Car movements and processing of messages occur during the round. Transition from round $i$ to round $i+1$ happens at a pre-specified time interval, e.g., every 200 milliseconds. Communication occur only at the end of the round.
2. A global map known by every vehicle. We discretize the map on a 2D grid, and model it as a graph $G = (V, E)$. 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 $y$ is the hop count (number of edges) in the shortest path from node $x$ to node $y$ in $G$.
3. 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

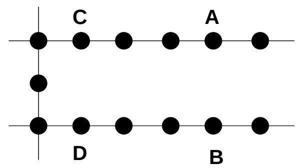


**Figure 3. Example map and communication.**

may either stay at its current location or move to an adjacent grid points. We only allow cars to make U-turns at intersections. Cars also know their own location from their GPS devices.

4. A single alert (as a simplification for ease of discussion). We represent the alert as a tuple of the form ⟨*location, start_time, duration, safety, operate*⟩. 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.

5. There is a source car $S$ who initiates the alert.

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 $W$.[3]

- A car can broadcast up to *one* message per round.

- No implicit message acknowledgment of wireless broadcasts. In other words, a car will not know if its broadcast is received by anyone.

- We *do not* model MAC-layer message transmissions or losses, e.g., signal interference, retransmissions, etc.. We do not expect our application-level protocol to have control over how the MAC-layer operates. We assume there exists an API for broadcasting a message. In the extreme, the underlying MAC-layer can operate in a *Time Division Multiple Access* (TDMA) mode, like cell phones, to ensure no interference between multiple broadcasts from nearby cars.

### 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. Appendix A gives formal definitions.

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(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 $t$.

[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.

[3]Communication range can not be arbitarily large because FCC has regulations on the maximum transmission power level.

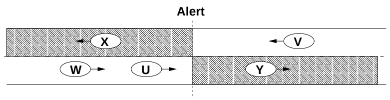


**Figure 4. Inbound and outbound.**

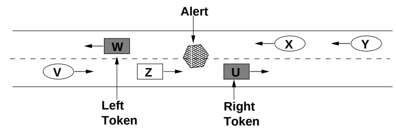


**Figure 5. Perimeter Tokens.**

### 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. Appendix D explains how BiPP handles 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*.*

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

[4]For the purpose of conveying the general principle of BiPP, here we ignored one exception to this invariant when using the broadcast suppression optimizations, described later in the section.

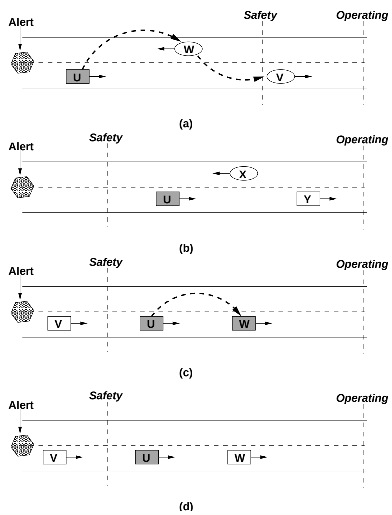


**Figure 6. Token Passing.**

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 rightmost 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 inactive 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 $d$. We make a conservative assumption that car $Y$ would travel that the maximum allowable speed, i.e., one position per round. Thus, the counter is decremented by 1 each round to ensure the token becomes active before outbound car $Y$ leaves the operating radius. 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 conservative assumption that car $Y$ exists and is moving at the maximum allowable speed, i.e., one position per time step.

Under this conservative 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

Informally, each car in BiPP keeps track of which tokens it has and the corresponding suppression counters. Every round, if a token is not suppressed, then the car broadcasts. After a broadcast, the suppression counter is reset as described previously to schedule when to broadcast next. If a car does not broadcast, then the suppression counter is decremented. Whenever a car receives a broadcast, depending on whether a car already has a token or not, the relative location to the sender, and traveling direction of both cars, it can decide to create a token for itself or destroy its own token. In all cases, the suppression counters are also updated to reflect the latest information. The pseudo-code description of BiPP and detailed rules for managing the tokens and suppression counters when receiving a broadcast are given in Appendix B.

Although BiPP does not always use the optimal (i.e., minimum) number of broadcasts for a particular traffic pattern, we can, however, give a strong statement on its correctness.

**Theorem 3.** *BiPP* correctly *implements a regional alert system as defined in Section 3 on a two-way road.*

How BiPP handles intersections to guarantee correctness is explained in Appendix D. Intuitively, under BiPP there is "always" a car beyond the safety radius "broadcasting." This car may not be the left-most or the right-most; nevertheless, any inbound cars eventually pass the broadcasting car and get the alert before breaching the safety radius. The detailed proof, dealing with reachability and on-time aspects of the correctness condition, is given in Appendix C. Another interesting question is whether BiPP is “minimal,” i.e., is every message necessary to guarantee correctness? In Section D, besides discussing intersections, we also offer our conjecture that BiPP is “minimal.”

## 5 Evaluation

We experimentally quantify, via simulation, the cost and benefit of using BiPP, as compared to two other protocols: (1) the naive protocol that always broadcasts, and (2) the IVG protocol that only uses inbound cars for disseminating the alert. This section gives some simulation results on reachability and overhead. Appendix E gives additional simulation results on the impact of communication range and safety radius.

### 5.1 Simulation Setup

The detailed simulation setup and mechanics are described in Appendix E. Here we give the relavent parameters needed to understand the results we present. We compared the protocols on a single two-way road, consisting of 99 nodes (positions) connected in a linear chain. The operating radius is set sufficiently large to cover the entire road. The communication range is 10 positions. We varied the number of cars in the simulation. Each car enters the road at a random time, selected uniformly from 1 to 1000; it also chooses randomly to go from left to right or from right to left on the road. In this simulation, each car has one of eight different speeds. The alert is in the middle of the road (node 50) and is active from time 132 to 790.

