Minimizing Broadcast Delay in Location-Based Channel Access Protocols

Shan-Hung Wu
National Tsing Hua University
Hsinchu, Taiwan, ROC
Email: shwu@cs.nthu.edu.tw

Chung-Min Chen
Telcordia Technologies - Applied Research
Piscataway, NJ, USA
Email: cchen2@telcordia.com

***Abstract*—Location-based channel access protocols have been proposed as a means to broadcast safety related messages through inter-vehicle communications. The protocols divide the road into fixed-size cells and assign a channel to each cell. To broadcast, a vehicle would use the channel assigned to the cell it is currently traveling within. To improve bandwidth utilization, a vehicle may acquire channels dynamically from adjacent cells that are not occupied by other vehicles. In a TDMA setting where each channel is a time slot, message delay occurs as the vehicle must wait for the arrival of the next time slot it owns. This message delay time depends heavily on the adopted cell-to-channel mapping function. We examine an existed naive channel allocation scheme and proposed three new ones. An analysis shows that the proposed schemes may reduce the delay by** 50% **to** 90%**.**

I. Introduction

Several large ITS (Intelligent Transportation Systems) initiatives have been promoting the use of vehicle-to-vehicle (V2V) communications to improve driving safety [1], [2]. Such safety applications will require exchange of time-critical information between nearby vehicles, such as braking data from the cars ahead and emergency road conditions, e.g. roadblock, black ice, detected by surrounding cars.

To meet the requirement of bounded latency, the vehicles would broadcast safety and emergency data to nearby vehicles [2], instead of forming local mobile ad hoc networks to route the data using network layer routing protocols. Existing physical and MAC layer protocols, e.g. WiFi or DSRC (Dedicated Short Range Communications), can be used for this broadcast purpose but as indicated in [7], these contention-based MAC protocols usually fall short in meeting the bounded latency and scalability requirements, when the vehicle density is high.

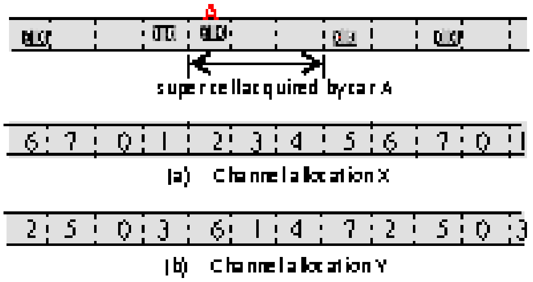
Alternatively, a number of location-based channel access protocols have been proposed [3], [4], [7]–[9], [11], which aim to avoid the aforementioned contention problem. The basic idea is to assign channels to vehicles based on their current location, assuming vehicles are all equipped with certain positioning devices (e.g. GPS). The concept goes as follows: First divide each lane of a road into a sequence of fixed-size (rectangular) cells, where the size of each cell is just large enough to contain one vehicle. Then assign a unique channel to each cell (while channel reuse may be adopted). The cell-to-channel mapping is static information and pre-loaded into each vehicle's on-board unit (OBU). A vehicle will broadcast on the channel that is allocated to the cell it is currently traveling within. Each vehicle will listen to all channels to hear the broadcast from other vehicles.

These location-based protocols alleviate the contention problem but introduce other issues. First, the vehicle moving speed may be too fast to allow a vehicle to complete meaningful data transmission before it moves to the next cell and switches to another channel. While a legitimate concern, this should not be a severe problem for safety applications where typical safety related data are tens to a few hundreds bytes [2], [9] and can be broadcast out quickly within the time constraint. Another problem is low bandwidth utilization which happens when vehicle density is low, as in this case many cells will be unoccupied and their associated channels will be left unused by anyone. Solutions to this problem have been proposed in [4], [8]. In [4], an adaptive protocol was proposed where a vehicle may dynamically acquire and use channels from non-occupied adjacent cells. Depending on the cell-to-channel mapping function adopted, a vehicle may experience different message waiting time with the acquired channels.

