LIMERIC: A Linear Message Rate Control Algorithm for Vehicular DSRC Systems

John B. Kenney  
Toyota InfoTechnology Center, USA  
jkenney@us.toyota-itc.com

Gaurav Bansal  
Toyota InfoTechnology Center, USA  
gbansal@us.toyota-itc.com

Charles E. Rohrs  
Toyota InfoTechnology Center, USA  
crohrs@mit.edu

ABSTRACT

Wireless vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication holds great promise for significantly reducing the human and financial costs of vehicle collisions. A common characteristic of this communication is the broadcast of a device’s core state information at regular intervals (e.g. a vehicle’s speed and location, or a traffic signal’s state and timing). The aggregate of these uncoordinated broadcasts will lead to channel congestion under dense traffic scenarios, with a resulting reduction in the effectiveness of the collision avoidance applications making use of the transmitted information. Active congestion control using distributed techniques is a topic of great interest for establishing the scalability of this technology for deployment. This paper defines a new congestion control algorithm that can be applied to the message rate of devices in this vehicular environment. While other published approaches rely on binary control, the Linear MESSAGE Rate Integrated Control (LIMERIC) algorithm takes advantage of full precision control inputs that are available on the wireless channel. The result is provable convergence to fair and efficient channel utilization in the deterministic environment, under simple criteria for setting adaptive parameters. This “perfect” convergence avoids the limit cycle behavior inherent to binary control. We also discuss several practical aspects associated with implementing LIMERIC, including: guidelines for the choice of system parameters to obtain desired utilization outcomes, a gain saturation technique that maintains robust stability under all conditions, convergence with asynchronous updates, and the implications of measurement noise for statistical properties of convergence. The paper illustrates key analytical results using MATLAB numerical results, and employs standard NS-2 simulations to demonstrate the performance of LIMERIC in several high density scenarios.

Categories and Subject Descriptors

C.2.1 [Network Architecture and Design]: Wireless communications; C.4 [Performance of Systems]: Performance attributes

General Terms

Algorithms, Performance, Design

Keywords

DSRC, linear rate control, congestion control, IEEE 802.11, NS-2

1. INTRODUCTION

To enable vehicular safety applications that can prevent collisions, and other Intelligent Transportation Systems (ITS) applications, the US Federal Communication Commission (FCC) has allocated 75 MHz of spectrum in the 5.9 GHz band for Dedicated Short Range Communications (DSRC) among vehicles (V2V) and between vehicles and roadside infrastructure (V2I) [1, 2]. The technology is being standardized in IEEE 802.11p [3, 4], the IEEE 1609 family of standards [5, 6], and SAE [13, 14]. Similar spectrum allocations and standards development are being pursued in Europe.

DSRC uses the 802.11 medium access control (MAC) protocol, which is based on carrier sense multiple access/collision avoidance (CSMA/CA). The basic rule is that when a node has a packet to send it first listens to the channel. If the channel is idle, the node transmits the packet. If channel is busy the node waits a random backoff time before transmitting the packet. MAC frame collision probabilities are reduced via this mechanism, but they remain nonzero. There are various types of collision, for example simultaneous countdown, hidden node, and same carrier sense time. The probabilities of these collision events increase with the number of nodes competing to access the wireless channel.

DSRC collision avoidance in the United States is based on a paradigm of frequent vehicle state broadcasts from each vehicle. These utilize the Basic Safety Message (BSM) format [13], which includes core state information such as location, speed, and brake status, as well as path history and prediction. These messages are typically on the order of 300-400 bytes, including all lower layer overhead [7]. By default they are transmitted over a 300-500 meter range, with a 6 Mbps data rate, and a 10 Hz message rate. The communication channel bandwidth is 10 MHz (DSRC Channel 172, i.e. the “safety channel”). In a high-density traffic scenario, it will not be unrealistic to have 150 or more vehicles present within the reception range of a transmitting vehicle, with even more vehicles within the interference range. In these scenarios the packet error rate (PER) will increase rapidly with vehicle density, as a result of increased frame collision probabilities. Field tests conducted with DSRC radios measured a PER as high as 79.5% for links with Received Signal Strength between -80 dBm and -85 dBm, and with 360 emulated vehicles [12]. Increased probabilities of packet loss make it more difficult for a vehicle to model the movement of its neighbors and recognize potentially dangerous situations. Hence, congestion on the DSRC safety channel can severely impact safety application performance. As a result, techniques to control and mitigate channel congestion are among the highest priorities for moving DSRC technology toward deployment. The VSC3 consortium is currently engaged in the V2V-Interoperability project in cooperation with the US Department of Transportation [15]. One of the primary goals of this project is to ensure the scalability of DSRC for safety communication.
DSRC congestion control has been the subject of recent research. In [9], a congestion control protocol is proposed that adapts message rate of a vehicle according its dynamics in order to maintain an acceptable tracking accuracy by other vehicles. Then, transmit power is adapted to maintain the channel load at a target level. In [10], a distributed transmit power control method is proposed, which reduces the power of safety message transmissions during congestion in order to control the load placed on the DSRC channel. In [11] a message rate control based approach is proposed to adapt the BSM transmission rate (frequency) based on a binary comparison between measured channel load and a target threshold. Binary message rate control is also the subject of [12], in which the authors propose using an additive increase multiplicative decrease (AIMD) message rate update mechanism for DSRC vehicular safety communication. They present results from prototype radio tests and computer simulations that illustrate effective message rate control for hundreds of emulated or simulated vehicles.