### 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 7. The x-axis gives the car density. The y-axis shows the precentage of these cars are notified before they breached the safety radius of the alert. The cruve for BiPP overlaps with the curve for Broadcast where cars that know about the alert continuously broadcast at every opportunity. Note that even with continuous broadcasts, fewer than 100% of the cars are reached because not all cars travel during the life of the alert.

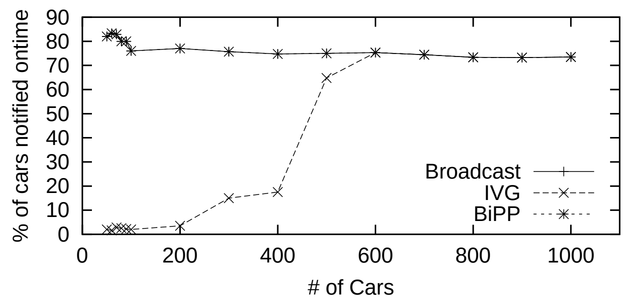


**Figure 7. Number of cars notified on-time.**

The BiPP curve coincides with the Broadcast curve, demonstrating that our algorithm does indeed reach “all” cars as the correctness condition requires. 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 coverage of cars as BiPP. If we change the setup to use a safety radius of 40, i.e., reducing the distance between the operating and safety radii, 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 giving the extra correctness guarantee. Our simulation varies the car density and uses a safety radius of 40. The result is shown in Figure 8. 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 8 is that when BiPP and IVG both reach the same number of

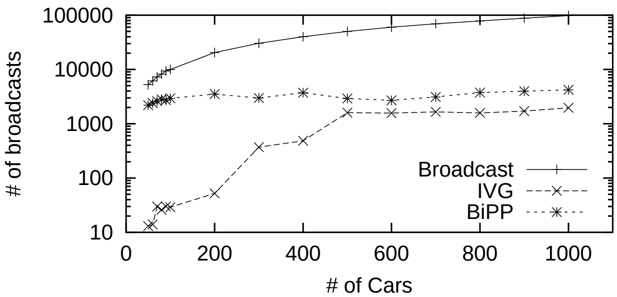


**Figure 8. Overhead in number of broadcasts.**

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.

## 6 Concluding Remarks

This paper explores how to build an efficient regional alert system by using bidrectional 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 the safety radius and car density. BiPP also performs superbly in notifying cars of the alert even when the 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 limited 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] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, 2000.

[7] 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.

[8] 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.

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

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

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

[12] 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.

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

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

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

[16] 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.

[17] X. Yang, J. Liu, F. Zhao, and N. Vaidya. A vehicle-to-vehicle communication protocol for cooperative collision warning. In *MobiQuitous*, 2004.

[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.

## A Formal Definitions

In this appendix, we give formal and rigorous definitions of reachability, on-time, and correctness.

### A.1 Notation

Recall that we model each car's trajectory as a set of pairs $\langle location, time \rangle$. For example, car $X \equiv \{\langle (1,1), 1\rangle, \langle (1,2), 2\rangle, \langle (1,3), 3\rangle, \ldots\}$ denotes a car that starts at coordinate $(1,1)$ at time 1, continues to coordinate $(1,2)$ at time 2, and so on. 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. Thus if $X \equiv \{\langle L_1, 1\rangle, \langle L_2, 2\rangle, \ldots, \langle L_k, k\rangle\}$, then we enforce the invariant $D(L_i, L_{i+1}) \leq 1$.

### A.2 Reachability and On-time

One important aspect of propagating alerts is to deliver the alert to all "reachable" cars within a time duration. Another aspect is to deliver the alert "on-time" before a car breaches the safety distance of an alert. Here, we formalize these two notions.

**Reachability** We begin with the *reachability graph* defined over a set of cars. A reachability graph is a directed graph based on the trajectories of the cars, the alert location $A$, and the operating radius. Let $\mathcal{V}$ be the set of cars and $W$ be the transmitting range, we use $R(\mathcal{V}, W, A, operating)$ to denote the resulting reachability graph.

In $R(\mathcal{V}, W, A, operating)$, each car $X \in \mathcal{V}$ is described by a set of nodes $X_1, X_2, \ldots, X_k$ where $X_1$ represents car $X$ at time 1, $X_2$ represents $X$ at time 2, and so forth. For each $X_i$, there is a directed edge from $X_i$ to $X_{i+1}$. There is also a directed edge from node $Y_i$ to node $Z_{i+1}$ if $D(Y_i, Z_i) \leq W$ (i.e., car $Y$ and $Z$ are within communication range at time $i$) and car $X$ is within the operating radius of alert $A$ at time $i$. For example, suppose there are two cars $A$ and $B$. Car $A$ has trajectory $\{\langle (1,1), 1\rangle, \langle (1,2), 2\rangle, \langle (1,3), 3\rangle\}$. Car $B$ has trajectory $\{\langle (1,3), 1\rangle, \langle (1,2), 2\rangle, \langle (1,1), 3\rangle\}$. If the transmission range $W$ is 1, then the resulting reachability graph is shown in Figure 9. Because $A$ and $B$ are only in range of each other at time 2, there is an edge from $A_2$ to $B_3$ and an edge from $B_2$ to $A_3$.

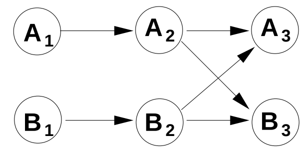


**Figure 9. Example of a reachability graph**

From this reachability graph, we can define reachability between two cars as follows.

---

**Definition 4.** *Given a set of cars $\mathcal{V}$, an alert $A$, a radius $operating$, and a transmission range $W$, a message at car $X$ at time $\mathcal{I}$ can* reach *a car $Y$* before *time $\mathcal{F}$ if there exists a directed path from $X_\mathcal{I}$ to $Y_\mathcal{F}$ in the reachability graph $R(\mathcal{V}, W, A, operating)$. We use the notation $X_\mathcal{I} \rightarrow Y_\mathcal{F}$ to indicate reachability.*

---

Note that the above definition only tells us whether it is *feasible*, i.e., exists a path, to propagate a message from car $X$ at time $\mathcal{I}$ to car $Y$ at time $\mathcal{F}$. The reachability definition does not, however, insist or guarantee a message must actually be forwarded between the cars. The decision of when and what to communicate is to be determine by algorithms that propagate alerts.

