Information Dissemination Control for Cooperative Active Safety Applications in Vehicular Ad-Hoc Networks

Ching-Ling Huang, Yaser P. Fallah, Raja Sengupta
Institute of Transportation Studies
University of California at Berkeley
Berkeley, CA, USA

Hariharan Krishnan
Electrical and Controls Integration Lab
General Motors R & D Center
Warren, MI, USA

Abstract—**Vehicular Ad-Hoc networks (VANETs) play a critical role in enabling important active safety applications such as collision warning and vehicle tracking. The most pressing challenge in enabling such applications is to maximize the amount of disseminated vehicle state information while avoiding network congestion. In this paper, we explore the structure of VANET tracking problem and propose an adaptive rate control algorithm based on network condition and tracking error. Proposed algorithm uses a closed-loop control concept and accounts for the lossy channel. This algorithm is shown to achieve better tracking performance than existing solutions. We first analyze the algorithm behavior in small scale Matlab simulations with bounded dynamical systems and then evaluate its performance using OPNET simulations with realistic vehicle trajectories from a microscopic traffic simulator (SHIFT).**

I. INTRODUCTION

Vehicular Ad-Hoc Networks (VANET), empowered by the recent advancements in wireless communication technologies, are expected to enable a host of new safety, navigation, and automation applications in vehicular environments. The most challenging application planned for deployment over VANETs is Cooperative Active Safety (CAS) [1], [2]. In the CAS concept, vehicles will send self-information, e.g. GPS position, speed, and heading, to neighboring vehicles over a wireless channel. The receiving vehicle can use the incoming messages to track the sending vehicle and detect if there is a threat. If so, it may warn its driver or perform emergent reactions.

Such safety messages are expected to be about 300 bytes long and may have to be transmitted as far as 150 meters [20]. Studies, such as those undertaken by the Vehicle Safety Communications Consortium (VSCC) [20], have suggested these messages might have to be broadcast by each vehicle as frequently as every 100 msec. While these studies consider safety applications and related requirements, the channel and networking side of the problem has not been fully understood yet.

The communications and networking requirements and possibilities are driven by the choice of technologies for VANETs. The most promising technology for enabling VANETs is the 802.11 based DSRC (Dedicated Short Range Communication [17]) technology [16]. In 1999, FCC allocated 75 MHz of bandwidth, divided into seven sub-channels, in the 5.850-5.925 GHz band for DSRC, to enable vehicle-to-vehicle (V2V) and Vehicle-to-Infrastructure (V2I) information exchange possible. In addition to IEEE 802.11 based standard, the IEEE 1609 family of standards [18] for Wireless Access in Vehicular Environment (WAVE) are being developed to enable networking and resource management services. These standards describe the interfaces between vehicular communications devices at different protocol levels, enabling a spectrum of applications including CAS. The specifications of the applications, such as CAS, is outside the scope of the standards. In particular, the frequency of broadcasting the tracking messages, and the distance to which these messages should be sent have to be determined by the application implementing CAS.

These issues and the study of DSRC networks have been the subject of several research efforts in recent years. For V2V message delivery, the work in [3] gives a detailed evaluation of DRSC channel and its latency performance. For safety applications in VANETs, [4] proposes to fairly allocate transmission power across all nodes in a max-min fashion, while fixing the frequency of messages. Message scheduler in [5] reduces rate by removing duplicated elements similar to compression ideas in source coding. Our approach, presented in this paper, correlates transmission behaviors with tracking error and channel congestion, which is distinctly different from previous work.

In this paper, we start by describing a mathematical framework for the problem of tracking via a multi-access channel. Based on the analysis in [12], [13], we propose a decentralized information dissemination rate control, in application layer, to decide how state information of a vehicle should be broadcast to neighboring vehicles for tracking purpose.

More specifically, the proposed algorithm uses a closed-loop feedback control concept and accounts for the lossy DSRC channel. This algorithm utilizes an *on-demand* structure to adapt its information rate to safety application demands, and allows all vehicles to efficiently share available channel resource.

The behavior of the proposed algorithm is first analyzed in a simpler environment through Matlab simulations. We do so to illustrate why it achieves better tracking performance than other methods, in a multi-access channel. To see the algorithm's performance in a realistic highway environment, the proposed algorithm has been evaluated through large scale simulations, involving vehicle trajectories generated by

Research supported in part by General Motors R & D Center through contract #TCS 70709 to U. C. Berkeley. The views expressed here are those of the authors and not of the sponsors. Contact: chingling@berkeley.edu

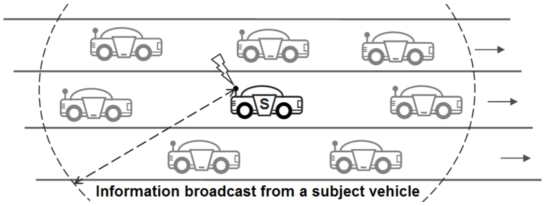

Fig. 1. Subject vehicle broadcasts its own state information to neighboring vehicles. Based on received information, other vehicles track the position, speed, heading of that subject vehicle.

a microscopic traffic simulator (SHIFT) [19] and wireless network simulations in OPNET, with the best known DSRC channel model reported in [8].

The rest of the paper is organized as follows: Section II describes VANET tracking problem. Section III provides insights into how to design a proper rate control scheme for a shared channel. Section IV describes our proposed rate control algorithm. Section V discusses the results of the large scale traffic/network simulation evaluations. Section VI concludes this paper.

II. ESTIMATION PROCESS DESCRIPTION

In this section, we state the problem of vehicle tracking using VANETs in an abstraction level. Fig. 1 illustrates this real-time tracking idea, in which each vehicle broadcasts its state information (e.g., position, speed, and heading) to the neighboring vehicles. Communication policies, which describe how information dissemination is controlled in such broadcast networks, will be described and analyzed in Sections III-V.