In this paper we propose a message rate control algorithm that improves on the binary approach in [11] and [12]. It uses linear feedback to adapt the message rate, and thus avoids the limit cycle behavior inherent in binary control. In Section 2 we introduce the algorithm and use analysis to prove deterministic convergence as a simple function of adaptation parameters, assuming knowledge of aggregate message rate in a single collision domain. Convergence is shown both with regard to fairness (vehicle message rates converge to each other) and to efficient channel utilization relative to a target. In addition to the basic analysis, including proof of convergence, Section 3 provides detailed discussions of several practical considerations in using LIMERIC. The first is a consideration of the network utilization in steady state, and steps that can be taken to select algorithm parameters to achieve desired utilization outcomes. The second examines the stability criteria and discusses techniques that can be employed to guarantee stable convergence under all operating environments. In particular, a gain saturation technique is introduced that is only engaged when device density exceeds conservative estimates. With this technique, LIMERIC enjoys perfect convergence in the vast majority of cases, but falls back to stable binary limit cycles in the rare case. The basic convergence proof assumes synchronous update behavior among the devices in a network. The paper also considers the more easily achieved asynchronous update case, and offers arguments for why stability is even more robust in that regime. Convergence under the asynchronous update case is not within the scope of the formal analysis of this paper, however. Finally, the paper considers the form of the input to LIMERIC. The convergence analysis assumes the aggregate message rate of all devices is observable at each device. The paper then considers the more realistic use of a standardized channel activity measure as a substitute for aggregate rate, and shows conditions under which the analytical results remain valid. While this algorithm is applicable in a variety of channel congestion environments, we focus on the DSRC safety channel use case. In section 4 we present computer simulation results that illustrate various facets of system performance, including the effects of synchronous and sequential update cases with MATLAB, and more practical asynchronous probabilistic updates using the NS-2 simulator. We summarize results and present conclusions in Section 5.

### 2. LIMERIC-A LINEAR CONTROL ALGORITHM

In general the objectives of an adaptive congestion control approach are threefold: 1) to be able to converge to a desired channel utilization level (channel load), 2) to achieve local fairness among immediate neighboring vehicles, and 3) to achieve global fairness among all vehicles contributing to congestion. The scope of this paper is limited to objectives 1 and 2, i.e. we simplify the network to a single collision domain where all the nodes measure the same channel load. For such a network, we aim to achieve fairness such that all the nodes can converge to same message rate. The algorithm defined in this paper uses only local information. Objective 3, which deals with the global fairness issue for a larger network with hidden nodes, is briefly discussed in 3.4 and will be studied as future work.

In most situations concerning rate adaptation in packet networks, each user has limited information about the amount of congestion in the network. Usually, this information can be summarized by a binary variable indicating either that the network is congested or not. For example, a code division multiple access (CDMA) network often uses the inability to access the network as an indicator of congestion and a successful access as an indicator of an uncongested state. In the Internet, the Transmission Control Protocol (TCP) uses the absence or late arrival of an acknowledgment (ACK) as an indication of congestion, and the timely arrival of an ACK as an implicit indication of the availability of additional capacity [20]. Routers capable of Explicit Congestion Notification (ECN) [21] can give the TCP source an explicit binary congestion signal. At layer 2, Asynchronous Transfer Mode (ATM) and Frame Relay provide explicit binary indicators of congestion [16, 17]. The limitation of binary information feedback leaves the rate control algorithm with certain known problems concerning convergence to fairness unless the distributed users are assumed to act synchronously [18].

Here we introduce the Linear MEssage Rate Integrated Control (LIMERIC) algorithm, which considers the situation where each user has more information about the state of the congestion in the network. In particular, we assume that each user can measure the fraction of network capacity that is in use at each moment. To model this mathematically, we let \( r_j(t), j = 1, 2, ..., K \) be a number between zero and one representing user \( j \)'s rate of transmission as a fraction of the total channel capacity assuming there are \( K \) active users on the network. We assume that each user can measure \( r(t) = \sum_{j=1}^{K} r_j(t) \), the total fraction of the capacity of the network by all users together. While \( r(t) \) in general varies with measurement location, in this section we make the simplifying assumption that all vehicles measure the same value. In the NS-2 simulation results presented in Section 4, we relax this assumption by utilizing a per-vehicle statistical carrier sense function, but still with essentially no hidden nodes; the algorithm retains excellent convergence and fairness characteristics. We discuss below the option of estimating \( r(t) \) based on the IEEE 802.11 Clear Channel Assessment (CCA) function [3]. We assume that \( K \) is known for analysis but we expect to create algorithms whose performance does not depend on the exact knowledge of \( K \). We also assume that each user is working towards the same aggregate message...
rate goal, represented by \( r_g \), and that it is desired to share this goal usage equally. We need to adapt the rates independent of exact knowledge of \( K \) since, if \( K \) were known with certainty, each user could simply set her rate at \( r_g/K \). Variations involving priority systems or weighted fair sharing are possible to include with modifications, but we will consider the base equal sharing case here.

