Minimizing Age of Information in Vehicular Networks

Sanjit Kaul and Marco Gruteser
WINLAB, Rutgers University, NJ, U.S.A
{sanjit,gruteser}@winlab.rutgers.edu

Vinuth Rai and John Kenney
TOYOTA InfoTechnology Center, CA, U.S.A
{vinuth,jkenney}@us.toyota-itc.com

***Abstract*—Emerging applications rely on wireless broadcast to disseminate time-critical information. For example, vehicular networks may exchange vehicle position and velocity information to enable safety applications. The number of nodes in one-hop communication range in such networks can be very large, leading to congestion and undesirable levels of packet collisions. Earlier work has examined such broadcasting protocols primarily from a MAC perspective and focused on selective aspects such as packet error rate. In this work, we propose a more comprehensive metric, the average system information age, which captures the requirement of such applications to maintain current state information from all other nearby nodes. We show that information age is minimized at an optimal operating point that lies between the extremes of maximum throughput and minimum delay. Also, via simulations we show that it cannot be achieved in 802.11 networks through pure MAC techniques such as contention window adaptation. This motivates our design of an application-layer broadcast rate adaptation algorithm. It uses local decisions at nodes in the network to adapt their messaging rate to keep the system age to a minimum. Our simulations and experiments with 300 ORBIT nodes show that the algorithm effectively adapts the messaging rates and minimizes the system age.**

I. Introduction

There is a new class of emerging applications that require nodes to periodically share their time critical status information with nearby nodes. Perhaps the most prominent examples are found in vehicular networks, where each vehicle shares its position and other vehicle dynamics with nearby vehicles to improve on-road safety. Though these applications use broadcast as their dissemination mechanism, their *QoS* requirements are far more stringent [1] than the typical use case for broadcast, traditionally, beaconing for announcing presence or non-critical information. For many new applications broadcasting is the dominant form of messaging. In vehicular networks, an entire band of 10Mhz is reserved solely for safety applications that broadcast their messages [2]. Likely high node densities make it important to have broadcast congestion control mechanisms [3] so that the applications can achieve desirable performance.

Vehicular network clusters may be spread over hundreds of meters and have hundreds of participating vehicles. A vehicle's state must be received by other vehicles at a sufficiently high rate so that, at any given time, it is recent enough for use by their on-board safety applications. The large numbers of vehicles possible in vehicular network clusters create the potential for congestion, even when only periodic status messages are sent. Most earlier work has looked at metrics that capture only selected aspects of the problem, like packet error rate and clear channel assessment (e.g., [4]–[6]). There is earlier work [7]–[9] that uses explicit feedback messages like the RTS/CTS exchange to detect congestion, and carry out PHY or source rate adaptation or TCP congestion control in wireless networks. In [10], a satellite broadcasts to terminals to reduce rate on detection of congestion. More recent work [11], perhaps the most directly related, provides control strategies in vehicular networks to maximize the average broadcast rate at which packets are received by a vehicle from its neighbors. The analysis assumes a saturated load (MAC always has a packet to transmit) and does not consider queuing delays. We are not aware of any work, however, that addresses the vehicular broadcast problem from a more comprehensive perspective, including queuing delays, delays from packet losses, and delays inherent in the selected application messaging rate.

In this paper, we use a system age metric to capture these issues. System information age Δ is the average end-to-end (application-to-application) delay observed in any vehicle's state within a certain cluster of nodes. We first show that to minimize information age the source rate of status messages needs to be adapted to an optimal operating point, which changes with node density. We then show through simulations that this operating point cannot be achieved through MAC layer contention window adaptation in an 802.11 MAC. This motivates a broadcast source rate adaptation algorithm that operates above the MAC layer. Given the broadcast nature of the messaging, senders do not immediately receive feedback on sent messages. Instead they infer feedback from the status messages received from other nodes. Our proposed algorithm relies entirely on local calculation of the system age metric and node broadcast periods, based on these received packets.