In this paper, we focus on the problem of how to find a good cell-to-channel mapping function with low message delay time [4]. This problem arises in the context of TDMA, i.e., when each channel is a time slot. We use an example to illustrate this issue. In the following the terms "channel allocation" and "cell-to-channel mapping" will be used interchangeably.

Example 1. Consider the case in Figure 1 where the top diagram shows cars traveling on a single-lane road divided into cells. The other two diagrams show two different possible cell-to-channel mappings. We assume each TDMA cycle contains 8 time slots. Car A is traveling in a cell where the two cells ahead are unoccupied by other vehicles. So A may acquire the channels allocated to this "super-cell", as indicated in the top diagram. With channel allocation X, depicted in (a), the three channels acquired by A will be $\{2, 3, 4\}$. Multiple cycles may pass during the course when A is still traveling within its current cell and A may attempt to broadcast data at any time. The longest message delay occurs when A wants to broadcast a message at the end of time slot 4, as it has to wait until the next available time slot, namely time slot 2, to arrive. Thus the

978-1-4577-0638-7 /11/$26.00 ©2011 IEEE



(a) Channel allocation X

(b) Channel allocation Y

Fig. 1. Cell-to-Channel Mappings

delay is $8-(4-2+1)=6$ time slots[1]. Now consider case (b) with a different channel allocation Y. The channels acquired by car A are $\{6,1,4\}$, or $\{1,4,6\}$ in time-line order. The longest message delay happens when A wants to broadcast at end of time slot 1, where A needs to wait until the next available time slot, namely slot 4, arrives. The delay is $4-1=3$, which is only half of case (a).

As illustrated above, different channel allocations result in different message delay. Since in safety applications time is of essence as it can spell out the difference between life and death, it is imperative to minimize message delay. Our goal therefore is to find a “good” cell-to-channel mapping function that will minimize the message delay. We propose and analyze a number of channel allocation schemes: a naive sequential allocation adopted in [3], [4], [7], a random allocation, a greedy method, and a Golden Ratio Sequence (GRS) based algorithm. We show that the greedy method can reduce the worst case message delay time of the naive sequential method by more than 50%. The GRS-based method reduces the worst case message delay time of the naive method by almost 90% when the number of channels is a Fibonacci number.

The rest of the paper is organized as follows: Section II provides overview of related work. Section III defines the problem and describes the various channel allocation schemes including the greedy and GRS-based algorithms which are our main contribution. An analytical comparison among the methods is provided in Section IV. Section V provides conclusions and addresses possible extensions.

II. Related Work

The first location-based channel access scheme for inter-vehicle communications, called Space Division Multiple Access (SDMA), was proposed in [3] . Much of what we describe as the basic concept in the previous section is based on this original work. It also addresses the issue of whether the one-vehicle-per-cell assumption is realistic with respect to vehicle speed and the positioning accuracy that can be achieved with modern technologies,

Another scheme called Location-based Channel Access (LCA) was proposed in [7]. LCA is similar to SDMA but takes into consideration channel reuse. An inequality constraint is derived that governs the relationship among parameters including number of channels, cell size, re-use distance, number of lanes, and maximum vehicle speed. This can be useful in practice to avoid channel interference and hidden node problems. It is also noted that the cell size must be large enough to allow the vehicle in the cell to complete meaning transmission before moving into next cell with a difference channel.

Both SDMA and LCA share a common drawback: when the road traffic is light, many of the cells will be unoccupied and the bandwidth associated with their channels will be wasted. Recognizing this bandwidth under-utilized problem, one may allow a vehicle to acquire the channels allocated to unoccupied adjacent cells, as proposed in [4]. A vehicle may detect adjacent cells unoccupied via on-board radar or infrared devices that are becoming more popular in modern vehicles.