**On-time** A regional alert system must also ensure that cars are notified, if possible, before they breach the safety distance of an alert. To formalize this concept, we first define the *first crossing time* $\mathcal{C}$.

---

**Definition 5.** *For a car $X \equiv \{\langle L_1, 1\rangle, \langle L_2, 2\rangle \ldots\}$ and an alert $A = \langle loc, start, duration, safety, operate \rangle$, the* first crossing time *of car $X$* with respect to *alert $A$ is the earliest time $\mathcal{C} \geq start$ such that $D(L_\mathcal{C}, loc) \leq safety$. We use the notation $\mathcal{C}(X, A)$ to denote the first crossing time of car $X$ with respect to alert $A$.*

---

Note the first crossing time is always after the alert becomes active. In other words, we exclude the case when a car $X$ crosses the safety boundary before the alert began because we can never notify $X$ "on-time". Nevertheless, $X$ would most likely hear about the alert as the protocol tries to notify other cars beyond the safety radius.

From the first crossing time, we can define *on-time.*

---

**Definition 6.** *An alert $A$ is delivered* on-time *to car $X$ if car $X$ receives a message about $A$ before the first crossing time $\mathcal{C}(X, A)$. We use the notation $A \rightsquigarrow X_{\mathcal{C}(X,A)}$ to denote $A$ arrived at car $X$ before $\mathcal{C}(X, A)$.*

---

### A.3 Correctness

With the notion of reachability and on-time, the correctness condition can be formally stated as follows.

---

**Definition 7.** *(*Correctness*) With transmission range $W$, an algorithm $\mathcal{A}$* correctly *implements a regional alert system if for all possible sets of cars $\mathcal{V}$, for all possible alert $A = \langle loc, start, duration, safety, operating \rangle$, and alert $A$'s associated source car $S \in \mathcal{V}$, the following condition holds.*

- *for each car $X \in \mathcal{V}$, if $start \leq \mathcal{C}(X, A) \leq start + duration$ and $S_{start} \rightarrow X_{\mathcal{C}(X,A)}$, then $A \rightsquigarrow X_{\mathcal{C}(X,A)}$.*

---

## B BiPP Protocol Details and Rules for Managing and Passing Tokens

We first give pseudo-code for the BiPP protocol and then talk about the detailed rules for token passing.

### B.1 BiPP Pseudo-Code

To implement BiPP, each car needs to maintain the following local state variables:

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

2. *right_suppress* and *left_suppress*: 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.

5. *direction*: this variable can take on the value of *left* or *right* to indcate the direction of travel.

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

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

The $alert$, $c\_loc$, 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.

The protocol can be described in pseudo-code as in Figure 10. The protocol has two components: a sending module and a receiving module. The sending module is executed once per round to determine whether the car should broadcast this round. The receiving module is executed once per receiving broadcast to update its alert, token holding, and suppression counters. For brevity, we only give right-code related to the right token.

The $reset$ routine schedules when the next broadcast should be. For an outbound car with the alert and beyond the safety radius, the next broadcast is delayed according to the suppression rules described in Section 4.2. Otherwise, it broadcasts every round. The $decrement\_counter$ routine reduces the suppression counter towards the next broadcast. We now describe the rules for token management.

### B.2 Rules

BiPP relies on passing and suppressing the *left* and the *right* tokens among cars to maintain a perimeter on a single two-way road. The detailed rules for how the tokens move and whether they active or not depend on four factors: 1) traveling direction of the sender and receivers, 2) relative locations of the sender and the receiver, 3) relation to the safety radius, and 4) whether the receiver know the alert or not.

Figure 11 gives detailed rules for managing the *right* token, i.e., the sender has the *right* token. The table summarizes what action the receiver will take upon hearing a broadcast about the alert. The meaning of the different columns in the table are as follows:

- *Recv Knows*: Indicates whether the receiver already knows the alert or not.
- *Sender/Recv Dir.*: Give the sender's and receiver's traveling direction.
- *Sender/Recv $\geq$ safety*: Indicate whether the sender and the receiver are to the right of the alert location and beyond the safety radius.
- *Recv $\geq$ Sender* Indicates whether the receiver is the right of the sender or not.

For the table entries, we also use the notation $distance\_to\_right$ for the distance from the sender to the right end-point of operating radius on the road. We use $|Sender - Recv|$ for the distance between the two cars.

The *Action* column provides specific tasks for the receiver. In BiPP, the action consists of two components: 1) decide whether to create an active token, indicated by the boolean variable $Recv.right$, and 2) manipulate the suppression counter for the token, indicated by $Recv.right\_suppress$. For example, consider the rule in the first row. The sender is traveling to the right and beyond the safety radius. The receiver, who does not know the alert, is traveling to the left and to the right of the sender. Because the sender is an outbound car beyond the safety radius, it is more efficient for the sender to continue propagating the alert. Therefore, the receiver will create an active right token and suppress it. The suppression count is set to the distance from the sender to right end-point because if there are no more cars on the road, the receiver will eventually become the right token holder when the sender leaves the area. Since the receiver does not know how fast the sender will travel, it takes a conservative estimate and assume the sender will travel as fast as allowed, i.e, one position per round.