In summary, our specific contributions include:

- Quantitatively showing that minimizing the system age cannot be achieved by maximizing throughput in a practical 802.11 system and that the minimum age at any given contention window (CW) size is achieved at an application messaging rate such that the offered load is much smaller than when saturated.
- Arguing that in systems with tail-drop FIFO queues rate control for broadcasts needs to be provided above the MAC layer.
- Via simulation we confirm the presence of a unique common period at which system age is minimized and propose a distributed algorithm that relies on local infor-

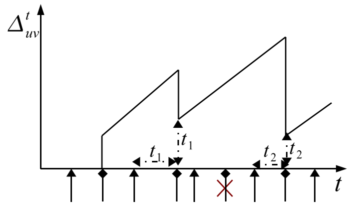

Fig. 1: Vehicle u *generates* its state at instants shown by arrows with triangular heads. Vehicle v *receives* a generated state packet at the next instant marked by diamond shaped heads. An erroneous reception is marked by a cross. The age of u's information accumulates with time until it is reset to the time elapsed between state generation and reception (t_1 and t_2 in figure). The average age Δ_{uv} is given by the area under the curve normalized by the interval of observation.

mation alone to achieve the minimum. Simulations and also experiments with 300 ORBIT testbed nodes show the efficacy of the algorithm.

The paper is organized as follows. In Sections II-B and III we show how the system age may be minimized and motivate rate control. The rate control algorithm is described and evaluated in Section IV. In Section V we discuss current limitations and how the proposed technique can complement others. In Section VI we describe the related works and conclude in Section VII. Next we define our system model.

II. System Model and Objective

We consider a network of vehicles that communicate with each other using an 802.11 based CSMA mechanism for channel access. Also, we assume that the clocks of all vehicles in the network are synchronized using, for example, GPS devices. Every vehicle executes an application that generates packets at a periodicity of T sec for *broadcast*. The packet contains the vehicle's state information, for example, location and velocity, and the time it was generated and also the period T (used by rate control). Let Δ^t_{uv} be the age of the state information of vehicle u at vehicle v at time t. A vehicle always has its current information. Thus, $\Delta^t_{uu} = 0$. At vehicles $\{v \neq u\}$, the state information of u ages during the time between any two successfully received broadcasts from u. Also, the age of any received information is the time that elapses between its generation and successful reception (see Figure 1). Over the time interval of interest $\mathcal{T}$, let Δ_{uv} be the average age of u's information at v. Thus $\Delta_{uv} = (1/|\mathcal{T}|)\int_{\mathcal{T}} \Delta^t_{uv} dt$. Let V denote the network of vehicles and let $N = |V|$. We define *system age* Δ as

$$\Delta = \frac{1}{\mathcal{N}} \sum_{u \in V} \sum_{v \neq u} \Delta_{uv} \tag{1}$$

where $\mathcal{N} = N(N-1)$. The system objective is to minimize the *system age* Δ[1].

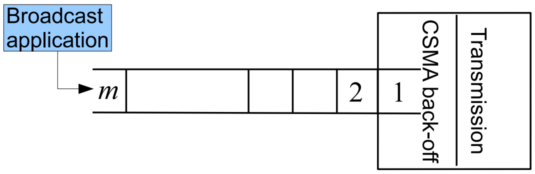

Fig. 2: Application at a node generates packets that are queued into a queue of size m. The packet at the head of the queue, numbered 1, awaits 0 or more backoff slots before its transmission starts.

A. Delays in an 802.11 MAC system

Here we summarize an 802.11 system and the delays experienced by packets in it. A more thorough summary can be found in [12]. Packets generated by the state broadcast application at a node arrive at its 802.11 link layer, where they are queued for transmission if the queue is not full, else they are dropped. Figure 2 shows the 802.11 link layer consisting of a FIFO queue of length m. Packets are transmitted over the wireless medium using the CSMA backoff mechanism, where in on receipt of a packet, if the channel is busy, the CSMA MAC at the node selects the number of slots to backoff (backoff counter) from a discrete uniform distribution over the interval $[0, \text{W0}-1]$, where W0 is the *contention window* (CW) size. The broadcast nature of the packets implies that the receivers do not explicitly acknowledge having received a packet transmission. This precludes the use of an exponential backoff mechanism and the contention window size stays fixed at the chosen W0 for broadcast applications.

Let the length of an idle slot be σs (equal to a PHY layer slot, 13μs in the 802.11p standard). Since counter decrements take place only during idle slots, the average time a packet spends waiting is a function of the average slot length $\overline{T}$, that is the average time that the MAC stays in a given state, described by the backoff stage and the value of the counter. Let a packet transmission/collision occupy an *average* of L slots (includes data payload, header and other delays like that of DIFS, SIFS and ACK (on unicast)). Thus $\overline{T} = p_I\sigma + (1-p_I)L\sigma$, where p_I is the probability that the medium is idle (no transmissions).

Let the CSMA MAC have a packet ready for transmission (head of queue) with probability q. A larger q implies a higher message rate, smaller broadcast period T. When $q = 1$, we say that the MAC is in saturation.