To facilitate active safety applications, a subject vehicle (denoted as S in Fig. 1) broadcasts its own state information to neighboring vehicles. Since this is a broadcast message, no ACK (acknowledgment) is sent by the recipients. Based on received information, other vehicles track vehicle S's state. Likewise, every vehicle needs to disseminate its own state information while estimating other vehicles' state.

The tracking functional blocks inside each car are shown in Fig. 2. For $n \geq 2$ nodes (vehicles), which share the same wireless channel, each node contains a plant $x_j(t)$, $j = 1, 2, ..., n$, a communication logic, and a bank of model-based estimators for tracking neighboring nodes (vehicles). On each node, there are a balanced control input $u_j(t)$ to model driver's accelerating/decelerating maneuver and a zero-mean noise input $\varepsilon_j(t)$ to model the mechanical disturbance within a car.

The communication logic in Fig. 2 operates according to a specified rate control policy which decides whether to broadcast self-state information vector $x_j(t)$ at each moment. The design of this rate control policy and communication logic is the main subject of this paper. Each node will try to estimate the states of other nodes using the information received over the shared channel.

To formulate the estimation process, let $\tilde{x}_{ij}(t)$ be the estimated state of sender j at receiver i. This estimated state

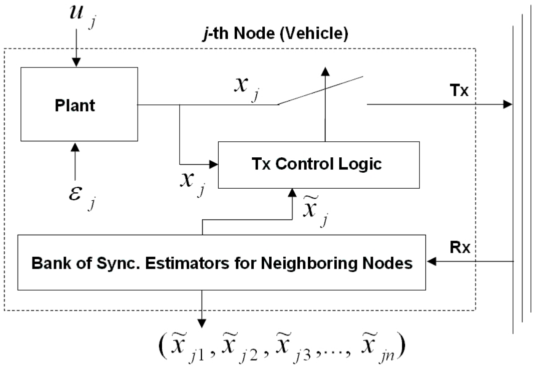

Fig. 2. VANET tracking functional blocks: each node (vehicle) contains a plant, a communication logic, and a bank of synchronized estimators for tracking other plants. The estimated states of neighboring vehicles will be fed into active safety applications.

is the expectation conditioned on all the previous received information from the lossy channel,

$$\tilde{x}_{ij}(t) = E[x_j | Y_i^1, Y_i^2, ..., Y_i^{t-1}] \tag{1}$$

where Y_i^t, $t = 1, 2, ...$, is the received information at moment t at the receiver i. When channel is idle or has a collision at t, $Y_i^t = \emptyset$. Otherwise, $Y_i^t = x_s(t)$ from a certain successful sender $s \in \{1, 2, ..., n\}$. Besides, all information contained in collided packets is treated as lost and there is no retransmission for lost packets (since message is broadcast).

For presentation convenience, let's assume the dynamics of a vehicle can be sampled at discrete time step and approximated as a LTI (linear time-invariant) system of appropriate dimensions:

$$x_j(t) = a_j \times x_j(t-1) + b_j \times u_j(t-1) + \varepsilon_j(t-1) \tag{2}$$

where a_j, b_j represent the mechanical characteristics and physical laws that govern vehicle j. The model-based estimator at receiver $i \neq j$ switches between following two modes:

- If no information regarding node j is received at $t-1$, i.e. $Y_i^{t-1} \neq x_j(t-1)$, use previous estimate $\tilde{x}_{ij}(t-1)$ and the known model (1) to carry on,
$$\tilde{x}_{ij}(t) = a_j \times \tilde{x}_{ij}(t-1); \tag{3}$$
- Else if state information of j is received at $t-1$, i.e. $Y_i^{t-1} = x_j(t-1)$, use it to reset estimation error,
$$\tilde{x}_{ij}(t) = a_j \times x_j(t-1). \tag{4}$$

Note that $\tilde{x}_{ij}(t)$ in (3), (4) is MMSE (minimum mean squared error) optimal estimate if we assume $u_j(t)$ and $\varepsilon_j(t)$ have zero mean. The tracking error $e_{ij}(t)$, the i-th node's estimation error for node j, is defined as

$$e_{ij}(t) = x_j(t) - \tilde{x}_{ij}(t). \tag{5}$$

Note that, due to hidden node problem or channel fading, this tracking error $e_{ij}(t)$ might vary for different vehicles. Our objective in this paper is to devise a decentralized communication policy that controls the transmission of state information in a way that the above error is reduced as much as possible. The next section describes several possible methods and provides an initial analysis. This error measure is used as a control parameter in some of the methods discussed. Our proposed communication policy is conceptually discussed in the next section, while a formal description is presented in section IV.

For general analysis of tracking over a multi-access channel, readers are referred to our preliminary analysis [12], [13] on asymptotic stability and MSE (mean squared error) tracking performance for different communication policies. If centralized scheduling is allowed, round-robin policy is shown to be optimal for homogeneous nodes [12]. For decentralized policy, tracking error and channel condition need to be taken into account to achieve a sub-optimal performance [13]. Our work is mainly inspired by [9], [10], [11], where remote tracking, via a lossy channel, of a sender-receiver pair is analyzed.

III. RATE CONTROL FOR INFORMATION DISSEMINATION IN MULTIPLE ACCESS CHANNELS

In this section, we analyze the problem of rate control for information dissemination in a shared channel. To get insights into how such rate control should be done, we present some Matlab simulations with bounded dynamical systems. For this purpose, we assume the following setting for the Matlab simulatins in this section. We assume that there are $n = 10$ nodes for a 100-second simulation duration. Discrete time step length is set to 50 msec. All identical nodes share one common wireless medium. There is no hidden terminal effect and thus $e_{ij}(t) = e_j(t), \forall i$ in (5). Channel errors are only due to collisions. Each node can detect collisions instantaneously, but there is no retransmission for lost packets. Similar to that in Fig. 2, all nodes contain an identical plant, with different RNs (random numbers) as simulation seeds, and a bank of model-based estimators. For the plant (2), x_j is a scalar with $a_j = 0.5$, $u_j(t-1) = 0, \forall t$, and $\varepsilon_j(t)$ follows a Gaussian distribution $N(0, 10^{-2})$ for all j.