Other rules in the table follow the same logic. The only addition is that new right tokens are generated when an outbound car passes the right safety radius. This token generation step is to efficiently maintain the token as close to the safety radius as possible described in Section 4.2.

```
send_right():
if right_token = True and right_suppress ≤ 0 then
    broadcast(my_alert, c_loc, right, right_token and left_token)
    reset(right_suppress)
else
    decrement_counter(right_suppress)
end if
update c_loc
if my_alert not null and c_loc ≥ my_alert.loc +safety then
    right_token = True, right_suppress = 0
end if

reset(right_suppress) :
if c_loc < my_alert.loc +safety then
    right_suppress = 0
else
    right_suppress = W + min(W-1, c_loc - my_alert.loc - safety)
end if

decrement_counter(right_suppress) :
if direction = left then
    right_suppress = right_suppress - 1
else
    if car moved this round then
        right_suppress = right_suppress - 2
    else
        right_suppress = right_suppress - 1
    end if
end if

recv():
update right_token, right_suppress, my_alert according to token passing rules.
```

**Figure 10. Pseudo-code**

| Recv Knows | Sender Dir. | Sender $\geq$ safety | Recv Dir. | Recv $\geq$ safety | Recv $\geq$ Sender | Action |
|---|---|---|---|---|---|---|
| No | Right | Yes | Left | - | Yes | Recv.right = True<br>Recv.right_suppress = distance_to_right_end |
| No | Right | Yes | Right | - | Yes | Nothing (unless intersection) |
| No<br>No | -<br>Left | No<br>Yes | -<br>- | -<br>- | Yes<br>Yes | Recv.right = True<br>Recv.right_suppress = 0 |
| Yes | Right | Yes | Right | - | Yes | Recv.right = False<br>Recv.right_suppress = -1 |
| Yes | Right | Yes | Left | - | Yes | Recv.right_suppress = distance_to_right |
| Yes | Right | No | - | - | Yes | Anomaly, broadcast once to suppress sender |
| Yes | Left | - | - | - | Yes | Recv.right = True<br>Recv.right_suppress = 0 |
| No | Right | Yes | Left | Yes | No | Recv.right = True<br>Recv.right_suppress = distance_to_right |
| No<br>No<br>No | Right<br>Right<br>Left | -<br>Yes<br>- | Right<br>Left<br>- | -<br>No<br>- | No<br>No<br>No | ERROR |
| Yes | Right | - | Left | - | No | Recv.right_suppress = distance_to_right |
| Yes | Right | Yes | Right | - | No | Nothing (our next broadcast will suppress sender) |
| Yes | Left | - | Right | - | No | Recv.right_suppress = $\|Sender - Recv\|$ |
| Yes | Left | - | Left | - | No | Recv.right = False<br>Recv.right_suppress = -1 |
| Yes | Right | No | Right | - | No | Recv.right = True<br>Recv.right_suppress = distance_to_right |

When a car $X$ that holds an active (possibly suppressed) right token is leaving the right-endpoint of the operating radius, $X$ initiates a single broadcast at the boundary of the operating radius.

For a car $X$ that satisfies the following three conditions: 1) traveling to the right, 2) knows about the alert, and 3) on the safety radius to the right of the alert, $X$ generates a new right token with suppression counter of $0$.

**Figure 11. Rules for managing the right token in BiPP on a single two-way road.**

The *left* token is managed in a similar and symmetric manner. BiPP does use one additional optimization. When a car broadcasts, the broadcast message includes which tokens the car current holds and whether they are active or not. The receiving car processes the active token as described in Figure 11. For the inactive token, it only applies the two rules where a token may be destroyed, i.e., the rules in Figure 11 where $Recv.right = False$. (Note, if we process the inactive token with all the rules, we will not be able to suppress unnecessary broadcasts as effectively.)

## C Correctness Proof for Single Two-Way Road

One distinguishing feature of BiPP is that BiPP guarantees to delivery an alert to all reachable cars on-time. In this section, we prove this claim for the single two-way road case in two steps. First, we show the correctness of a simpler version of BiPP. And then, we show that BiPP notifies the same set of cars on-time as the simplified version.

### C.1 Correctness of Simplified BiPP

Consider a simplified BiPP, denoted by sBiPP, where we always maintain the token as close to the operating radius as possible. Any token carrier always broadcasts every round, regardless whether the token carrier is inbound or outbound. In other words, we do not use the optimization, described in Section 4.2, where we maintain the token on an outbound car that is just beyond the safety radius. Unlike BiPP where a *right* token may be passed to the left among outbound cars, in sBiPP, a *right* token is always passed right.

The specific rules for sBiPP are listed in Figure 12. Whenever a car receives a broadcast about the right token, it creates a token for itself. If the receiver is further to the right than the sender, then the token is immediately active, i.e., the receiver will begin to broadcast immediately. Otherwise, the token is suppressed for a period of time until it's possible for the sender to have left the operating radius. We now show that sBiPP has the same guarantees as the naive protocol where every car broadcasting continuously within the operating radius after receiving the alert, denote the latter naive case as BROADCAST.

| Recv ≥ Sender | Action |
|---|---|
| Yes | Recv.right = True<br>Recv.right_suppress = 0 |
| No | Recv.right = True<br>Recv.right_suppress = distance_to_right |

**Figure 12. Simplified BiPP's rules for managing a right token.**

**Lemma 8.** *For all time $t$, the location of the right-most broadcasting car under sBiPP is the same as the location of the right-most broadcasting car under BROADCAST.*

*Proof.* We prove by induction. Suppose that at time $t$, locations of the right-most car for sBiPP and BROADCAST are identical. We need to show for time $t + 1$, the locations of the right-most cars are still the same.

There are three cases to consider: 1) under BROADCAST, the right-most car at $t + 1$ is the same as $t$ (i.e., the same car continued to move), 2) under BROADCAST, the right-most car at $t + 1$ received a broadcast from the right-most car at $t$ (i.e., token passing to the right), and 3) under BROADCAST, the right-most car at $t$ left the operating area, thus at $t + 1$, the second right-most car becomes the right-most.

By construction of sBiPP where all cars sharing the same location as the right-most car all have active tokens, for cases 1 and 2, sBiPP behaves identically as BROADCAST. For case 3, note that sBiPP suppresses the right token of the second right-most car while BROADCAST does not. However, because our suppression is conservative in that it assumes the right-most car would travel at maximum allowable speed, the suppression counter would have expired before the right-most car leaves the operating radius. Therefore, in case 3, the second right-most car in sBiPP would be broadcasting at $t + 1$, i.e., has an active right token. Consequently, the locations of the right-most broadcasting car are identical for sBiPP and BROADCAST, as required by the induction. □

Similarly, the left-most broadcasting cars also have identical location under sBiPP and BROADCAST. From this, we can show that sBiPP satisfies the correctness criterion.