A broadcast packet at the head of the queue will wait an *average* of $(\text{W0}-1)/2$ slots before being transmitted, if the channel is busy, else it will begin transmission (we included DIFS in the average transmission time).

Let T_s be the average service time for a packet generated by the application, that is the average time elapsed between a packet arrival into the queue and the end of its transmission. We have, $T_s = T_w + T_b + T_x$, where T_w is the average time a packet spends in the queue before arriving at the head, T_b is the average time spent by the packet at the head waiting for backoff to end, and T_x is the sum of the propagation delay (negligible) and the transmission delay (size of the packet divided by the PHY layer rate (bps)). Further, $T_b = (1-p_I)\overline{T}(\text{W0}-1)/2$, where we assumed that the counter

[1] In the network simulations later, we average over discrete time instants. We calculate $\Delta_{uv} = 1/n\sum_{i=1}^{n}(t_i - T_u(t_i))$, where $T_u(t_i)$ is at time t_i, the time at which the state information of u last received by v was generated. The difference $t_i - T_u(t_i)$ is the instantaneous age of u's information at v at t_i. The number of time samples n is chosen large enough to sample age at a large enough rate. The system age is then obtained as in Equation (1).

is selected independently of the slot size.

B. Information Age and Throughput

To put the concept of information age into perspective, let us consider how it relates to throughput and queuing delay for a simplified network containing a queue into which the application sends packets. The service time of a packet is T_s, after which it is received. In our network, throughput can be increased by reducing the broadcast period T.

Does increasing throughput minimize information age? Consider first a lightly loaded network, where the service time is well below the broadcast period, i.e., $T_s << T$, and thus the queue is always empty when a new packet is generated (the waiting time $T_w = 0$). In this case the average information age is $\Delta = T_s + T/2$, where T_s is negligible. Thus, it is clear that reducing T will lead to a lower information age.

The service time in general, however, is also dependent on T, since the network load affects queuing delays. This will become particularly noticeable when the network becomes more heavily loaded (smaller T). In fact, classic networking theory based on M/M/1 models with infinite queues tells us that queuing delays rise sharply as throughput approaches its maximum [13]. Beyond a certain optimal operating point, these delays will substantially increase the service time T_s and thereby increase the information age. Conversely, delay is minimized as the throughput approaches zero. However, this would make T, and hence the system age, very large. Thus, information age is minimized neither by solely maximizing throughput nor by solely minimizing packet delay. Instead information age reaches its minimum at an optimal operating point between these two extremes, akin to how the power of the network can be maximized in classic network theory.

Our system differs in a number of subtleties from this classic model. The interaction of MAC contention and queuing policies with finite queues with information age is more intricate. We will study this next through simulations.

III. Minimizing Information Age in a CSMA System

Is there an effective MAC layer solution for minimizing information age in a CSMA system? Given our insight that *information age is minimized at an optimal operating point where throughput is high and queuing delays are low*, we now consider several MAC layer strategies in a CSMA system for reaching this operating point. In particular, we will investigate (i) whether the 802.11 contention mechanisms could provide congestion control necessary for minimizing information age and (ii) whether minimizing the queue size can *effectively* avoid the queuing delays, so that the problem of minimizing information age is reduced to maximizing throughput.

We investigate contention mechanisms because in a CSMA multiuser access system, the information age problem is further complicated by the effect of load on packet loss rates. Higher load also results in increased collisions during transmissions, which would lead to a higher information age. In fact, even throughput cannot be maximized by merely increasing source rate. Instead CSMA uses a contention window mechanism that seeks to achieve the optimal operating point. We will investigate whether similar mechanisms also minimize information age.

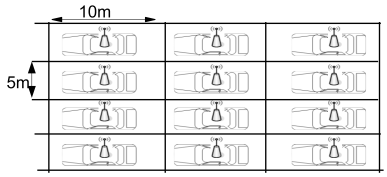

Fig. 3: A four lane road network with cars placed very close to each other to simulate a high density environment.

With respect to queuing, it is clear that smaller transmission queue sizes reduce queuing delay and therefore improve information age at higher load. In the class of status information sharing applications that this paper considers, there is actually little value in buffering outdated information for transmission. Can the queuing delays be avoided simply by configuring the MAC with small buffer/queue sizes? Or, would it be worth redesigning the hardware to eliminate queues for such status messages altogether.

Since the affect of these issues on information age is quite intricate, we resort to simulations to shed light on these questions.