Each node keeps tracking other nodes and broadcasts its own state information according to specified communication policy. Since we are aiming at designs for VANETs, only the following three decentralized policies are considered.

Probabilistic: At each time step, every node generate a packet and broadcasts its own state information with a specified probability p.

Error-Dependent: This policy is due to [9]. At each time step, the communication intensity $\lambda_j(t)$ for node j at t is a function of the current tracking error $e_j(t)$,

$$\lambda_j(t) = \alpha \times e_j^2(t) \tag{6}$$

where $0 < \alpha < \infty$ is a positive real number, representing the sensitivity to estimation error $e_j(t)$. The calculation of $e_j(t)$ is possible because each node also keeps track of all successfully broadcast information and runs an estimator for its local plant to understand the estimation error on other nodes. $\lambda_j(t)$ in (6) is then converted to the Tx (transmission) probability $p_j(t)$ by a continuous mapping from a non-negative real to $[0, 1]$,

$$p_j(t) = 1 - \exp(-\lambda_j(t)). \tag{7}$$

Err-Coll-Dependent: This policy is modified from error-dependent method, and its communication intensity $\lambda_j(t)$ for node j at t is decided by a similar form of (6),

$$\lambda_j(t) = \frac{\alpha}{1 + \beta \times c(t)} \times e_j(t)^2 \tag{8}$$

where $0 < \alpha < \infty$ is a fixed positive real as defined in (6), $0 < \beta < \infty$ is the sensitivity to collisions, and $c(t) \in [0, 1]$ is a time-averaged channel collision intensity for immediate past γ time steps, $\gamma \in N$,

$$c(t) = \frac{1}{\gamma} \sum_{s=t-\gamma}^{t-1} I(s) \tag{9}$$

with $I(s)$ representing channel status of the s-th time step: when there is a collision, $I(s) = 1$; otherwise, $I(s) = 0$. Similarly, $\lambda_j(t)$ is then converted to $p_j(t)$ by (7).

Err-Coll-Dep policy tries to capture two important factors, i.e. $e_j(t)$ and $c(t)$, for tracking over a multi-access channel. Our proposed rate control algorithm in Section IV is a generalization of this policy. Throughout this section, values of (β, γ) in (8) and (9) are fixed at (30,10) for simplicity. By tuning the error sensitivity α in (6) and (8), one can get different average packet rates for Error-Dependent and Err-Coll-Dep policies.

A. Tracking Performance

The time-averaged tracking performance is plotted in Fig. 3 with each point representing the statistic from one simulation run. Since all nodes are identical, we refer MSE (mean square error) as $\frac{1}{nT}\sum_{j=1}^{n}\sum_{t=1}^{T} e_j(t)^2$ in our discussion where T is the total time steps.

First of all, three policies exhibit U-shape curves in Fig. 3(a): 1) when average rate is too low, tracking MSE is large because less state information is transmitted to remote estimators on other nodes; 2) when average rate is too high, MSE becomes large again because many transmitted packets collide in the shared channel and thus the amount of available state information to remote estimators decreases.

The observation that more state information transmitted results in higher tracking accuracy in [9], [10] is not valid in a multi-access channel because of collisions. In this section, we refer collision ratio as $\frac{1}{T}\sum_{t=1}^{T} I(t)$ where $I(t)$ is defined as the same indicator function as in (9) and T is the total time steps. Corresponding collision ratios of three policies are plotted in Fig. 3(b) for comparison.

Secondly, for Probabilistic policy, the minimum tracking MSE is achieved with average rate 2 pkt/node/sec. This corresponds to Tx probability $p_j = \frac{1}{10}$, for all j, which matches with previous analysis $p_j^* = \frac{1}{n}$ in [12].

Thirdly, the Probabilistic policy, i.e. no rate control, cannot achieve the same minimum MSE as that of Error-Dependent and Err-Coll-Dep policies. Error-Dependent and Err-Coll-Dep policies achieve lower minimum MSE with a lower average rate (0.4 pkt/node/sec). Similar to the analysis in [9], [12], controlled communication policies result in better tracking because timely delivered state information eliminates large tracking errors.

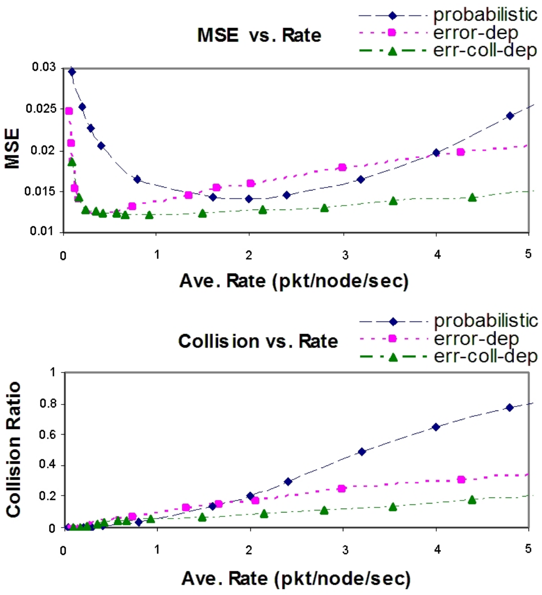

Fig. 3. (a) Average packet rate vs. tracking MSE (mean squared error), (b) Average packet rate vs. average collision ratio for three policies. Among three, Err-Coll-Dep policy achieves lower estimation MSE by adapting its packet rate based on estimation error and channel collision intensity.