In [8] a hybrid method that combines the location-based channel access scheme and a CSMA/CD scheme is proposed. The method uses a larger cell size that may contain multiple vehicles and assigns a number of channels to each cell. Vehicles within the same cell compete for channel access using the RR-ALOHA protocol [5]. This method tries to find a balance between the problems of the location-based protocols (e.g., poor bandwidth utilization due to low vehicle density or high moving speed, and the issue of determining the right cell size) and the contention problem of a pure CSMA/CD protocol (when vehicle density is high).

In contrast to the above decentralized approach where no infrastructure support is needed, [6], [9] proposed a method where the system information such as the cell size and cell-to-channel mapping are kept at a central server and broadcast to the vehicles. The broadcast can be carried out through FM-based RDS (Radio Data Broadcast System) as most vehicles are equipped with a FM receiver[2]. The system may optimize the bandwidth utilization by dynamically "re-programming" the parameters and broadcast such information to the vehicles.

A performance comparison between the location-based channel access method and a simple slotted ALOHA protocol with spatial reuse is presented in [11]

III. Minimizing Message Delay

We consider the location-based channel access protocols where each lane of a road is divided into a sequence of rectangular cells each of which is only large enough to contain one vehicle. Each cell is allocated an unique channel from the channel pool. A vehicle will broadcast data using the channel assigned to the cell it is currently traveling within, without causing conflict with other broadcasting vehicles. Note channel reuse may be adopted by properly configuring parameters such as radio transmission range, cell size and number of channels [4], [7], [8].

[1] Note we have excluded the time slot on which the vehicle is sending the data as part of the message delay. This simplifies notations without affecting the results and allows us to focus on the delay incurred by channel allocation schemes.

[2] FM-based RDS has been deployed in many European and Asian countries and part of the U.S. for traffic information broadcast.

We further narrow the setting to the case described in [4], where the channels are time slots (namely TDMA) and a vehicle can acquire, in addition to its own channel, the channels of adjacent cells that are not occupied by other vehicles, as illustrated in Figure 1. We call the collection of a set of consecutive cells whose channels are acquired by the only vehicle traveling in the area a *super-cell*. As illustrated in Example1, to broadcast a message, the vehicle must wait for the next arriving time slot that is available from the channel pool of the super-cell.

While the exact delay for a specific message broadcast depends on the time instance it takes place and the specific set of channels the vehicle has acquired, the *worst case* message delay can be uniquely determined given a cell-to-channel mapping function. Our goal is to find a good cell-to-channel mapping function, namely a channel allocation scheme, that will minimize the worst case message delay.

A. Problem Formulation

First, we provide a mathematical description of the problem. We assume each cycle (frame) of the underlying TDMA contains N time slots (channels), numbered $0, 1, 2, \ldots, N-1$. The lane under consideration is divided into a sequence of fixed-size cells labeled $0, 1, 2, \ldots$. To make full use of channel-reuse, the channel allocation scheme should always be a "cyclic" scheme. That is, cells i and j should receive the same channel if $i \equiv j \bmod N$, otherwise they receive different channels. Each cyclic scheme then can be uniquely represented by a permutation $P = [a_1, a_2, \ldots, a_N]$ over the numbers $0, 1, 2, \ldots, N-1$, where cell i is assigned channel $a_{(i \bmod N)}$. Any subsequence of P then corresponds to the set of channels associated with a super-cell that may be dynamically acquired by a vehicle.

Our objective is to find a permutation P such that for any subsequence S of P, the numbers in S are uniformly distributed over the range $[0, N]$. It is easy to see a permutation with this property will render low worst case message delay. Given a permutation $P = [a_1, a_2, \ldots, a_N]$, define a subsequence S of length m, $1 \leq m \leq N$, of P as:

$$S = [a_{i \bmod N}, a_{i+1 \bmod N}, \ldots, a_{i+m-1 \bmod N}],$$

for some $1 \leq i \leq N$. That is, S is a subsequence that starts at the i'th position of P with length m, possibly wrapped around to the front if the end of P is reached. For brevity, in the following we will simply use a_j to denote $a_{j \bmod N}$. Sort the numbers in S in ascending order and let the result sequence be

$$S^* = [b_1, b_2, \ldots, b_m].$$

Now, define the *delay* between any two adjacent numbers b_j and $b_{(j+1) \bmod N}$ as:

$$d_j = \begin{cases} b_{j+1} - b_j, & \text{if } j < m \\ N + b_1 - b_m, & \text{if } j = m. \end{cases}$$