A. Simulation Setup

We use the Network Simulator (version 2.33)[2], with PHY layer extensions by Chen et al. [14]. The extensions provide an implementation of the 802.11a/p physical layer and handle capture and packet decoding criteria as in the standards. For our simulations we use the 802.11p standard, which has been proposed to support messaging between vehicles [2]. All simulations use a four lane network illustrated in Figure 3. The placement of vehicles is chosen to simulate as high as possible wireless medium congestion levels for the two dimensional road network. The physical grid shape ensures that effects like capture seen in simulations are similar to those expected in real road networks. Mobility is not considered and we assume that over the few seconds that the rate control takes to converge (see Section IV-A) the wireless connectivity of the vehicles under consideration is more or less the same. The MAC in the simulator does not transmit any periodic management messages. Table I lists default values for various parameters used.

The application generates broadcast packets/messages at the configured periodicity plus a small random offset to ensure that packet generation at the different nodes is not synchronized. The packet is sent to the link layer of the ns-2 simulation stack. The minimum queue size supported is 1 packet. The queue will hold at least one packet when another one is awaiting completion of backoff and transmission (Packet 1 in Figure 2). In general, a queue size of k in ns-2 corresponds to $m = k+1$ in Figure 2. Last but not the least, all cars/nodes/vehicles are in communication range of each other.

[2] http://www.isi.edu/nsnam/ns/

Pathloss Exponent	2.0
Frame capture thresholds	10dB
Transmit Power	1W
PHY Rate	6Mbps
Number of cars	$\leq$ 400
Receive Sensitivity	-99dBm
Application Payload Size	300 bytes

TABLE I: Simulation Parameters

To address the contention mechanism question, we consider different contention window sizes in our simulation. We know from Bianchi [12] that the *system throughput* of a saturated 802.11 based network (nodes always have a packet ready for transmission) can be maximized when the contention window size W0 is set to its optimal value W*, which depends on the number of nodes in the network[3]. We will therefore conduct simulations with standard contention window sizes and with W* for our network.

At a PHY rate of 6Mbps, an application payload size of 300bytes and overheads like packet headers and DIFS (=2*σ + SIFS) [16] we get W* $= 3772$ slots for $N = 400$ nodes.[4] Such an inordinately large contention window setting is not possible in current 802.11 implementations. We consider it here to understand whether it would be worth enabling in future implementations.

To address the queuing question, we will introduce a hypothetical system, which we will call the *Latest* state *Out* system, that eliminates queuing delays. Whenever a packet transmission opportunity arises, it fills the packet with the latest available state information. In other words, the age of information in the packet, if received successfully, is equal to its transmission time T_x. We compare this system with standard FIFO tail-drop queuing system as used in 802.11 implementation we are aware of. We refer to this system in short as **FIFO**. Here, the state information in a packet is not updated after it is generated by the application and queued. We will investigate, however, whether the affect of queuing delays on system age can be virtually eliminated by using small queues.

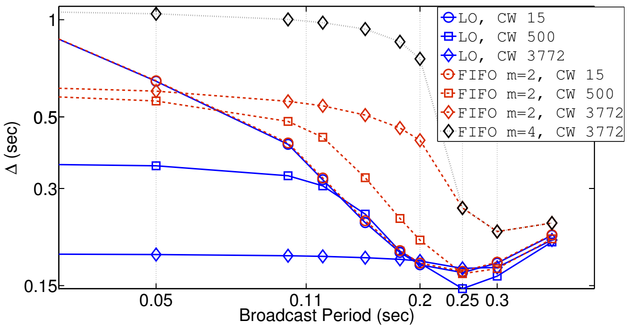

Fig. 4: System age as a function of broadcast periodicity for **LO** and **FIFO** assuming a queue size of $m = 2$. Selected W0 $=$ $15, 500, \mathrm{W}^* = 3772$. For W0 $= 3772$ we also show age for $m = 4$. The circles markers show different cases for CW 15, the squares CW 500 and the diamonds show CW 3772.