**Lemma 9.** *sBiPP correctly implements a regional alert system as defined in Section 3.*

*Proof.* Proof by contradiction. Suppose sBiPP is not correct, then there exist a map $M$, a traffic pattern $P$, an alert $A$, and safety/operating radii $S$ and $O$ such that a car $C \in P$ is not notified on-time under sBiPP while BROADCAST did notify $C$.

Without loss of generality, suppose car $C$ enters the operating radius via the right. Let $t_e$ be the time car $C$ enters the operating radius and let $t_s$ be the time car $C$ breaches the safety radius. Because BROADCAST did successfully notify $C$ before time $t_s$, there exists a car from which $C$ received a broadcast between time $t_e$ and $t_s$. There may be multiple such cars, so let us choose car $D$ and time $t_n$ where $t_e \leq t_n \leq t_s$ such that 1) car $D$ was in communication range of $C$ at time $t_n$, 2) $D$ broadcasted at time $t_n$, and 3) car $C$ did not know the alert before $D$'s broadcast. In other words, $D$ is the first car to notify by $C$. By construction, car $D$ must the right-most car at time $t_n$ under BROADCAST. By Lemma 8, car $D$ would have broadcasted under sBiPP at time $t_n$ also. A contradiction since $C$ would have known about the alert before breaching the safety radius. □

### C.2 Correctness of BiPP

From the correctness of sBiPP, we can show BiPP is also correct. We need the following lemma.

**Lemma 10.** *For all time $t$, if the location of the right-most broadcasting car under sBiPP is beyond the right safety radius, then there exists a car* scheduled to broadcast *under BiPP that is beyond the right safety radius. If the location of the right-most broadcasting car under sBiPP is within the right safety radius, then BiPP has a car broadcasting at the same location as sBiPP.*

In other words, BiPP is just as aggressive as sBiPP to propagate the token when the right-most broadcasting is within the safety radius. BiPP is less stringent when the cars are beyond the safety radius. Here, *scheduled to broadcast* means the car will broadcast sufficiently frequently to ensure all cars passing through its range are notified before breaching safety radius.

*Proof.* Proof by induction. Suppose the lemma holds for time $t$, need to show for time $t + 1$. There are four cases to consider depending on the location of the right-most broadcasting car under sBiPP at time $t$ and $t + 1$.

*Case 1*: the right-most broadcasting car under sBiPP is beyond the right safety radius at both time $t$ and $t + 1$. Two things can happen: 1) no car has left the operating radius, and 2) the right-most car under sBiPP has left the operating radius. If no car has left, there is only one way for the induction hypothesis to be false at time $t + 1$ — at time $t$, an inbound car $F$ is broadcasting, and at time $t + 1$, $F$ moves into the safety radius. However, this scenario cannot happen because by construction, car $F$ under BiPP would only become active to broadcast if it did not meet any outbound car to suppress its broadcast. Moreover, all the inbound cars would be broadcasting with the token being passed to the right. Therefore, if the right-most car in sBiPP is beyond the safety radius, i.e., not $F$, then there must be another inbound car beyond the safety radius that has also began broadcasting under BiPP.

The situation when the right-most car $E$ under sBiPP leaves the operating radius is more complicated. There are two sub-cases: A) BiPP did not schedule a car at the same location as car $E$ to broadcast, and B) a car at car $E$'s location was scheduled. Subcase A is trivial because the induction hypothesis still holds since BiPP scheduled another car that has not left the operating radius. For subcase B, there are two scenarios. The first scenario is an inbound car, beyond the safety radius, took over the broadcasting because the suppression counter had expired. Fortunately, both sBiPP and BiPP do the same thing, thus our claim still holds. The second scenario is that under sBiPP another outbound car $F$, beyond the safety radius, is broadcasting. In this second scenario, there are no cars between $E$ and $F$. Now under BiPP, if $F$ knows the alert, then $F$ would have generated a new right token when it passed the safety radius. This token can only be removed if $F$ knows another outbound car beyond the safety radius is also broadcasting. Therefore, either $F$ or another outbound car beyond the safety radius is broadcasting. The remaining possibility of $F$ not knowing the alert can only happen if car $E$ holds the only left and the right tokens at time $t$; otherwise, car $F$ would have passed the left token holder and received

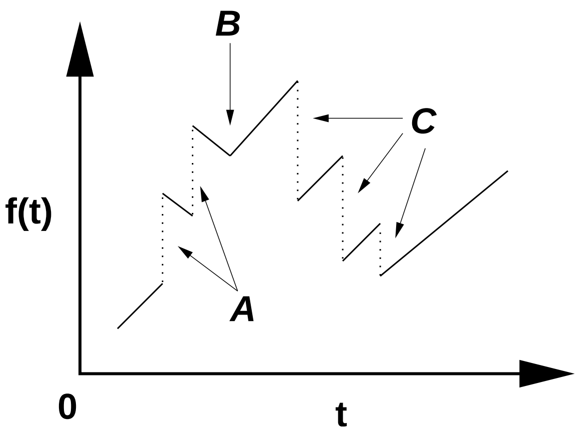


**Figure 13. Over time, the location of the right-most car scheduled to broadcast.**

the alert. However, if $F$ knows the alert under sBiPP at time $t$, then the induction hypothesis, applied to the left token, guarantees that at time $t$ some car to the left of $F$ holds a left token, which is contradictory to car $E$ holding the only left token. Therefore, car $F$ must know the alert at time $t$; hence, as shown earlier, the induction hypothesis holds for time $t + 1$.

The other three cases are argued in similar manner. For brevity, we omit them. □

Similarly, the left token has the same property. From Lemma 10, we can now show that BiPP is correct.