We can consider the difference \( r_g - r(t) \) as the error in the total rate being used in the network. Each user can adjust her own rate linearly with respect to the full precision of this error. Compared to a binary control, which would use only the sign of the error, such a linear scheme has an advantage in that it can converge to fair rates with no limit cycles and that such convergence can be analyzed using tools from linear systems theory. Since each user updates her rate episodically we consider this system in discrete time. We initially consider the synchronous update case, i.e. all nodes perform updates at the same time instants.

Let each user update her rate by:

\[
r_j(t) = (1-\alpha) r_j(t-1) + \beta (r_g - r(t-1))
\]  

(2),

where \( r(t) \) is given by Eq.1, \( 0 < \alpha < 1 \), \( \beta > 0 \).

In this equation, the parameter \( \alpha \) creates an exponential forgetting function that promotes fair convergence. The message rate offset is linear in the total rate error with scale factor \( \beta \). The characteristics of the algorithm are analyzed below as a function of \( \alpha \) and \( \beta \).

An important quality of what we refer to as the synchronous case is that all users employ the same measured \( r(t-1) \) that includes all users’ rate adjustments from the preceding time, \( t-1 \), but none of the updated rates \( r_j(t) \) from other users. This is in contrast to what we call the sequential case where user 2’s update is based on a measured \( r \) that includes an intermediate update in user 1’s rate, and user 3 includes both user 1 and user 2’s most recent updated rates in her calculation and so on.

The system defined by Eqs. 1 and 2 can be analyzed by forming a vector

\[
\tilde{r}(t) = \begin{bmatrix} r_1(t) & r_2(t) & \ldots & r_K(t) \end{bmatrix}^T
\]

(3).

With this, Eqs. 1 and 2 become:

\[
\tilde{r}(t) = A \tilde{r}(t-1) + b r_g
\]

(4),

where

\[
A = \begin{bmatrix}
1 - \alpha - \beta & -\beta & \ldots & -\beta \\
-\beta & 1 - \alpha - \beta & \ldots & -\beta \\
\vdots & \vdots & \ddots & \vdots \\
-\beta & -\beta & \ldots & 1 - \alpha - \beta
\end{bmatrix}
\]

and

\[
b = \begin{bmatrix} \beta \beta \ldots \beta \end{bmatrix}^T
\]

(5).

For a square matrix \( A \), a non-zero vector \( v \) is an eigenvector of matrix \( A \) if there exists a scalar \( \lambda \) (eigenvalue) such that \( Av = \lambda v \). Linear system theory can be used to show that the matrix \( A \) above has one eigenvalue at \( z=1-\alpha - K \beta \) and \( (K-1) \) eigenvalues at \( z=1-\alpha \). A set of unnormalized eigenvectors are:

\[
\begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \end{bmatrix} , \quad \begin{bmatrix} 1 \\ -1 \\ 0 \\ 0 \end{bmatrix} , \quad \begin{bmatrix} 1 \\ -2 \\ 0 \\ 0 \end{bmatrix} , \quad \begin{bmatrix} 1 \\ -3 \\ 0 \\ 0 \end{bmatrix} , \quad \ldots 
\]

where the first eigenvector is associated with eigenvalue \( 1-\alpha - K \beta \) and the other eigenvectors are associated with eigenvalue \( 1-\alpha \).

A linear discrete-time system is stable if the poles are inside the unit circle, which in this case corresponds to eigenvalues taking on values between +1 and -1. Given the constraints \( 0 < \alpha < 1 \) and \( \beta > 0 \), and with the added condition,

\[
1 - \alpha - K \beta > -1 \text{ or } \alpha + K \beta < 2 \quad (6),
\]

the system represents an asymptotically stable discrete-time system with a single convergence point. The set of equations:

\[
r_j(t) = \frac{\hat{b} r_g}{(\alpha + K \beta)}, \quad j = 1,2,\ldots,K \quad (7),
\]

can be shown to be a steady state solution to equation (2), and thus they are the convergence point. Note that this solution is fair, i.e. the rates \( r_m \) and \( r_n \) are equal in steady state. The dynamics of the convergence to fairness are controlled by the eigenvalues given by \( 1-\alpha \) and is assured independent of \( K \), the number of users. The speed of this convergence to fairness is discrete exponential as \( (1-\alpha)^j \) from any initial condition of unfairness. This eigenstructure leads us to derive directly from Eq. (2), for any two users:

\[
r_m(t) - r_n(t) = (1-\alpha)(r_m(t-1) - r_n(t-1))
\]

(8)

or

\[
r_m(t) - r_n(t) = (1-\alpha)^j (r_m(0) - r_n(0))
\]

(9)

Now consider the dynamics of the total rate of usage on the channel which is controlled by the eigenvalue at \( z=1-\alpha - K \beta \). If we sum across the individual \( r_j \) in the vector Eq. 4, we get:

\[
r(t) = (1-\alpha - K \beta) r(t-1) + K \beta r_g
\]

(10)

This difference Eq. can be solved to get:

\[
r(t) = r_f + (1-\alpha - K \beta)^j (r(0) - r_f)
\]

(11),

where

\[
r_f = \frac{K \beta r_g}{\alpha + K \beta}
\]

(12)

Therefore \( r \) converges to \( r_f \) when inequality (6) is satisfied. Comparing Eqs. (7) and (12), it is evident that the fairness property ensures that each \( r_j \) converges to \( r_f/K \).