Define the *delay* of subsequence S, denoted d_S, as the maximum delay among all adjacent elements in the sorted sequence S^*:

$$d_S = \max_{1 \leq j \leq m} d_j.$$

The *additive error* of S is defined as

$$AR(S) = d_S - \left\lceil \frac{N}{m} \right\rceil.$$

Note $\lceil \frac{N}{m} \rceil$ is the lowest possible delay of any subsequence of length m. So $AR(S)$ is essentially a measure of deviation from the lower bound of the delay. In the following we will ignore the ceiling operation in the above expression as it plays no significant role in the results.

Example 2. Referring to Example 1 and Figure 1.(b), the variables defined above will hold values as: $N = 8$, $P = [0, 3, 6, 1, 4, 7, 2, 5]$, $S = [6, 1, 4]$, $S^* = [1, 4, 6]$, $d_1 = 4 - 1 = 3$, $d_2 = 6 - 4 = 2$, $d_3 = 8 + 1 - 6 = 3$, $d_S = 3$, $AR(S) = 3 - \lceil \frac{8}{3} \rceil = 0$.

Define the *additive error* of a permutation P as

$$AR(P) = \max_{\forall \text{ subsequence } S \text{ of } P} AR(S).$$

That is, $AR(P)$ is the maximum additive error among all subsequences of P. Our objective is to find a permutation P with low additive error $AR(P)$.

B. Sequential Allocation

We describe the simple, naive allocation scheme which was the prototype adopted in [3], [4], [7] when applying to a single-lane road. **Sequential Allocation**: Cell i is assigned channel $i \bmod N$. Equivalently,

$$P = [0, 1, \ldots, N-1].$$

Case (a) in Example 1 is an example of sequential allocation. We prove that the Sequential scheme is actually the worst scheme (in terms of additive error).

Lemma 3. *The additive error of the Sequential scheme is $N - 2\sqrt{N}$.*

Proof: Consider any subsequence S of P which can be represented as $[a, (a+1) \bmod N, \ldots, (a+m-1) \bmod N]$, where $0 \leq a \leq N-1$ and $1 \leq m \leq N$. The largest delay between any two adjacent elements in the subsequence will occur between the last and the first element (as the delay for all others are 1), so $d_S = N - m$ and thus the additive error of S is $AR(S) = (N - m) - \frac{N}{m}$. The value m at which the maximum of $AR(S)$ occurs can be found by taking derivative of the above and solving the following equation: $-1 + \frac{N}{m^2} = 0$, which yields $m = \sqrt{N}$. Substituting it back to the formula leads to $AR(P) = N - 2\sqrt{N}$. ■

Lemma 4. *For any allocation scheme (permutation) P of length N, the additive error of any subsequence of length m is at most $N - m - \frac{N}{m}$.*

Proof: We will prove by Pigeonhole Principle. For any subsequence S of length m, we claim the largest possible delay between any two elements in S^* is $N - m$. Otherwise, suppose there is a delay greater than $N-m$, say $N-m+1$, then there will remain $N-(N-m+1)=m-1$ positions in P from which S should draw its m members, which is a contradiction. ■

It is clear that the maximal additive error of P occurs when $m=\sqrt{N}$, which equals to $N-2\sqrt{N}$. The above two lemmas lead to the following theorem:

Theorem 5. *The Sequential allocation scheme is the worst scheme (in terms of additive error) and its additive error $N-2\sqrt{N}$ serves as a tight upper-bound of the additive error of all possible allocation schemes.*

C. Random Allocation

The Sequential scheme is a poor scheme because the numbers in any subsequence are always consecutive numbers and therefore the last and first numbers will have the largest possible delay. To minimize the delay, one will wish to have a permutation such that the numbers in any subsequence are uniformly distributed over $[1,N]$. This leads to the following random allocation method.