With the same average packet rate, Err-Coll-Dep policy can achieve lower tracking MSE compared with that of Error-Dependent, especially when packet rate is high. Err-Coll-Dep can also achieve MSE close to the minimum for a wider range of rates (0.4 to 3 pkt/node/sec), i.e. the curve acts as if tracking MSE is not affected by the increasing packet rate and the gradually congested channel.

B. Collision Response

Fig. 4 depicts collision response while different communication policies are used; this measure is a key performance metric for a multi-access channel. High collision ratio means that the probability of a successful transmission is low and lots of bandwidth is thus wasted. For this figure, the average packet rates of nodes are all around 0.2 pkt/node/sec, which means that the channel is only lightly loaded. Apparently, Error-Dependent policy has higher collision ratio and more consecutive collisions compared with others, which can also be observed in Fig. 3(b) when the rate is small.

For Error-Dependent policy, when a collision happens, it can be infered that at least two nodes want to send out state information at the same step because their tracking errors are relatively large. In the next time step, those nodes would still have high broadcast probabilities since their tracking errors remain large (because no state information is successfully transmitted due to previous collisions). The same situation may go on, i.e. consecutive collisions, until all involved nodes can get a chance to transmit state information.

A design based on Error-Dependent policy does not consider possible channel collisions and aggressively keeps sending more messages until one successful transmission, which is the main cause of frequent consecutive collisions even when the channel is in lightly loaded condition.

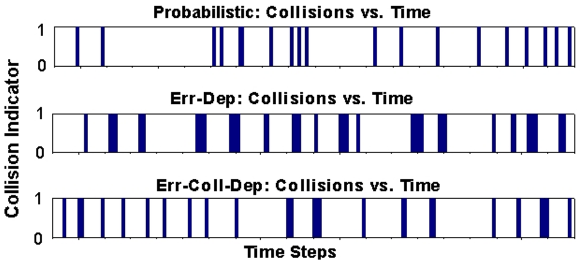

Fig. 4. Collision response of (a) Probabilistic, (b) Error-Dependent, and (c) Err-Coll-Dep policies with average rate around 0.2 pkt/node/sec, i.e. channel is lightly loaded. Corresponding collision ratios are 0.95%, 2.55%, and 1.30%.

On the contrary, Err-Coll-Dep policy alleviates this problem by considering channel collisions: when there is a collision, all nodes throttle their rates and thus reduce consecutive collisions. Therefore, the collision ratio of Err-Coll-Dep design is always lower than that of Error-Dependent in Fig. 3(b).

Note that, although Fig. 4 shows that Error-Dependent and Err-Coll-Dep policies have higher collision ratio in a lightly loaded channel, they still achieve lower tracking MSE than that of Probabilistic policy (compare their performance in Fig. 3(a) when average packet rate is below 0.5 pkt/node/sec). Again this shows that, the effectiveness of using a controlled communication policy comes from timely delivery of state information even though some channel bandwidth is sacrificed.

C. Multiplexing Gain

To get an insight into how efficient each communication policy can be in sharing the limited bandwidth resource, we study the performance of each policy in two different environments: a shared channel environment with possibility of collision (like in VANETs), a perfectly scheduled channel with no colision (denoted here as dedicated channels, possible in infrastructure networks). The multiplexing gain of the three policies is compared in Fig. 5. Here, multiplexing gain is defined as the tracking performance improvement when sharing the entire channel bandwidth instead of assigning a dedicated smaller portion of it to every node. For each policy, we compare the following two scenarios: 1) 10 dedicated channels, each with 10% bandwidth, assigned individually to 10 nodes; 2) a multi-access channel, with 100% bandwidth, simultaneously shared by all 10 nodes.

Fig. 5(a) shows that there is no multiplexing gain for Probabilistic policy. On the contrary, in Fig. 5(b) & (c), the multiplexing gain exists when Error-Dependent's packet rate is less than 0.6 pkt/node/sec, and when Err-Coll-Dep's packet rate is less than 1.5 pkt/node/sec. Once average rate is higher than those thresholds, channel collisions become dominant and the tracking performance degrades.

The multiplexing gain for Error-Dependent and Err-Coll-Dep exists because plants on different nodes are decoupled, and therefore not all nodes need to broadcast state information at the same time.

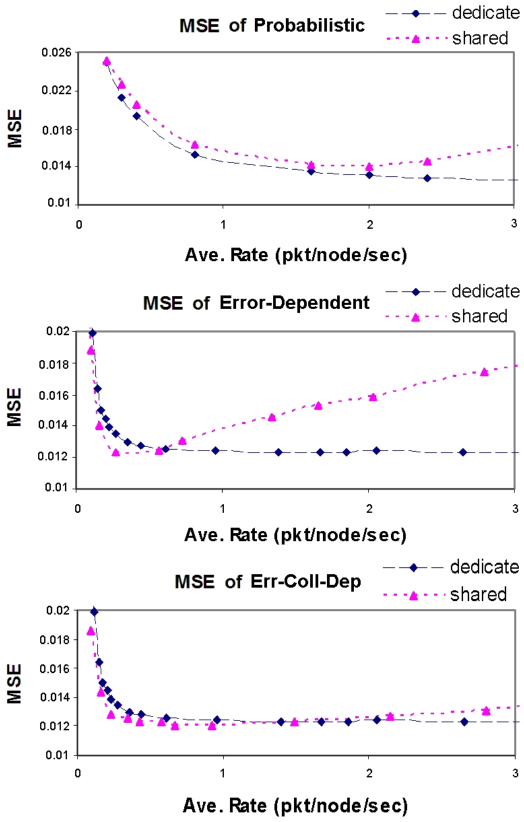

Fig. 5. Tracking MSE of a system using dedicated channels or a shared channel for (a) Probabilistic, (b) Error-Dependent, and (c) Err-Coll-Dep policies, of which (b), (c) show multiplexing gain.