To maintain stability of the total rate of usage on the channel, \( \beta \) must be chosen so that the stability condition of Eq. (6) is met for the largest envisioned value of \( K \). However, if \( \beta \) is chosen to
be small (to allow for a large $K$) and the number of users happens to be small so that $K\beta$ is small compared to $\alpha$ the overall usage of the channel will be smaller than desired, as seen in equation (12). Of course, $\alpha$ could be chosen to be small but this then slows down the rate of convergence both to fairness and to having the channel usage close to the desired rate. We believe that these kinds of tradeoffs are present in all such adaptive algorithms and that it is a strength of examining the linear algorithm to allow the analysis that makes the tradeoffs clear.

We can think of two ways to address the need for stability while the condition in inequality (6) involves the unknown quantity $K$. One can easily think of adapting $\beta$ in a slower acting loop. Every node is measuring the overall rate $r(t)$. If $r(t)$ is significantly less than $r_g$, this indicates that $\beta$ is too small and should be increased. When $\beta$ is too large all nodes will see a large and oscillating $[r_g - r(t)]$ and would know that $\beta$ should be decreased. As long as this $\beta$ adaptation occurs more slowly than the original adaptation of $r$, this should work well.

A second method of ensuring stability is to apply a maximum magnitude saturating non-linearity to the adjustment term $\beta(r_g - r(t-1))$ in Eq. (2). Then, when $K$ is too large for the chosen static $\beta$ the system will maintain stability and react more like the systems with binary feedback, but with the advantage that it can return to a response with no limit cycles when either $\alpha$ or $\beta$ decreases to satisfy inequality (6). These two methods appear to be complementary. In this paper we employ a static $\beta$ and explore the use of the gain saturation approach and various other practical aspects like selection of system parameters, asynchronous updates, implementation in distributed systems in detail in Section 3.

3. DISCUSSIONS

3.1 Selection of system parameters

This section presents suggestions for how the adaptive parameters $\alpha$ and $\beta$ should be set, as well as the goal rate $r_g$. This paper only considers the case where all three of these parameters are constants, but the work could be extended to allow for dynamic parameters.

These parameters, along with the number of vehicles $K$, determine the following algorithm behaviors:

- Stability – inequality (6)
- Convergence equilibrium – Eq. (12)
- Convergence speed – Eqs. (9) and (11).

Ensuring stability is the highest priority, which requires $\alpha + K\beta < 2$. $\beta$ has a more important role in this expression than $\alpha$, since it needs to counter the effect of $K$. Therefore, $\alpha$ is selected more for its impact on equilibrium and convergence speed, discussed below. Given that $\beta$ is chosen to be constant, it is tempting to consider whether there exist a largest $K$, i.e. $K_{\text{max}}$ and to set $\beta$ proportional to $1/K_{\text{max}}$. However, while one can attach loose probabilities to various neighborhood sizes, it is not possible to identify a maximum value. Furthermore, choosing $\beta$ to be very small has negative consequences for the other behaviors, as explained below. So, in this paper $\beta$ is set to ensure stability over a broad range of neighborhood sizes, but to allow for a small probability that inequality (6) is not satisfied. Section 3.2 provides a means of guaranteeing stability for that low probability case.

The choice of a value for $\alpha$ in the range 0 to 1 involves a trade-off that can best be seen in Eqs. (9) and (12). From Eq. (9) it is seen that the speed of convergence to fairness depends entirely on $\alpha$, and that larger $\alpha$ corresponds to faster convergence. From Eq. (12) we see that the aggregate rate equilibrium point $r_j$ depends on the unknown $K$. In general it may be desired that $r_j$ varies as little as possible with $K$. Choosing $\alpha$ small, i.e. closer to 0, promotes that objective. As a compromise, in this paper $\alpha$ is set to be 0.1.

Returning to the selection of $\beta$, from Eq. (11) we see that the aggregate rate converges with time constant $1-\alpha-K\beta$. For values of the time constant between 0 and 1 the system converges monotonically in a noiseless environment. For values between 0 and -1, the system converges, but in an oscillatory way. For values less than -1 the system is unstable. The choice of $\beta$ value should be small enough to provide stability for realistic values of $K$ and large enough to support desirable convergence behavior even when $K$ is small. In [8] it is shown that when $K=150$ vehicles communicate V2V safety messages at 10 Hz and 20 dBm, the system can be said to be moderately congested, with channel activity measures between 60 and 70%. For smaller values of $K$ the relationship between $K$ and the IEEE 802.11 frame collision rate can reasonably be approximated as linear. For larger values of $K$ the frame collision rate increases much more rapidly. These qualitative observations are typical for characterizing the onset of congestion. In this paper, $K=150$ is somewhat arbitrarily defined as the V2V safety congestion onset threshold, and $\beta$ is set to 1/150.