Random Allocation: Given N channels, select a permutation $P=[a_1,a_2,\ldots,a_N]$ at random and with equal probability from all possible permutations. Allocate cell i with channel $a_{i \bmod N}$.

Note we do not need to enumerate all $N!$ possible permutations. We may construct the random permutation as follows: first draw a number at random with equal probability from the pool of alll N numbers, then remove the number from the pool and draw the next number, again at random with equal probability, and repeat the process until all N numbers are drawn. This will take $O(c\cdot N^2)$ time where c is a constant overhead of generating a random number.

Theorem 6. *Let $\Pr(d_S=N-m \mid \; |S|=m)$ denote the probability that, given a subsequence S of length m, the delay of S equals $N-m$. Then under the Random allocation,*

$$\Pr(d_S=N-m \mid \; |S|=m)=\frac{N}{C_m^N}.$$

Proof: For length m there are C_m^N possible combinations of channel assignment with equal probability. The worst case delay $N-m$ occurs only when the m numbers are consecutive numbers and there are only N such cases: $[i,(i+1) \bmod N,\ldots,(i+m-1) \bmod N]$, for $i=0,1,\ldots,N-1$. ■

Based on the above it is reasonable to expect that the Random allocation will perform better than the Sequential allocation, as the chance of getting a "bad" permutation is relatively small.

D. Greedy Method

We propose next a scheme based on greedy method which works as follows: Start with an empty permutation P and insert the numbers $0,1,\ldots,N-1$ one-by-one into P. When inserting a number number x, we select a position in P and place x into that position (called the *pivot* position). The number that previously occupies the pivot position is moved to the end of P. After N iterations the permutation is fully constructed.

The greedy heuristics that guides the selection of the pivot position is described below. Suppose so far a permutation $P=[q_0,q_1,\ldots,q_{n-2}]$ of length $n-1$ has been constructed and we are to insert the next number, $n-1$, into P. There are $n-1$ candidates of pivot position. Then final pivot position is selected to be the one that minimizes the delay (distance) to its two adjacent numbers:

$$\min_{p\in\{1,2,\ldots,n-1\}} \max(d_{p-1},d_p),$$

where

$$d_j=\max(|q_{j+1}-q_j|\,,N-|q_{j+1}-q_j|).$$

The detailed steps are outlined in Algorithm 1.

Algorithm 1: The Greedy Channel Allocation

```
Input: An integer $N$
Output: A permutation $P$ of length $N$
$P[0]=0$;
for $i=1$ to $N-1$ do
    Find the pivot position $p$ in $P$ such that
    $\max(d_{p-1},d_p)\le\max(d_{j-1},d_j)$ for all $j$;
    $P[i]=P[p]$;
    $P[p]=i$;
end
```

Effectively, at each candidate pivot position we only consider the two local subsequences of length 2. We place the incoming number into a position that will result in the lowest delay of these two subsequences. Note that by placing the number into the selected pivot position will not change the delay of other subsequences of length 2, except the two subsequences at the tail of P that contain the last element (the one that was swapped from the pivot position to the end). This greedy "local optimization" avoids the situation where two consecutive numbers $k,k+1$ are placed next to each other which will result in a large additive error of $N-2-\frac{N}{2}$. Note the Greedy algorithm takes $O(N^2)$ to compute the final permutation.

E. Golden Ratio Scheme

Let $\phi=\frac{1+\sqrt{5}}{2}$, i.e. the well-known Golden Ratio number. We construct the target permutation P as follows:

Step 1. Generate N points p_i, $i=0,1,...,N-1.$, between $[0,1]$ such that $p_i=(i*\phi) \bmod 1$. The operation $x \bmod 1$ is a short notation of $x-\lfloor x\rfloor$, i.e., taking the fractional part of x.