In addition, Err-Coll-Dep policy has wider rate range of multiplexing gain than that of Error-Dependent policy because of lower collision ratio in the shared channel. By properly selecting parameters (β, γ) for Err-Coll-Dep, its tracking performance could be very close to the case that each node is assigned a dedicated channel (see Fig. 5(c)).

IV. PROPOSED RATE CONTROL ALGORITHM

Our proposed policy follows the same spirit of Err-Coll-Dep policy discussed in Section III; some minor modifications are introduced to address the fact that there is no explicit ACK for broadcast messages in DSRC.

On-demand Rate Control: First, the proposed algorithm runs every 50 msec due to 20 Hz sampling of vehicle's on-board sensors. At each time step $t \in N$, i.e. multiples of 50 msec, the j-th vehicle calculates transmission probability $p_j(t)$ based on *suspected* tracking error $\tilde{e}_j(t)$ on neighboring vehicles toward its own position *in Euclidean sense*. That is, we use a scalar error $\tilde{e}_j(t)$ to simplify the calculation of $p_j(t)$:

$$p_j(t) = 1 - \exp\left(\frac{-\alpha}{1 + \frac{\beta}{\gamma}\sum_{s=t-\gamma}^{t-1} \Lambda_j(s)} \times \tilde{e}_j(t)^2\right) \qquad (10)$$

where α and β are sensitivity parameters as in (8), and $\gamma = 20$ is the one-second time window to evaluate recent channel utilization. (In Section V, we use $\beta = 0$ for Error-Dependent and $\beta = 30$ for Err-Coll-Dep policy in simulation.) Here $\Lambda_j(s)$ represents CCA (Clear Channel Assessment) [16], at time step $s < t$, reported by j-th 802.11 PHY module: when channel is sensed busy, $\Lambda_j(s) = 1$; otherwise, $\Lambda_j(s) = 0$. Since a wireless transceiver cannot detect collisions for broadcast, we use channel utilization in (10) as a substitute for $c(t)$ in (8). This substitution is possible since collision ratio is a monotonically increasing function (which is not required to be known) of the channel utilization in CSMA/CA netowkrs. Based on $p_j(t)$ from (10), the j-th vehicle stochastically generates a packet with its own state information, and places the packet in its MAC layer queue.

Since there is no explicit ACK for broadcast in DSRC, after each transmission, we use PER (Packet Erasure Rate) $\Omega_j(t)$ to stochastically decide the *suspected* error $\tilde{e}_j(t)$ in (10),

$$\tilde{e}_j(t^+) = (1 - \zeta_j(t)) \times \tilde{e}_j(t) \qquad (11)$$

where $\zeta_j(t)$ is a Bernoulli trial with $\Pr(\zeta_j(t) = 0) = \Omega_j(t)$. If successful, i.e. $\zeta_j(t) = 1$, $\tilde{e}_j(t^+)$ is reset to zero; otherwise, $\tilde{e}_j(t^+)$ accumulates from $\tilde{e}_j(t)$ based on first-order kinematic model. Note that $\tilde{e}_j(t)$ is not the actual estimation error on neighbors of the j-th vehicle; it is only a measure used by j-th vehicle to adjust its own communication rate.

Finally, PER $\Omega_j(t)$ used by Bernoulli trials in (11) is derived on-the-fly by checking the inconsistency in sequence number of recently received packets from all corresponding neighbors within 1-second history log of vehicle j. That is, the j-th vehicle uses the number of lost packets divided by the number of total packets from a certain neighbor to infer the channel loss rate and $\Omega_j(t)$ is this measure averaged over all its neighbors at time t. By symmetric assumption of the channel, $\Omega_j(t)$ tells the j-th vehicle the likelihood of its previous transmission being erased in the channel.

In short, the proposed design has an on-demand nature since: 1) it increases information rate when a vehicle suspects the tracking error of neighboring vehicles toward itself has increased; 2) it throttles rate when channel seems to be in congestion to avoid further collisions (message loss and performance degradation). In Section V, this policy will be evaluated in large scale simulations and compared with existing solutions.

V. EVALUATION USING LARGE SCALE SIMULATIONS

In this section, we describe the settings of large scale VANET simulations, our implemented algorithms, and simulation results. In essence, we simulate microscopic highway traffics and compare tracking performances of different decentralized information dissemination policies.

A. Simulation Environment and Implemented Rate Control

As in [7], for a 1-km stretch of a 4-lane highway, we use trajectory files of position, speed, heading sampled at 20 Hz, produced by SHIFT traffic simulator. Total simulation time is 30 seconds for each run. This traffic scenario corresponds to an average flow rate of 2106 vehicles/hr/lane, an averaged velocity of 30 mile/hr, and a mean gap of 0.8 second between

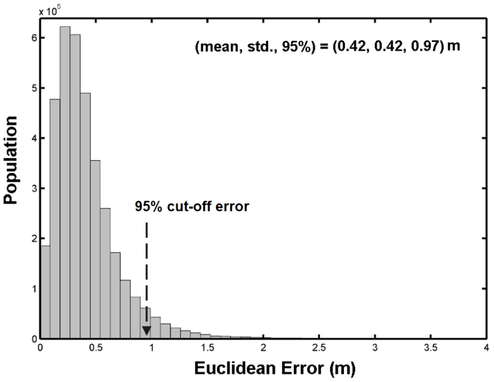

Fig. 6. Histograms of Euclidean error while using 100-msec beaconing, which exhibits an exponential tail. The 95% cut-off error is used as the accuracy measure; 95% of the error falls below this number.

vehicles in the longitude direction. During the simulation, every 50 msec, each vehicle gets a measurement of its own status from on-board sensors and decides whether to generate a packet or not, according to specified policy. The on-board measurement noise is modeled exactly as in [7], which was in turn based on experimental data [1].