For the parameter assignment ($\alpha=0.1$, $\beta=1/150$), the system behavior of LIMERIC is as follows:

- From inequality (6), stability is guaranteed for $K < 285$.
- From Eq. (12), for $K < 285$ the aggregate rate converges to $r_f = r_g \left(\frac{1 + \frac{15}{K}}{15}\right)$.

- From Eq. (11), for $K < 285$ the system convergence time constant is $|0.9 - (K/150)|$, i.e. it is never slower than a system with constant 0.9, and it is fastest when $K$ is approximately 135.

Finally, from Eq. (13) it is apparent how to choose the goal rate $r_g$ in order to achieve a desired equilibrium $r_j$. Ideally this equation would be independent of the unknown $K$, but there remains some dependence. For large values of $K$, $r_j \approx r_g$. For smaller values of $K$, $r_j$ converges to a smaller fraction of $r_g$.

This is not a concern for V2V safety communication, however, since it is a generally recognized that there is a small diminishing return for any marginal increase in vehicle message rate above approximately 10 msg/sec. V2V safety messages are typically on the order of 3 Kbits, so there is room for approximately 2000 msg/sec in a channel using a 6 Mbps data rate. Imagine that $r_g$ is set to 0.6, equivalent to about 1200 msg/sec. From Eq. (13), if $K$ is small, e.g. 15, the equilibrium $r_j$ is only 0.55 $r_g$, or 0.3. This corresponds to an aggregate rate of about 600 msg/sec. From Eq. (9) these are shared fairly among the 15 vehicles, i.e. the individual vehicle rate in steady state, $r_j = r_j/K, \forall j=1 \ldots 15$. So, even though $r_j$ is only half of $r_g$, it still allocates approximately 40 msg/sec/vehicle, far in excess of the maximum “useful” rate.

In practice, each vehicle’s rate will be capped, e.g. at 10 msg/sec (or equivalently, 0.005 of the channel’s 2000 msg/sec capacity),
external to the adaptive algorithm. In that case, if \( r_g \) is set to 0.6, then using Eq. (13) the aggregate steady state message rate is:

\[
rf = K \cdot \min \left[ 0.005 \frac{0.6}{K + 15} \right].
\] (14)

For values of \( K < 10^5 \) the per-vehicle maximum rate applies. When \( K > 10^5 \) the adaptive equilibrium is such that each node converges to a rate below its cap, and \( rf \) is within 87.5% of \( r_g \).

MATLAB plots are presented below to demonstrate the unstable case. The maximum and minimum data rates for each node have been assumed to be 10 msg/sec and 0 msg/sec respectively. We have set the values of \( \alpha, \beta \), and \( r_g \) to be 0.1, 1/150, and 0.6 (equivalent to 1200 msg/sec) respectively. In Fig. 1, 200 nodes are participating in congestion control and hence inequality in Eq. (6) is satisfied. It can be observed from Fig. 1 that LIMERIC converges to 5.58 msg/sec for every node (only selected nodes are shown), as governed by Eq. (7). In Fig. 2, we change the total number of nodes to 300, so now the inequality (6) is not satisfied and the algorithm becomes unstable. Hence, for \( \beta = 1/150 \), there can be scenarios with large number of nodes (>285) where the algorithm can become unstable. Hence, next we modify LIMERIC using the idea of gain saturation.

**Fig1:** LIMERIC for total nodes = 200, inequality (6) satisfied

**Fig2:** LIMERIC for total nodes = 300 without gain saturation, inequality (6) not satisfied

### 3.2 Gain Saturation to Prevent Instability

The stability condition for LIMERIC is shown in inequality (6), as a function of \( a, \beta, \) and \( K \). Since \( K \) is unknown and can vary quite a bit in dynamic environments, it is not possible to choose static values of \( a \) and \( \beta \) for which stability is assured. In a distributed and unreliable wireless network, it is not straightforward to estimate \( K \). One strategy for assuring stability is to adjust \( a \) and/or \( \beta \) dynamically in response to observed message rate convergence or explicit estimates of \( K \) in order to keep inequality (6) satisfied. The \( a \) and/or \( \beta \) can be varied in a slower adaptation loop as compared to the message rate adaptation of Eq. (2). The feasibility of this approach is promising, and is left for future study. The strategy adopted in this paper is to introduce stability-preserving adaptation constraints to the basic update Eq. (2). This section describes this gain saturation approach, and illustrates the stabilizing effect that it achieves.