Step 2. Sort p_i in ascending order. Let $rank(p_i)$ be the order of p_i in the resultant sorted sequence.

Step 3. Construct the permutation P as

$$P=[rank(p_0),rank(p_1),\ldots,rank(p_{N-1})].$$

The sequence $Q = (i \cdot \phi) \bmod 1$, $i = 0, 1, \ldots, n$ is called the Golden Ratio Sequence (GRS). GRS is known to have the following nice "uniform distribution" property [10]: for any subinterval $I = [a, b] \subset [0, 1]$,

$$\lim_{n\to\infty} \left(\frac{1}{n} |Q \cap I| \right) = b - a.$$

That is, the number of points from Q falling into the subinterval I is in proportion to the length of I. Further, it can be reasoned that any subsequence of Q will also hold the same property and thus the points in the subsequence are uniformly distributed over $[0, 1]$.

What Steps 2 and 3 do is to scale the GRS from the range of $[0, 1]$ to the range of $[1, N]$. As a result, any subsequence of P will preserve the "uniformity" property of the original GRS and thus have low delay between any two consecutive numbers (channels). Applying the GRS scheme to Example 1 where $N = 8$ (a Fibonacci number) results in an allocation as $P = [0, 5, 2, 7, 4, 1, 6, 3]$. Note this is "isomorphic" to the allocation in case (b) of the example, where the latter is obtained by taking $(N - P[i]) \bmod N$, for $i = 0, 1, \ldots, 7$. Though GRS has the nice uniform distribution property, our channel allocation problem exhibits a major difference from the problem of uniformly placing N points in the $[0, 1]$ range. In the latter case, it only considers subsequences that are completely contained within $[0, 1]$ and does not consider those "wrap-around" subsequences. In our problem we need to consider "wrap-around" subsequences because a super cell may sit across the boundary of the repeated permutation pattern. For those wrap-around subsequences the nice uniformity property will in general not sustain. But in the cases when N is a Fibonacci number the property does survive. It is also not hard to see that when $N = F + 1$ for some Fibonacci number F, the scheme has poor performance because the permutation P will most likely to have 0 and 1 as the first and last element. In this case the wrap-around subsequence $[1, 0]$ will have the worst case delay of $N - 1$ (this happens when a safety message is generated at the end of time slot 1 and has to wait until time slot 0 in the next cycle arrives).

IV. Performance Comparison

We compare the performance of the four allocation schemes in terms of additive error. Since additive error is the worst case deviation from the lowest possible message delay for any super-cell, it serves as a good performance metric. For a given channel allocation scheme represented by a permutation P , we compute its additive error by enumerating all subsequences of P. For each scheme, we compute the additive errors for $N = 5, 8, 13, 21, 34, 55, 89$, and 144 channels. These are Fibonacci numbers where the GRS scheme is applicable. Figure 2 shows the result. As discussed earlier, the Sequential scheme exhibits the largest message delay as it is the worst case scenario. The Random scheme improves over the Sequential scheme by reducing $20\% - 30\%$ delay as the number of channels increase. The Greedy method shows a significant reduction of delay over the Sequential scheme by more than 50%. The local optimization strategy adopted by the Greedy algorithm appears to work quite well in practice. The GRS scheme is the best of all and shows extremely low delay compared to other schemes. We have also experimented with non-Fibonacci numbers for N. The results for all schemes except the GRS scheme are rather continuous and stable with respective to the curves shown in the figure. For the GRS scheme, the delay shows great fluctuation for non-Fibonacci numbers (due to the reason we explained earlier).

An analysis on average delay, as opposed to worst case delay, shows similar results assuming a vehicle is equal likely to broadcast a message at any random time. Due to space constraint it is not presented here.