Upon receiving information from the channel, each vehicle updates its estimation of neighboring vehicles. We choose the first-order kinematic model, i.e. a constant speed predictor, for tracking neighboring vehicles. Using this model, a vehicle is assumed to run with the same speed and heading after its last information broadcast was received by a certain predictor (on another vehicle).

SHIFT generated vehicle trajectories are used in our OPNET based simulations of the communications policies and DSRC. For DSRC simulation, we modified OPNET's 802.11a PHY module to work at 5.9 GHz with 10 MHz bandwidth and a DSRC channel model. We follow the best known DSRC channel model reported in [8] and simplify the channel at far distances as Rayleigh fading, instead of pre-Rayleigh. The reason for this simplification is that we are considering a straight highway scenario while [8] considers urban scenarios with intersections and corners, which lead to pre-Rayleigh fading observations. In addition, we assume that path loss exponent is always 2.31. The 802.11 transceiver operates with 3 Mbps raw data rate, -95 dbm Rx sensitivity, and 600 mW Tx power. The payload size of each message is 300 bytes.

In our evaluations we focus on the following decentralized policies:

- **Beaconing** at 100-msec interval as suggested by [20];
- **Probabilistic** policy which utilizes fixed transmission probability to generate packets;
- **Threshold** policy which triggers communication when perceived tracking error violates specified threshold [7];
- On-demand rate control: 1) **Error-Dependent** ($\beta = 0$), 2) **Err-Coll-Dep** ($\beta = 30$) policies.

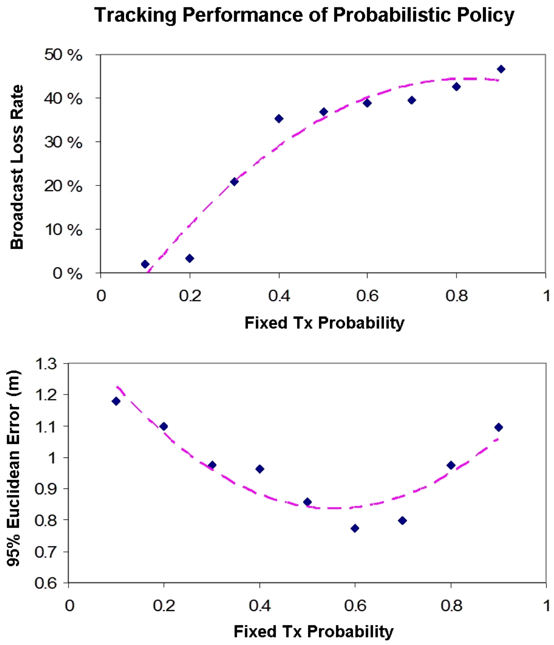

Fig. 7. Tracking performance while using Probabilistic policy: (a) broadcast loss rate; (b) 95% cut-off of Euclidean error. Broadcast loss rate increases as the Tx probability increases. The 95% cut-off error has a minimum of 0.8 meter when Tx probability is roughly 0.6. Similar to the U-shape curve in Fig. 4 (a), the same trade-off exists while selecting Tx probability.

B. Simulation Results and Performance Comparison

We evaluate the tracking performance of different decentralized policies in the proximity of each vehicle. The proximity of a subject vehicle is defined as the circular area of 150-meter radius to satisfy most safety applications identified in [20]. That is, a subject vehicle will try to estimate all vehicles within its 150-meter radius.

After each simulation run, we collect statistics from neighbors within this proximity of a certain vehicle and then calculate below performance metrics over all vehicles to explore law of large number:

- **Broadcast loss rate**: this metric reflects channel congestion status and reliability of transmitted messages;
- **Euclidean tracking error**: mean, standard deviation, and 95% cut-off error. Specifically, 95% cut-off error is our main performance measure, i.e. 95% of collected error population falls below this number, which represents the tracking accuracy of the policy.

First, a typical result from our simulations looks like Fig. 6, which shows the tracking error *in Euclidean sense* while each vehicle broadcasts its own state information at 100-msec interval. Shown results correspond to a broadcast loss rate of 32.98% which is the highest among all policies. In Fig. 6(c), its 95% Euclidean cut-off error, i.e. 0.97 meter, is used as baseline for comparison.

From Fig. 7 to Fig. 10, the performance metrics of different policies are plotted. In those figures, second order curves with least squared fitting are used to indicate the trends.

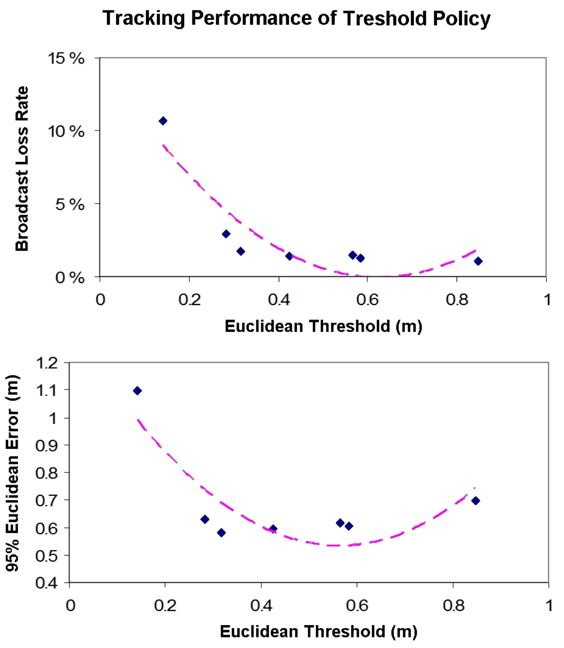

Fig. 8. Tracking performance while using Threshold policy: (a) broadcast loss rate; (b) 95% cut-off of Euclidean error. Since more communication is triggered by a smaller threshold, broadcast loss rate increases as threshold decreases. The 95% cut-off error has a minimum of 0.6 meter when threshold is around 0.5 meter.