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

*Proof.* Let $f(t)$ be the location of the right-most car under BiPP that is scheduled to broadcast at time $t$. For convenience, let $f(t) > 0$ mean the car is to the right of the alert, thus $f(t) > safety$ implies the car is beyond the safety radius. This function $f(t)$ is piecewise "continuously" with a few "jumps" that strictly decreases the value of $f(t)$. More formally,

$$f(t+1) = \begin{cases} f(t) \pm 1 & \\ f(t) - \delta & \text{for some } \delta \end{cases}$$

The "continuous" plus or minus 1 adjustments are from the movement of the cars. The "jumps" occur when the tokens are passed. The jumps can increase $f(t)$ if the tokens are passed between inbound cars; or they can decrease $f(t)$ if the tokens are passed between outbound cars. Figure 13 illustrates the various cases. The x-axis gives the time; the y-axis gives the location. The dotted line highlight the "jumps" in $f(t)$. The value of $f(t)$ gradually increases if the token carrier is outbound. Similarly, $f(t)$ decreases if the carrier is inbound. There are three things to note, namely labeled $A$, $B$, and $C$ in Figure 13. The jumps at $A$ correspond to token being passed from an inbound car to another inbound car further to the right. The transition at $B$ indicate the token has been passed from an inbound car to an outbound. And jumps at $C$ occur when tokens are passed between outbound cars. For clarity, we use $f(t) = operating$ to indicate that the car is at the operating radius

With this definition of $f(t)$, we prove our claim by contradiction. Suppose BiPP does not notify a car $D$ on-time when sBiPP does. Let $t_e$ be the time when $D$ enters the operating radius. Let $t_s$ be the time $D$ breaches the safety radius. Let $g(t)$ denote the location of car $D$ at time $t$. Note $g(t_e) = operating$, $g(t_s) = safety$, and $g(t)$ is continuous. Consider $f(t)$ from $t_e$ to $t_s$. If there are no jumps in $f(t)$, then $f(t)$ is continuous. By the intermediate value theorem, $f$ and $g$ must cross each other at some time $t_c$. By construction, BiPP would have notified car $D$ "around" time $t_c$, depending on when the suppression counter expires.

Now let us examine what happens when there are jumps. If the jump is upward at time $t_j$, then the token is passed to an inbound car. Suppose $g(t_j) < f(t_j)$, then the broadcast that caused the token to jump would have notified $D$ also — a contradiction. If $g(t_j) > f(t_j)$, then we ignore the jump and consider $f$ and $g$ from time $t_j$ to $t_s$. As a result, we can ignore all upward jumps and focus on downward jumps.

Consider the first downward jump, say occur at some time $t_j$ where $t_e \leq t_j \leq t_s$. The new $f(t_j)$ value can be below the safety radius or above. By Lemma 10, $f(t)$ can only be below safety if and only if the right-most broadcasting car under sBiPP is within the safety radius at time $t_j$. In other words, if $f(t_j)$ is below safety radius, then it must be the case that an outbound car left the operating radius and the second right-most car is within the safety radius. Note, an outbound car leaving the operating radius implies $f(t_j - 1) = operating$. Now because $f(t_e) < operating$, $f(t)$ is continuous, and $g(t)$ is continuously decreasing, $f$ and $g$ would intersect by the intermediate value theorem. At the point of intersection, car $D$ would have received a broadcast about the alert — a contradiction. Thus, we can assume jumps never yield a new $f(t)$ value below safety.

For the second case where $f(t_j) \geq safety$, if $f(t_j - 1) > g(t_j - 1) + W$ where $W$ is the communication range, then by the same argument as before, $f$ and $g$ would have crossed each other earlier and $D$ would have received a broadcast about the alert before $t_j$. If $g(t_j - 1) + W \geq f(t_j - 1) > g(t_j - 1)$, then $D$ can also hear the broadcast that caused the right token to pass backwards. Thus the only remaining case is $f(t_j - 1) < g(t_j - 1)$. In this case, we can simply ignore time from $t_e$ to $t_j - 1$ and repeat the above analysis of downward jumps from time $t_j$ and onward to time $t_s$ until there are no further downward jumps. Let that time be $t_d$.

At time $t_d$ when there are no more downward jumps, $f(t_d) < g(t_d)$. Because $f$ is continuous and $g(t_s) = safety < f(t_d)$, $f$ and $g$ must cross. Thus $D$ would be notified by an outbound car — a contradiction. Therefore, BiPP notifies the same cars on-time as sBiPP. Because sBiPP is correct, BiPP is also correct. □

## D Intersections

BiPP handles intersections by dividing intersecting roads into four road segments and handling each segment individually as a single two-way road. In this approach, we need to address two questions: 1) where are the alert locations on various road segments, and 2) what are the safety and operating radii for these segments.

We illustrate, via a simple example of one intersection shown in Figure 14(a), how we segment an intersection into road segments and assign alert locations and radii for the various segments. In this example, the original alert location is in road segment $D$. The safety radius of the alert is entirely in segment $D$, while the operating radius covers the intersection.

BiPP partitions the intersection into four segments, namely $A, B, C$, and $D$, as shown in Figure 14(b). In order to run our single two-way road algorithm on each of these segments independently, BiPP must assign a "virtual" alert location and "virtual" radii for the seg-