B. Results

In practice (using FIFO) smaller periods can lead to significantly larger age at large CW: Figure 4 shows Δ as a function of the broadcast period T set at each car, in the network of 400 cars. The simulations were done assuming **LO**[5] and **FIFO** for a queue size of $m = 2$ (ns2 LL queue size of 1) and $m = 4$. We see that in comparison to **LO**, for large CW sizes of 500 and 3772, a significantly greater system age is observed even for the smallest size **FIFO** ($m = 2$), especially for very small periods (that is under saturation). However, under saturation, using **LO** and W0 $=$ W* keeps Δ very close to the minimum achieved (≈ 0.15s) by any of the plotted configurations[6]. However, the same is clearly not true when **FIFO** ($m = 2$) is used, where ages greater than 0.5s are observed for very small T. Increasing the queue size by 2 to $m = 4$ increases the age for saturated loads by about twice. In fact, *for no chosen T and CW size does **FIFO** queuing achieve a smaller system age.*

Also, for small T, the difference in achieved Δ for **LO** and **FIFO** ($m = 2$), is larger for W0 $= 3772$ than for W0 $= 500$. The reasons for it are two-fold and insightful. Using the optimal CW size of 3772 maximizes the throughput (inter-arrival rate) at a receiver and so given the **LO** system, the age achieved when W0 $= 3772$ is less than when using W0 $= 500$. However, under **FIFO**, a larger CW size means a received packet will wait for longer, which, unless the smaller CW size (for example, see W0 $= 15$) leads to much larger collision errors, will lead to a larger Δ. In fact, at W0 $= 15$ the queuing and backoff delays are very small, for very small T too, leading to a negligible difference between **LO** and **FIFO** (see figure).

Saturating the load under **LO**, with W0 $=$ W*, is intuitively equivalent to delivering the latest state information as fast as possible. *For **LO**, unlike **FIFO**, maximizing the throughput is equivalent to minimizing the system age.*

Other than the requirement of updating the state at the very last instant and supporting a W* that increases with N in hardware, **LO** is also not practical as it requires a broadcast application to transmit at very high rates, which could overwhelm the resources available at a radio node that is expected to handle not only multiple kinds of safety applications, but also other applications, like infotainment, with different QoS requirements.

However, note that the system age achieved by **LO** can be

[3] Several algorithms have been proposed to adapt W0 based on node density, for example [15].

[4] For a single stage back-off system that the backoff application uses, the optimal value CW size under *saturated load* conditions is W* $=$ $N\sqrt{2T_c/\sigma}$ [12], where N is the number of nodes in the network, T_c is the *average* packet collision time and σ is the PHY layer slot length, which is $13\mu s$ for 802.11p.

[5] Queue size does not matter for LO as state is updated at the last moment.

[6] The age is 0.15 and not close to T_x, the aging of a received packet under **LO**, because of packets received in error due to collisions

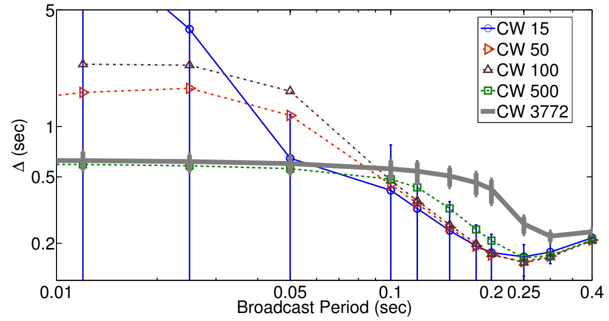

Fig. 5: Variation of Δ with period for different CW sizes.

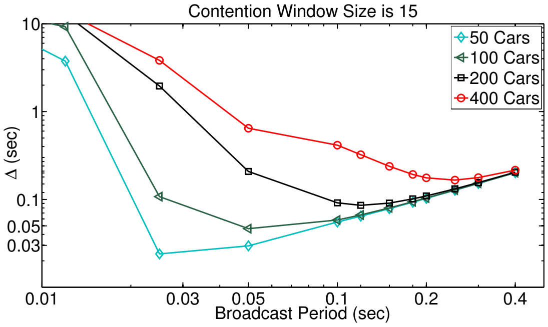

Fig. 6: Variation of Δ for different sized car networks.

obtained by **FIFO** too, albeit at smaller CW sizes and large values of the period T for the small queue size of $m = 2$. In Figure 4, CW sizes of $15, 500$ achieve the minimum age at a period of 0.25s, for $m = 2$. At these CW sizes the queuing delays at large T, for $m = 2$, are negligible.

The infeasibility of implementing ***LO*** *motivates designing a rate control mechanism, which for* ***FIFO*** *with a small queue size can find the T that achieves the optimal (minimum) system age for a network.* For rest of the paper we assume a **FIFO** queue of smallest size ($m = 2$) (greater m, as we showed for W0 $= 3772$, lead to larger Δ for other CW sizes too) and look at how the achieved system age can be minimized.