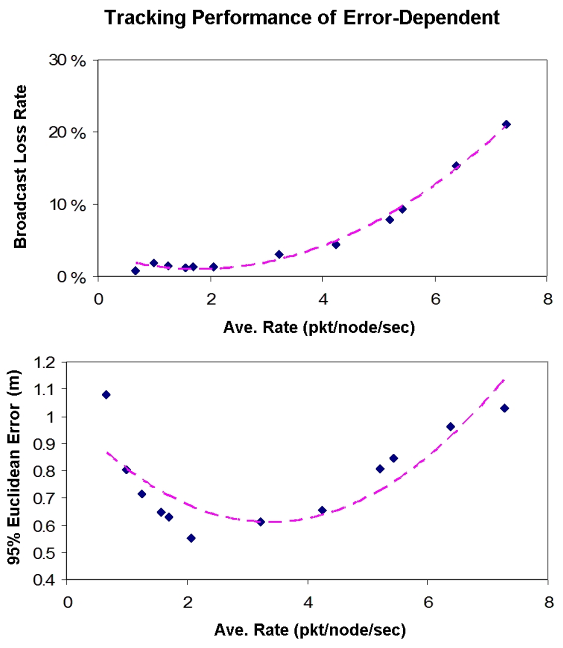

Fig. 9. Tracking performance while using Error-Dependent policy ($\beta = 0$): (a) broadcast loss rate; (b) 95% cut-off of Euclidean error. Its 95% cut-off error has a minimum of 0.6 meter when mean rate is around 4 pkt/node/sec, which corresponds to the optimal α value around 5.

TABLE I

ERROR SENSITIVITY USED IN ERROR-DEPENDENT POLICY ($\beta = 0$)

α	Ave. Rate (pkt/sec)	Std. Rate (pkt/sec)	95% Error (m)
0.1	0.761	0.115	0.975
0.5	1.522	0.191	0.764
1	1.853	0.223	0.591
5	3.746	0.277	0.612
10	4.614	0.294	0.623
20	5.778	0.323	0.946
25	6.213	0.442	0.964
30	7.651	0.498	1.109

Specifically, Table I & II compare mean and standard deviation of packet rate while using two on-demand policies.

For the Probabilistic policy, its corresponding performance is shown in Fig. 7 w.r.t. different Tx probabilities. Fig. 7(b), similar to Fig. 3(a), has a U-shape curve for 95% Euclidean error and the optimal transmission probability for this traffic scenario is 0.6. Its optimal tracking performance of 95% Euclidean error is around 0.8 meter. This policy achieves better tracking performance than beaconing by introducing randomness in its transmission behavior.

The performance of Threshold policy is shown in Fig. 8. With different thresholds, they trigger different amount of packet rate. As shown in Fig. 8(a), when threshold decreases, it triggers more transmission and thus causes more collisions in the channel. Its 95% Euclidean error also shows a U-shape curve and optimal threshold for this scenario is around 0.5 meter. Its minimum 95% error is 0.6 meter which is a further improvement from Probabilistic policy.

Results of Error-Dependent policy are shown in Fig. 9 and Tables I lists the error sensitivity α and corresponding tracking accuracy. In Table I, as we increase α, we increase mean and standard deviation of the packet rate; optimal rate is around 4 pkt/node/sec when roughly $\alpha = 5$. Its 95% Euclidean error in Fig. 9(b) shows a U-shape curve and its broadcast loss rate increases as the average rate increases. Error-Dependent can achieve similar min. 95% error (0.6 m) with the same average rate as the Threshold policy.

Finally, Fig. 10 and Table II show the tracking performance of Err-Coll-Dep policy. Compared with that of beaconing, Threshold, Error-Dependent, and Err-Coll-Dep policies all have fair amount of reduction in 95% cut-off error.

C. *Why Proposed Algorithm Works*

Although 802.11 CSMA/CA helps reduce collisions in DSRC channel, considered rate controls have similar behaviors as those policies analyzed in Section III. That is, a good rate control algorithm should consider tracking error and channel congestion at the same time.

For both Error-Dependent and Threshold policy, they only consider tracking error and ignore channel congestion. They can outperform beaconing and Probabilistic policy *if* we properly choose their parameters. However, optimal α and error threshold may vary greatly w.r.t. different vehicle flow densities or available channel resource; thus these parameters cannot be easily chosen beforehand.

Err-Coll-Dep policy can alleviate the above problem and make parameter design easier. By comparing Table I and II,

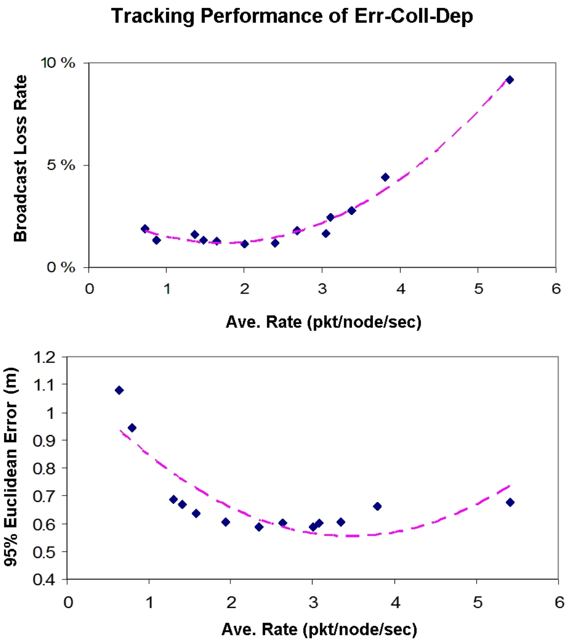

Fig. 10. Tracking performance while using Err-Coll-Dep policy ($\beta = 30$): (a) broadcast loss rate; (b) 95% cut-off of Euclidean error. This policy has similar curves as in Fig. 9, but its optimal α has a wide range of values (compare α values and corresponding performance in Table I and II).