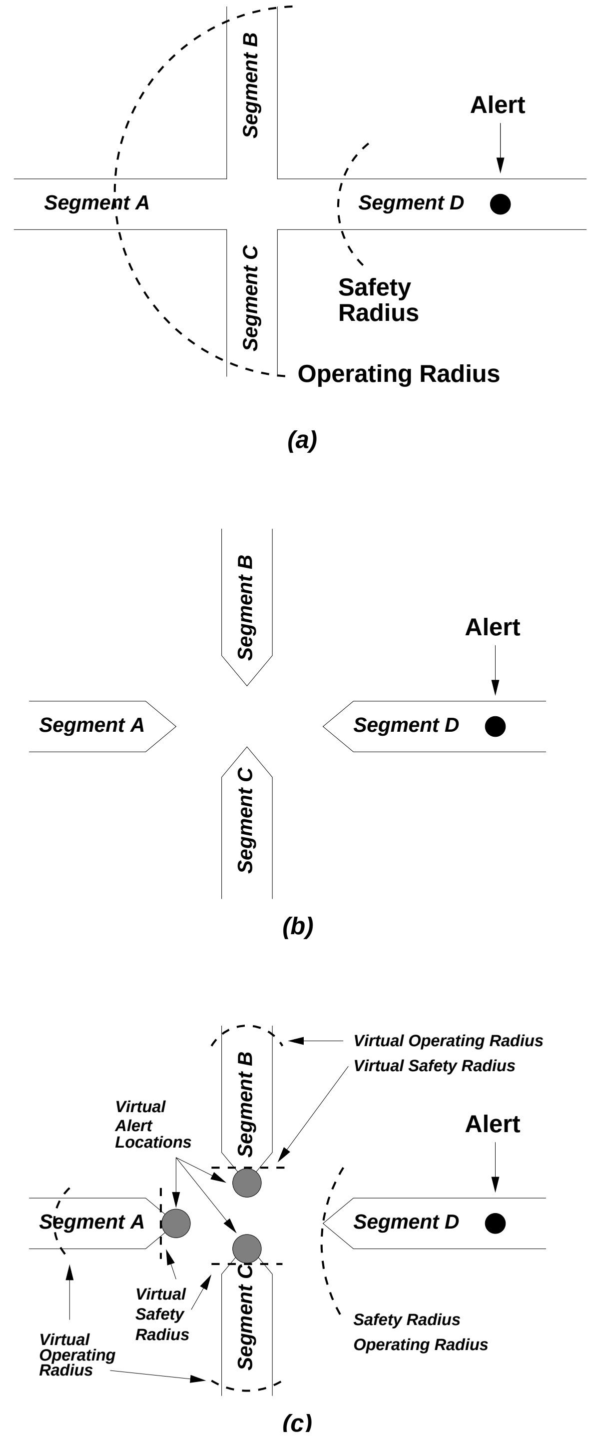


**Figure 14. Example of segmenting an intersection.**

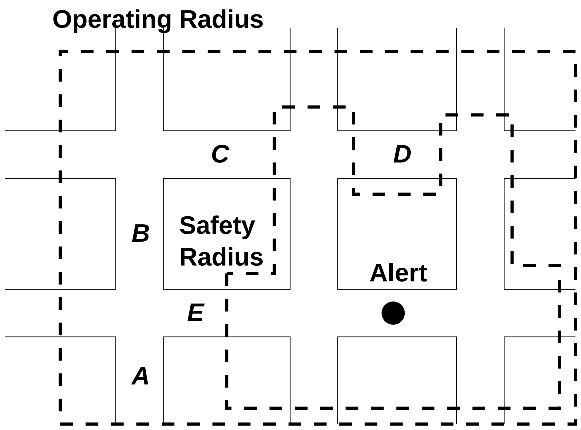


**Figure 15. Complex example of intersections**

ments. Figure 14(c) shows how these assignments are done for this simple example. Because segment $D$ contains the original alert, it simply retains it. However, segment $D$'s safety and operating radii are both extended so that they cover the intersection. To ensure correctness, the safety radius must be expanded because BiPP maintains the perimeter near the safety radius instead of the left-most or right-most cars. As a result, the left-most car near the intersection may not be broadcasting. One can construct counter-examples where the car nearest the intersection must continuously broadcast in order to notify cars on-time before they breach the safety radius.

Unlike segment $D$, segments $A, B$, and $C$ do not contain the alert or the safety radius. In order to run BiPP on these segments, we introduces a virtual alert location at the intersection for these three segments. The safety radii for the segments are also set to $0$, i.e., at the intersection, because no cars can violate the safety condition on these road segments. Segments $A, B$, and $C$ also retains their original operating radius. For cars on these three segments, they simply execute BiPP with the virtual alert location and radii.

In addition to the simple example shown in Figure 14 with only one intersection, Figure 15 shows a scenario in a city-grid where the operating radius encompasses multiple intersections. The safety radius is set according to a fixed traveling distance from the alert location, thus covers some parts of road and intersections. Segment $A$ in Figure 15 is in a similar situation as the simpler example with just one intersection, as in Figure 14. Hence, BiPP would set up a virtual alert location for segment $A$ at the intersection with an appropriate operating radius and a safety radius of $0$. Segments $C$ and $E$ in Figure 15 are in identical situation as segment $D$ in Figure 14. Thus BiPP would expand the safety radius of those segments to cover the entire road segment. Because the safety radius cover the entire segment, where we placed the virtual alert location on those segments does not matter. BiPP simply puts the alert location in the middle of the segment.

Unlike the others segments $B$ and $D$ in Figure 15 are a little different. For segment $D$, because the safety radius occurs on both end-points, we have to propagate an alert along the middle section of the road segment as quickly as possible to ensure correctness. Therefore, we again have to extend the safety radius to cover the entire segment. For segment $B$, one may think that we can more "lazy" in propagating and broadcasting the alert; unfortunately, "lazy" propagation does not work. We must still propagate as quickly as possible. Consider a counter-example where there is a car $X$ at the intersection of segments $B$ and $E$. Also suppose that there is another car $Y$ at the intersection of segments $B$ and $C$. Now if car $X$ receives a broadcast about an alert, then it may be the case that the only way to notify $Y$ about the alert is for $X$ to propagate the alert as quickly as possible along segment $B$ via other cars on segment $B$ to car $Y$. As a result, we can not take any advantage of the "lazy" propagation in the single two-way road case.

From this grid example shown in Figure 15, we illustrated that even if a road segment is not in the safety radius of an alert, we may still have to treat the segment as if it was entirely within the safety radius and propagate the alert quickly. Under BiPP, we use the simple rule below to determine whether the safety radius would be expanded for a particular segment.

- For a segment $X$, if both end-points of $X$ are within the operating radius of an alert $A$, then segment $X$ should be treated as if it is entirely within the safety radius.

In other words, for the grid example in Figure 15, the safety radius would expand to include segments $B, C, D$, and $E$ in their entirety, as shown in Figure 16.