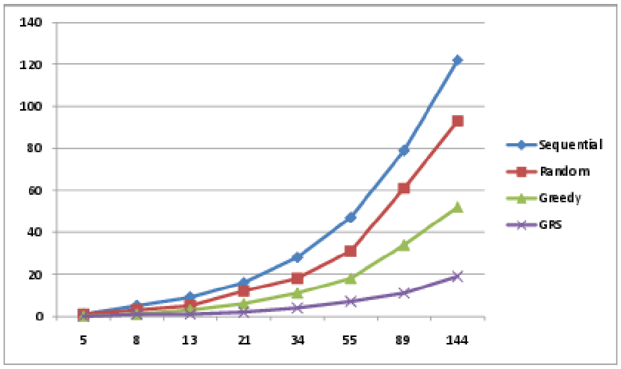


Fig. 2. Comparison of message delay. X: number of channels. Y: additive error.

V. Conclusion

Cell-to-channel mapping affects broadcast delay in location-based channel access protocols. We have examined the existed Sequential allocation method and proposed the Random, Greedy and Golden Ratio sequence based allocation schemes. All schemes can be computed off-line ahead of time and involve very little overhead. The analysis shows that the Greedy and GRS allocations are the top performer but GRS is applicable only when the number of channels is a Fibonacci number.

The analysis is based on worst case delay, but the result will be similar for average case delay. All schemes can be extended to multi-lane roads by using a disjoint set of channels for each lane. Channel reuse can also be achieved by carefully configuring the parameters including radio transmission range, cell size, and number of channels. It will be interesting to extend the schemes to the hybrid method proposed in [8] which could be more practical in use.

As the future work, we will extend our schemes to the multi-lane and crossing scenarios in vehicular networks.

References

[1] "Cooperative Vehicle-Infrastructure Systems (CVIS)," www.cvisproject.org, ERTICO.

[2] "Vehicle safety communications project task 3 final report," www.intellidriveusa.org, 2005, the CAMP VSC Consortium, US DOT, IntelliDrive.

[3] S. V. Bana and P. Varaiya, "Space division multiple access (SDMA) for robust ad hoc vehicle communication networks," in *Proc. IEEE 4th Int. Conf. Intell. Transp. Syst.*, 2001, pp. 962–968.

[4] J. J. Blum and A. Eskandarian, "A reliable link layer protocol for robust and scalable inter-vehicle communications," *IEEE Trans. on Intelligent Transportation Systems*, no. 1, March 2007.

[5] F. Borgonovo *et al.*, "ADHOC MAC: New MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services," *Wireless Networks*, vol. 10, pp. 359–366, 2004.

[6] Y. Choi, R. Rajkumar, P. Mudalige, and F. Bai, "Adaptive location division multiple access for reliable safety message dissemination in vanets," in *Proc. 6th Int. Symp. on Wireless Communication Systems*, Sep. 2009, pp. 565–569.

[7] S. Katragadda, M. Ganesh, R. Ranga, K. Mohan, and R. Sachin, "A decentralized location-based channel access protocol for inter-vehicle communication," in *Proc. 57th IEEE VTC*, Apr. 2003, pp. 1831–1835.

[8] Y. C. Lai, P. Lin, W. Liao, and C. Chen, "A region-based clustering mechanism for channel access in vehicular ad hoc networks," *IEEE J. on Selected Areas in Communications, Special Issue in Vehicular Communications and Networks'*, 2010, to appear.

[9] R. Mangharam, R. Rajkumar, M. Hamilton, P. Mudalige, and F. Bai, "Bounded-latency alerts in vehicular networks," in *Proc. 2007 Mobile Networking for Vehicular Environments*, May 2007, pp. 55–60.

[10] J. Matousek, *Geometric Discrepancy*. Springer, 1991.

[11] N. Wen and R. Berry, "Location-based MAC protocols for mobile wireless networks," in *Proc. Information Theory and Applications Workshop*, Jan. 2007, pp. 41–45.