TABLE II
ERROR SENSITIVITY USED IN ERR-COLL-DEP POLICY ($\beta = 30$)

α	Ave. Rate (pkt/sec)	Std. Rate (pkt/sec)	95% Error (m)
1	1.239	0.221	1.079
5	1.994	0.138	0.975
10	2.194	0.230	0.781
20	2.309	0.278	0.660
40	2.974	0.196	0.632
80	4.686	0.291	0.657
160	5.343	0.337	0.724
480	6.912	0.455	0.796

one can notice that, when $\beta = 30$, as long as we pick α large enough, i.e. $\alpha \geq 20$, the performance of Err-Coll-Dep policy can operate around the optimal point, i.e. 0.6 to 0.7 meter of 95% cut-off Euclidean error, for a wide range of α values.

Besides, in Err-Coll-Dep case (Table II), the fluctuation or standard deviation of rate does not increase as fast as that in Error-Dependent case (Table I). Our simulations of various highway traffic conditions, reported in [15], also confirm this robustness property of Err-Coll-Dep policy.

VI. CONCLUDING REMARKS

In this paper, we presented a decentralized information dissemination control algorithm for CAS applications in VANETs. This algorithm has an on-demand structure and adapts its communication rate to environment factors such as network congestion or sudden vehicle movement (which causes increased estimation error).

Our simulation results, both in Matlab and through large scale traffic/network simulations, confirm that the proposed algorithm requires less packet rate and yet attains much better tracking accuracy and robustness against channel congestion than the 100-msec beaconing policy suggested by [20]. This work is distinctly different from designs in [4], [5] and provides a new perspective of dealing with information dissemination in VANETs.

Throughout this paper, we assume a uniform Tx power for all vehicles while adjusting individual packet rate to enable statistical multiplexing. As enlightened by [14], there might be further performance improvement if a variable transmission power is used by each vehicle to explore spatial reuse. Our future work includes joint designs to adapt rate and power simultaneously in VANETs.

REFERENCES

[1] R. Sengupta, S. Rezaei, S. E. Shladover, J. A. Misener, S. Dickey, H. Krishnan, "Cooperative Collision Warning Systems: Concept Definition and Experimental Implementation," *Jour. of Intell. Transport. Sys.*, vol.11, no.3, pp. 143-155, 2007.

[2] X. Yang, J. Liu, F. Zhao, N. Vaidya, "A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning," *Proc. of 1st MobiQuitous*, Aug. 2004.

[3] J. Yin, T. Elbatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, T. Talty "Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks," *Proc. of 1st ACM VANET*, Sept. 2004.

[4] J. Mittag, F. Schmidt-Eisenlohr, M. Killat, J. Harri, H. Hartenstein, "Analysis and Design of Effective and Low-overhead Transmission Power Control for VANETs," *Proc. of 5th ACM VANET*, Sept. 2008.

[5] C. Robinson, L. Caminiti, D. Caveney, K. Laberteaux, "Efficient Coordination and Transmission of Data for Vehicular Safety Applications," *Proc. of 3rd ACM VANET*, Sept. 2006.

[6] S. Rezaei, R. Sengupta, H. Krishnan, X. Guan, "Reducing the Communication Required by DSRC-based Vehicle Safety Systems," *Proc. of 4th ACM VANET*, Sept. 2007.

[7] S. Rezaei, "Cooperative Vehicle Safety," Ph.D. dissertation, U.C. Berkeley, Aug. 2008.

[8] L. Cheng, B. E. Henty, D. D. Stancil, F. Bai, P. Mudalige "Mobile Vehicle-to-Vehicle Narrow-Band Channel Measurement and Characterization of the 5.9 GHz Dedicated Short Range Communication Frequency Band," *IEEE JSAC*, vol.25, no.8, Oct. 2007.

[9] Y. Xu, J. Hespanha, "Communication Logics for Networked Control Systems," *Proc. of the Amer. Contr. Conf.*, June 2004.

[10] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poola, M. J. Jordan, S. S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE Trans. Automat. Contr.*, vol.49, no.9, Sept. 2004.

[11] X. Liu, A. Goldsmith, "Kalman filtering with partial observation losses," *Proc. of Conf. Decision and Contr.*, Dec. 2004.

[12] C. L. Huang, R. Sengupta, "Analysis of Channel Access Schemes for Model-based Estimation over Multi-access Networks," *Proc. of IEEE Multi-session conference on Systems and Control*, Sept. 2008.

[13] C. L. Huang, R. Sengupta, "Decentralized Error-Dependent Transmission Control for Estimation over a Multi-Access Network," *VINT workshop, WICON*, Nov. 2008.

[14] P. Gupta, P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Info. Theory*, vol.46, no.2, Mar. 2000.

[15] C. L. Huang, X. Guan, Y. P. Fallah, R. Sengupta, H. Krishnan, "Robustness Evaluation of Decentralized Self-Information Dissemination Control Algorithms for VANET Tracking Applications," *submitted to VTC-fall 2009*, Feb. 2009.

[16] IEEE 802.11 WG, Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE, Aug. 1999.

[17] Wireless Access in Vehicular Environment (WAVE) in Standard 802.11, Specific Requirements: IEEE 802.11p/D1.0, Feb. 2006.

[18] IEEE 1609.3 Wireless Access in Vehicular Environment: Networking Services, 2007.

[19] California PATH SHIFT, http://path.berkeley.edu/SHIFT/.

[20] Vehicle Safety Communications Consortium (VSCC), "VSC Project, Task 3 Final Report: Identify Intelligent Vehicle Safety Applications Enabled by DSRC," 2005.