We modify the LIMERIC update Eq. (2) as follows:

\[
rf(t) = (1 - \alpha \epsilon_j(t-1)) + \beta \min[X, \beta |r_g - r(t-1)|]
\] (15)

In other words, if the magnitude of the update offset, \( \beta \cdot (r_g - r(t-1)) \), exceeds a threshold \( X \), the update is limited to \( \pm X \), with the sign reflecting the error term \( r_g - r(t-1) \). Under conditions that would otherwise cause instability, i.e. inequality (6) is not satisfied, the system with saturation non-linearities will converge instead to limit cycle behavior. That is, each vehicle’s message rate with oscillate with a fixed amplitude around a point. The trapping of the behavior within the limit cycle (as opposed to a growing exponential or oscillation) and the size of the limit cycles can be established by using Describing Function Analysis, a well-known control theory technique [19]. The logic of the theory is that when the error signal is large, the saturation value produces the equivalent of a small gain and the system tends to converge to smaller error signals. However, if the linear gain \( \beta \) in Eq. 15 is too large, the system will tend to diverge from small error signals. The limit cycle results when these two effects balance. The behavior of the system when there is a saturation non-linearity with too large a gain \( \beta \) in the linear region is similar to the behavior of an adaptive system with binary feedback, e.g. TCP. If the gain in the linear region is small enough to satisfy inequality (6), after a possible initial transient the system converges as if the saturation were not present. Therefore, adding the saturation non-linearity provides stability insurance in the unlikely case that inequality (6) is not satisfied, but it does not interfere with the perfect convergence shown in Eq. (7) when inequality (6) is satisfied.

In the prior section it was noted that in practice a V2V safety system will likely impose a maximum message rate per node. Including gain saturation logic in the LIMERIC update imposes \( X/\alpha \) as a maximum steady state fraction of channel capacity that a node can use (corresponding to 2000*X/\alpha msg/sec rate). If a particular \( r_j(t) \) takes on a larger value, say \( (X + \epsilon_j/\alpha) \), then after the next update it will necessarily contract to a value no higher than \( (1 - \epsilon_j/\alpha) (X + \epsilon_j/\alpha) + X = [X + \epsilon_j/\alpha (1 - \epsilon_j/\alpha)]/\alpha < r_j(t) \). The rate will continue to contract in subsequent updates until it is no greater than \( X/\alpha \). Setting the saturation gain \( X \) too small can impose an unwanted rate limit, and can also slow convergence. The value of \( X \) is also proportional to the limit cycle peaks, so it should also not be set too large. In this paper \( X \) is set to \( 0.005 \) (corresponding to 1 msg/sec). For \( \alpha = 0.1 \), this creates a maximum per-node channel...
usage of .005 (corresponding to 10 msg/sec), which is consistent with common industry assumptions about maximum message rates.

Note that while the positive and negative gain saturation values have equal magnitude in Eq. (15), it is also possible to use asymmetric saturation, e.g. X_positive > 0 and X_negative < 0. This does not impact stability, but alters the precise form of the limit cycle and the average message rate in steady state. Note that the maximum message rate is only a function of the positive saturation gain.

Fig. 3 is a MATLAB plot of the convergence trajectory for several nodes under a LIMERIC configuration identical to that of Fig. 2, except that the update includes gain saturation, i.e. it uses Eq. (15) rather than Eq. (2). Comparing Figs. 2 and 3, the stabilizing effect of gain saturation is evident; instead of diverging, the trajectories converge to a limit cycle.

3.3 Asynchronous Case

So far we have assumed that all nodes participating in congestion control are synchronized, but in practice it might be complicated to achieve synchronization. In general, nodes will perform message rate updates asynchronously, i.e. they will not update at the same time instants. So, the synchronous update analysis in Section 2 no longer strictly holds.

The mechanism that can cause instability when the algorithm is implemented synchronously gives reason to believe that convergence may occur more easily when an algorithm is implemented asynchronously. In the synchronous case at each update instant, all K nodes move their rates in the same direction. The analysis then depends on the potentially large gain \( K*\beta \) appearing in Eq. (11) and the system's eigenvalue as \((1-\alpha-K*\beta)\).

If the synchronicity is broken up, in each period, some nodes will increase rate while others decrease rate, making the effective gain smaller and convergence more likely for larger numbers of nodes.

A scenario that can be realized more easily than synchronous updates is that all vehicles use the same update period, but that each vehicle’s specific update time within the period (i.e. phase) is independent of the others. In this special case of asynchronous updates, the vehicles perform updates in a sequential pattern across the period, and that pattern is repeated in each period. One can assign vehicle index 1 to the first vehicle to update in the period, index 2 to the next vehicle and so on. When vehicle 2 performs its update, the aggregate rate it uses includes vehicle 1’s most recent update.

Fig. 4 illustrates the more robust stability that LIMERIC experiences under the sequential update regime. Using the same parameters as the configuration for Figs. 2 and 3, i.e. with inequality (6) not satisfied, it shows once again the convergence trajectory for a subset of nodes. But now, rather doing a synchronous update, nodes are performing their updates in a sequential manner, as described above. In order to explore the stability properties of this case, the updates do not use gain saturation, i.e. they follow Eq. 2, not Eq. 15. While this scenario produces instability in the synchronous update case of Fig. 2, for sequential updates the algorithm is stable. Given the parameters of this example, Eq. 12 identifies the equilibrium that would be expected for the synchronous update case if it could be stabilized, i.e. \( r_f = (300*0.6*1/150)/(0.1 + 300/150) = 0.57 \). The per-vehicle message rate corresponding to that fraction of aggregate channel capacity is approximately 2000*0.57/300 = 3.8 msg/sec, which is roughly the steady state rate to which the trajectories converge in Fig. 4. Further examples of robust stability for non-synchronous updates are provided in Section 4, both for sequential update cases and for more general asynchronous cases.