**Theorem 11.** *BiPP, with road segmentation and expanded safety radius, correctly implements a regional alert system as defined in Section 3.*

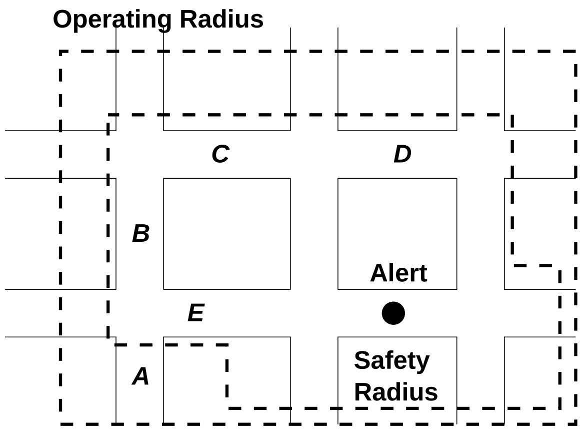


**Figure 16. The expanded safety radius after segmentation**

The proof uses similar arguments as in the single two-way road case. With the expanded safety radius to cover all intersections in the operating radius, the alert would be propagated as quickly as possible. We skip the details here.

The expanded safety radius does require BiPP to use more broadcast messages. Recall that while inside the safety radius, to guarantee correctness, cars "pushes" the alert to the left-most and right-most cars as aggressively as possible. Hence, the suppression components of BiPP will not be used at all. However, it is unclear whether anything more intelligent can be done. In particular,

**Conjecture** Protocol BiPP is *minimal* in that if we remove any broadcast message from the execution, then there exists a "similar" traffic pattern that would cause BiPP to violate the correctness condition.

Here, "similar" can be defined as identical traffic pattern up to the point where a BiPP broadcast message is considered to be unnecessary. The conjecture could be true because having a grid of intersections introduces many "loops" and routes that cars can take before they enter the safety radius. On the other hand, the conjecture could be false because having multiple cars from different segments broadcasting near a single intersection does not appear to be minimal. If BiPP is not minimal, it would interesting to find additional optimizations or rules to make BiPP minimal.

# E More Evaluation

We first give the detailed simulation setup. We then show results from different communication range and safety radius. For varying communication range, we compared our BiPP with IVG to see how the range affect cars notified on-time and the broadcast overhead. For varying safety radius, we are interested to see how BiPP's broadcast overhead is affected.

## E.1 Detailed 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.[5] 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, i.e., the road has as many lanes as necessary.

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. In this section, we use a broadcast radius of 10 nodes. Upon receiving a broadcast, each car is given the opportunity to process the message and adjust its state.

[5] We did not use a more realistic simulator like NS-2 that simulates MAC-layer protocols because we are only interested in quantifying the differences due to the application-level protocols.

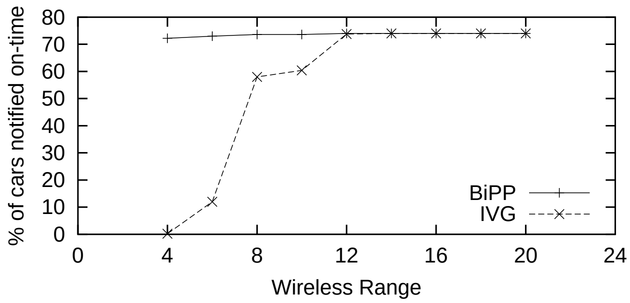


**Figure 17. Reachability with Varying Range**

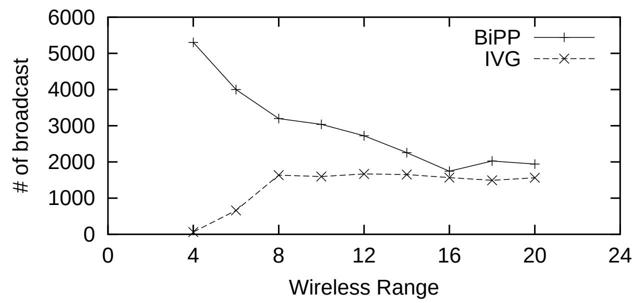


**Figure 18. Overhead with Varying Range**

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. (Note that because we choose a random car to start the alert, the actual start time of the alert is not the beginning of the simulation. Thus some cars would have passed along the road before the alert even begins.) The duration of the alert is also chosen randomly between 500 to 1000 rounds. In this section, we use a safety radius of 10 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. We varied the car density between 50 to 1000 cars for our simulations.

### E.2 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 17 and 18 show the result with BiPP and IVG. The x-axis gives the communication range $W$.

The number of cars reached is shown in Figure 17. 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.

One may argue that rather than using our BiPP, one can simply use IVG with very large communication range. However, there are two pitfalls in the argument. First, with larger communication range, senders are much more likely to interfere with each other's transmission. We did not model interference in our simulation because we assumed the communication range would be small. If one is to use very large ranges, then interference must be considered before one can claim that larger communication range is sufficient. Second, in reality, communication ranges can not be increased arbitrarily because governmental regulatory bodies like the FCC limits the transmission power level. Hence, simply extending communication range in IVG is not a viable solution.

Despite the fact that smaller communication ranges in BiPP do not affect the number of cars reached, it does increase the broadcast overhead. Figure 18 shows that 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.)

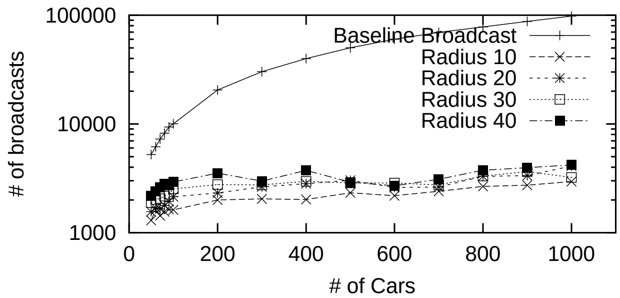


**Figure 19. Varying safety radius**

### E.3 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 19. 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.