A contention window based strategy for reducing age is infeasible: The typical use case for an 802.11 system has been unicast applications, where a given packet generated by an application needs to be delivered. In dense networks, where there are a large number of users contending for the channel to transmit their packet, packet delivery rates can be improved by increasing the CW size, as it reduces the probability of collisions. In our application a given packet is not so important, because holding it for too long will only make it stale. Figure 5 shows why a strategy that changes the CW size in response to a congested channel is unsuitable to make Δ smaller. At small T (close to saturation), increasing CW size from 15 to 500 or 15 to 50 reduces the age. However, increasing the CW from 50 to 100 makes it worse. CW sizes of 500 and 50 alleviate the collision probability significantly, such that they do better than 15 at the small T, but for their larger queuing delays. On the other hand W0 $= 100$ does not reduce the collisions enough and adds to the queuing delays at W0 $= 50$. Finally, contrary to behavior at small T, at large T, larger CW sizes can make delay worse because of larger queue wait times. Note that the minimum age achievable at W0 $= 3772$ is 0.22s instead of the 0.15s for smaller CW sizes, where the minimums are at T far from saturation, at $T = 0.3$s and $T = 0.25$s respectively.

Age of information seen by all nodes is close to the optimal: In Figure 5, for W0 $= 15$ and W0 $= 3772$, for each T we plot a bar the end points of which denote the minimum and the maximum Δ_{uv} observed at the T. As T reduces, the length of the bar increases significantly for W0 $= 15$. This is a result of physical layer capture, which allows larger rates of packet delivery between close by nodes than those that are farther apart. For the larger CW size of 3772, collisions are less probable and the benefits of capture are very limited. However, the unfairness is negligible even for the small CW size at and around the period at which the system delay is minimized (0.25s for W0 $= 15$ and 0.3 for W0 $= 3772$). The observation holds true for similar sized networks and implies that when operating at the period corresponding to the minimum age, links between all node pairs u and v can achieve an age close to minimum Δ.

In what follows all our evaluation will be for a CW size[7] of 15 and a queue size of $m = 2$.

The existence of a unique period that minimizes system age, together with the infeasibility of LO and using changes in CW motivates *application layer rate control*: In Figure 6 we plot for networks of size $50, 100, 200, 400$, W0 $= 15$, and $m = 2$, the system age Δ with the message period T set the same at all nodes. As is seen in the figure and also Figures 4 and 5 (different queue and CW sizes), Δ is minimized at a period, typically, to the right of where saturation occurs for larger CW sizes, and to the left of periods that leave the wireless medium very lightly loaded[8]. The existence of a minimum motivates designing of a rate control algorithm that can achieve it in a distributed manner. Intuitively, a minimum must exist at a period larger or equal to the one that saturates the load offered to the CSMA MAC.[9]

IV. Rate Control Algorithm

We require an algorithm that runs on a node and adapts its broadcast period such that system age Δ is kept minimum for the network. Assume that a node v's broadcast contains its state information, the time the information was generated and the periodicity T_v set at the node's application.

Lack of global information: The exact calculation of system age, given in Equation (1), requires knowledge of Δ_{ij} for all vehicles pairs in the network. A given vehicle v can, however, only estimate a subset of the terms Δ_{ij}, where $j = v$, based on the broadcasts it receives from the other vehicles. Let $\Delta_v = \sum_{i \neq v} \Delta_{iv}$ and let $\hat{\Delta}_v$ be the estimate of Δ_v calculated

[7] The value of W0 $= 15$ is proposed for use in the 802.11p MAC and is also the default for broadcast applications in WiFi standards like 802.11a/b/g.

[8] In the Figure 6 the minimum Δ(s) are $\approx$ $0.02, 0.05, 0.1$ and 0.15 seconds for $50, 100, 200$ and 400 node networks respectively, achieved at T of $0.03, 0.05, 0.1$ and 0.25 seconds respectively.

[9] A period smaller than the largest period (fastest rate) that saturates the MAC, will not add to the load offered to the MAC queue and thus will not change the system age (See flat regions (constant age) in Figure 4).