Gain saturation was introduced in Section 3.2 to provide guaranteed stability, for any \( K \). By contrast, the enhanced stability observed for asynchronous updates is expected to have a limit. So, gain saturation should still be applied in practice. A rigorous analysis of the asynchronous update case is beyond the scope of this paper, and will be a topic of future research.

Gain saturation was introduced in Section 3.2 to provide guaranteed stability, for any \( K \). By contrast, the enhanced stability observed for asynchronous updates is expected to have a limit. So, gain saturation should still be applied in practice. A rigorous analysis of the asynchronous update case is beyond the scope of this paper, and will be a topic of future research.

3.4 Implementation in Distributed System (using CBF)

The analysis in Section 2 utilizes \( r(t) \), the sum of the individual vehicles’ current message rates \( r_j(t) \). The earlier analysis assumes that \( r(t) \) is available at each vehicle. This could be accomplished, with non-trivial complexity, by having each node include its current rate \( r_j(t) \) in its safety message broadcasts, though the unreliable channel would make the aggregate rate thus obtained a noisy estimate of the exact \( r(t) \). A simpler approach takes advantage of a readily available measure from which \( r(t) \) can be derived. The IEEE 802.11-2007 standard [3] defines the Clear
Channel Assessment (CCA) function to determine at any given time whether the channel is classified as idle or busy. This CCA function is measured in every compliant IEEE 802.11 chip. The idle or busy state can be tracked over an interval of time to determine the ratio of time that the channel is busy. In this paper the interval measure is referred to as the Channel Busy Function (CBF), and it is a function of time \( t \). CBF is also referred to in other documents as the Channel Busy Ratio or the CCA Busy Fraction.

A well understood property of the CSMA mechanism used in MAC protocols like IEEE 802.11 is that there is a one-to-one relationship between the “offered load” of a set of nodes and the resulting CBF [22]. Offered load is the aggregate quantity of data transmitted on the channel. In the V2V safety communication regime, where each message is broadcast and there are no Acknowledgment frames or retransmissions, offered load is equivalent to \( r(t) \). Therefore, there exists a one-to-one function that maps from \( CBF(t) \) to \( r(t) \).

The precise mapping function depends on a number of things, including the number of transmitters, the distribution of packet lengths, and the distribution of packet transmission times. In the case of V2V safety communication the sources are nearly periodic and the packet lengths are approximately constant. As shown in [23], under these conditions and if the number of vehicles \( K \) is at least 30, the mapping function is relatively insensitive to \( K \), i.e. the aggregate offered load is more important than the number of vehicles it is divided among. Fig. 5 shows obtained CBF field-test results versus aggregate message rate load. For example, a measured CBF of 0.58 corresponds approximately to 30 nodes sending 40 msg/sec, or 1200 aggregate msg/sec. In practice, the mapping function can be modeled as accurately as is desired. NS-2 simulation results presented in Section 4.3 illustrate the effect of implementing LIMERIC by using measured \( CBF(t) \) as an input, rather than measuring \( r(t) \) directly.

Even though global fairness for a larger network is out of the scope of this paper, here we provide a brief discussion of this issue. Some research is reported in [11] demonstrating global fairness in a larger network that uses a binary feedback-based message rate control. Global fairness means that nodes contributing to congestion at a given point within their respective interference ranges participate fairly in controlling that congestion, even if not all of the nodes are within 1-hop communication range of each other (i.e. some are hidden). In [11] global fairness is achieved via a multi-hop exchange of additional protocol information among the vehicles. We believe that LIMERIC can be extended to a larger network and it can achieve global fairness via a similar information exchange, for example each node’s measured CBF. We plan to study this aspect of LIMERIC in more detail as future work.

**4. NUMERICAL RESULTS**

In this section we present numerical results for LIMERIC for synchronous, sequential and probabilistic update scenarios. The synchronous and sequential update scenarios are deterministic and hence the equations are implemented in MATLAB. The probabilistic scenario has been simulated in network simulator (NS-2).

**4.1 Synchronous Case (MATLAB)**

In this subsection we present results of LIMERIC for the synchronous scenario (where all the nodes update their message rate at the same time). Parameters are set as follows: \( \alpha = 0.1, \beta = 1/150, X = 0.0005 \) and \( r_{\beta} = 0.6 \). The latter two correspond roughly to 1 msg/sec and 1200 msg/sec, respectively. We vary the number of nodes from 120 to 1000 in Figs. 6 and 7. For these \( \alpha \) and \( \beta \) values the algorithm should achieve perfect convergence when the number of nodes \( K \leq 285 \). With the gain saturation mechanism (Eq. (15)), the algorithm should converge to a limit cycle, rather than become unstable, when \( K \geq 285 \).

In Fig. 6, we plot the synchronous scenario results for LIMERIC for 120 and 210 nodes (only selected nodes are shown). It can be seen from the figure that nodes converge to a message rate governed by Eq. (7). In Fig. 7, we plot the results for 300 and 1000 nodes. These are extreme scenarios for which the network is very heavily congested. We observe from the plots that even for these extreme scenarios LIMERIC remains stable and enters into tolerable limit cycles.

**4.2 Sequential Update Case (MATLAB)**

In this subsection we plot the LIMERIC results for the sequential update scenario. The values of \( \alpha, \beta, X \) and \( r_{\beta} \) are set as in 4.1. We vary the number of nodes from 120 to 1000 in Figs. 8-9. It can be observed from the plots that from a lightly loaded scenario (nodes=120) to an extremely loaded scenario (nodes=1000), LIMERIC with sequential updates is in each case perfectly convergent.

**4.3 Probabilistic Case (NS-2)**

We simulated LIMERIC using the NS-2 simulator for more realistic scenarios. In this configuration of LIMERIC, a node does a message rate update when it generates a packet, so the interval between updates is variable and also varies from one node to another. The parameters used for DSRC simulations are as follows: transmit power = 20 dBm, Enhanced Distributed Channel Access (EDCA) parameters AIFSN = 6 and CW\_min = 7 (AC0 from IEEE 1609.4-2006), over the air message size = 378 bytes, data rate = 6 Mbps, channel bandwidth = 10 MHz, carrier sense threshold = -92 dBm and reception threshold = -92 dBm. The noise floor was set to be -99 dBm and the preamble capture has been enabled with a capture threshold set to 4 dB. Notice that a 6 Mbps channel has capacity for approximately 2000 378-byte messages. The nodes are placed such that there are few hidden nodes, using the exact topography and transmit attenuation values reported in [8], see Fig. 10. Details of the NS-2 implementation are discussed in [23]. As there are few hidden nodes, in this simulation we are using a minor simplification that one common CBF measurement is used among all the nodes. The target CBF is
set to 0.6. For LIMERIC the values of $\alpha$, $\beta$, and $X$, are again set as in 4.1. The maximum and minimum message rates for a node are set to 10 msg/sec and 0 msg/sec, respectively. We present CBF and message rate results for varying number of nodes from 90 to 300.

In Fig. 11, we plot the CBF for 180 nodes. It can be observed that CBF converges approximately to a range between 0.5 and 0.6. Further, in Fig. 12 the mean and 90% range CBF (represented by error bars) are presented to quantify the variation. It can be observed from the figure that the CBF obtained from NS-2 simulations is very close to theoretical CBF obtained by mapping $r_f$ computed by Eq. (14). In Figs. 13-16 we plot the message rate curves for $K = 120$, 180, 210, and 300, respectively. It can be observed from the plots that the message rates converge very close to the rate $r_f$ (scaled to msg/sec) predicted by the mathematical analysis in Eq. (7). This demonstrates that LIMERIC also converges in a fair and efficient manner for realistically noisy vehicular network scenarios simulated by NS-2. It is also interesting to see in Fig. 16 that for a highly loaded scenario of 300 nodes, for which inequality (6) is not satisfied, the nodes converge to message rate of approximately 4 msg/sec, albeit with a bit higher variation.

Fig. 6: LIMERIC for synchronous scenario with total nodes = 120, 210, inequality (6) satisfied

Fig. 7: LIMERIC for synchronous scenario with total nodes = 300, 1000, inequality (6) not satisfied

Fig. 8: LIMERIC for sequential update scenario with total nodes = 120, 210

Fig. 9: LIMERIC for sequential update scenario with total nodes = 300, 1000

Fig. 10: Physical WSU Layout in the Testbed, each WSU has two DSRC radios as in [23]
Fig 11: CBF plot for LIMERIC in NS-2 with total nodes = 180

Fig 12: Mean and 90% range CBF results using NS-2 simulator

Fig 13: Message rate plot for LIMERIC in NS-2 with total nodes = 120

Fig 14: Message rate plot for LIMERIC in NS-2 with total nodes = 180

Fig 15: Message rate plot for LIMERIC in NS-2 with total nodes = 210

Fig 16: Message rate plot for LIMERIC in NS-2 with total nodes = 300
5. CONCLUSIONS
This paper defines a new linear message rate congestion control algorithm, LIMERIC, which is designed for DSRC V2V safety communication. The linear control allows LIMERIC to converge perfectly in a noiseless environment, which provides a significant improvement over the limit cycle convergence behavior inherent in traditional binary control algorithms. It also avoids fairness problems observed in some binary controls. The convergence and stability properties are proved for the noiseless case. Gain saturation is introduced to the adaptation to provide robust stability independent of number of vehicles, and results are provided to illustrate the stabilizing effect. The paper also discusses practical considerations related to sequential updates and the determination of aggregate message rate via a channel busy measurement. Finally, the paper presents numerical results to illustrate the behavior of LIMERIC under a variety of challenging operating scenarios. LIMERIC is easily implementable, and can be an important tool in controlling channel congestion for DSRC V2V safety, among other environments.

6. REFERENCES