Algorithm 1 Update Broadcast Period at Node v.

```
Require: $T_v$, $\widehat{T_{\mathcal{R}}}$, $\hat{\Delta}_v$, $\hat{\Delta}_v^-$, $\Lambda$, $\beta$, $\delta_s$.
Ensure: Updated value of $T_v$, the period at node $v$.
if $|\widehat{T_{\mathcal{R}}} - T_v| > \delta_s$ then
    $T_v = \widehat{T_{\mathcal{R}}}$
    $\Lambda$ = INCR{/*Forcing nodes to believe they took the same action. Setting to DECR is fine too.*/}
else
    if $\hat{\Delta}_v > 2\widehat{T_{\mathcal{R}}}$ then
        $\Lambda$ = INCR{/*Implicitly assuming congestion*/}
    else if $\hat{\Delta}_v > \hat{\Delta}_v^-$ then
        $\Lambda = \Lambda^c$ {/*Reverse action. Take complement*/}
    end if
end if
if ($\Lambda$ == INCR) then
    $T_v = \beta T_v$.
else
    $T_v = T_v/\beta$.
end if
return $T_v$
```

by v at the end of a *Measurement Interval* $\mathcal{I}$. Let $\mathcal{R}$ be the set of nodes v receives packets from during $\mathcal{I}$. The node uses the broadcasts it receives during $\mathcal{I}$ to calculate $\hat{\Delta}_v = \sum_{i\in\mathcal{R}} \hat{\Delta}_{iv}$, where $\hat{\Delta}_{iv}$ is the average age of i's information at v calculated over $\mathcal{I}$, as described in Section II.

All information and estimates calculated are discarded at the end of an interval. Node v also calculates the average $\widehat{T_{\mathcal{R}}}$ of the periods of all nodes in $\mathcal{R}$. The period corresponding to a node u is set to the period T_u in the packet last received from u. We have, $\widehat{T_{\mathcal{R}}} = (1/|\mathcal{R}|)\sum_{u\in\mathcal{R}} T_u$. The measurement intervals need not be synchronized across the nodes in the network. We now describe the algorithm's actions.

Keeping the spread of chosen T_v at nodes limited: The spread at a node v is given by $|\widehat{T_{\mathcal{R}}} - T_v|$. If the spread was greater than δ_s, the node's period is set to the estimated average over periods of other nodes, $\widehat{T_{\mathcal{R}}}$. If the spreads[10] are not limited, the system age was seen (in simulations) to converge to a value other than its minimum. Note that the existence of a ***unique*** minimum system age was assuming a common broadcast period for all nodes in the network. It is not clear that the same will hold true when nodes are allowed to choose from a broad range of periods. Choosing a very small δ_s, however, can be detrimental and lead to the algorithm at a node setting $T_v = \widehat{T_{\mathcal{R}}}$ at the end of every $\mathcal{I}$, and descent towards the minimum age may not occur.

Increase period blindly, if medium congested: If the algorithm determines that $\hat{\Delta}_v > 2\widehat{T_v}$, it assumes that the medium is congested[11] and increases the period irrespective of the result of Λ^-. The motivation behind the step is that if the age is greater than twice the period then the age must be achievable at a much larger period. Remember that if the medium is not congested, say if $T_s \ll T$ (see Section II-B), then the average achievable age is $\approx T/2$.

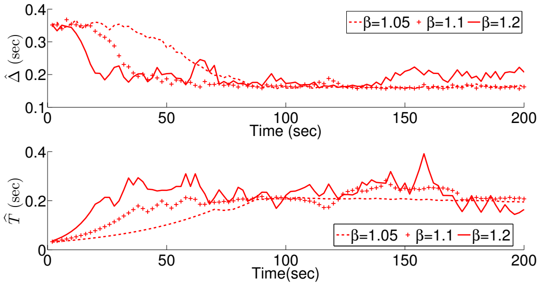

Fig. 7: Effect of selection of β on rate control. 400 nodes, start period of 0.03s and $|\mathcal{I}| = 2$s.

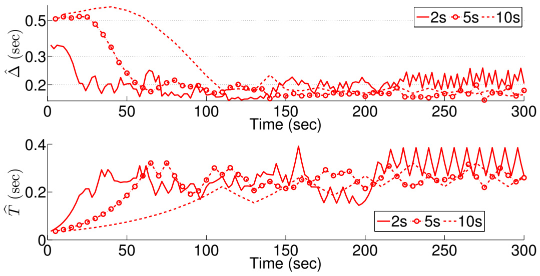

Fig. 8: Effect of selection of $\mathcal{I}$ on rate control. Start period of 0.01s, 400 nodes, $\beta = 1.2$.

Approaching the minimum, choosing the descent direction: A node invokes Algorithm 1 at the end of its current $\mathcal{I}$. Let $\hat{\Delta}_v^-$ be the estimate of Δ_v that was calculated in the previous $\mathcal{I}$. The algorithm needs to decide whether the current period T_v must be increased (action Λ = INCR) or decreased (Λ = DECR), given $\hat{\Delta}_v^-$, $\hat{\Delta}_v$ and Λ^-, where Λ^- is the action it had taken at the end of the previous $\mathcal{I}$. The choice the algorithm makes over successive $\mathcal{I}$ must take the system age Δ closer to its optimum (minimum) value.

The algorithm, however, has no knowledge of whether, for the given network, the minimum is approached by increasing or decreasing the current T_v. Based on its local information it repeats the action at the end of the previous $\mathcal{I}$ if $\hat{\Delta}_v < \hat{\Delta}_v^-$, else it performs the opposite, that is increase the period if it had earlier been decreased and vice-versa. The factor $\beta > 1$ by which the period must be increased or decreased is an input parameter and, as we will show later, its selection is a trade-off between speed and accuracy of convergence.

A. Algorithm Evaluation

We now show the effect of the parameters β and length of $\mathcal{I}$ on the algorithm's performance, followed by an evaluation of the algorithm using a large number of simulations, and experiments on the ORBIT grid. The average system age $\hat{\Delta} = (1/N)\sum_{v=1}^{N} \hat{\Delta}_v$ and the average period selected by nodes at the end of an interval $\widehat{T} = (1/N)\sum_{v=1}^{N} T_v$ are calculated offline at the end of every interval for evaluation purposes.

Effect of β: In Figure 7 the length of $\mathcal{I}$ is set to 2s and 400 nodes start broadcasting with a period of 0.03s (medium

[10] We used a δ_s of $1/2(\widehat{T_{\mathcal{R}}})$ for $\widehat{T_{\mathcal{R}}} < 0.1$ and 0.05 for larger $\widehat{T_{\mathcal{R}}}$ in experiments and simulations.

[11] Ideally, this should be supported by an indication from another mechanism (e.g., the PHY), so that very large packet error rates are not confused with congestion.

is very congested). For $\beta = 1.2$, the algorithm achieves system age $\hat{\Delta} < 0.2$s (top sub-plot) within 20s (minimum achievable, see Figure 6, is 0.17s). However, $\hat{\Delta}$ sees large fluctuations as time progresses. The bottom sub-plot shows $\widehat{T}$, and is also more jagged as a result of different nodes v selecting more varied periods than when β is smaller. For the setting of $\beta = 1.1$, $\hat{\Delta}_v$ crosses 0.2s at about 40s and stays smooth and below for most of the 200s of the simulation[12]. The value of $\beta = 1.05$ takes much longer than 40s but is not much better than $\beta = 1.1$ in following the minimum age. We find that in general $\beta = 1.1$ has desirable convergence performance.

Effect of length of $\mathcal{I}$: In Figure 8, we look at interval lengths $|\mathcal{I}|$ of $2, 5, 10$sec for $\beta = 1.2$. The 400 nodes start broadcasting with an initial period of 0.01s. From the plot we see that the initial estimated Δ for $\mathcal{I} = 2$s, is ≈ 0.3 instead of the 0.5 obtained for $\mathcal{I} = 5, 10$. The difference in averages is because for the smaller interval, a node v does not receive sufficient packets from other nodes to accurately estimate $\hat{\Delta}_v$[13]. The inaccurate estimates can lead to nodes choosing different Λ_v, leading to different selected periods T_v and less smooth convergence. Note the jagged nature of the curves corresponding to $|\mathcal{I}| = 2$s. A smaller interval will get the network closer to the optimal age faster, however. In the figure, for the interval of 2s, $\hat{\Delta}$ first goes below 0.2s at about 20s into the experiment. It takes > 100s for the interval of 10s.

Note that forcing a node v to increase its period (see Algorithm 1) whenever $\hat{\Delta}_v > 2\widehat{T_{\mathcal{R}}}$ (the smooth descent in $\hat{\Delta}$ for time < 30s and $|\mathcal{I}|$ of 2s in plot), makes the algorithm more robust to such averaging related errors and takes the system closer to the minimum age quickly. This is because as long as the nodes satisfy the condition the exact value of $\hat{\Delta}_v$ at v is unimportant.

Quality of convergence: We ran the rate control algorithm with start periods selected randomly over the range of $(0.03, 0.5)$ for a network of 400 cars. A total of 124 simulations were run, out of which about 100 had a common initial period set at each vehicle, while for the remaining each vehicle randomly and independently selected a start period, at the beginning of simulation, from the interval above. For $\beta = 1.1$ and $|\mathcal{I}| = 2$s, the median time it took for $\hat{\Delta}$ to go below 0.2sec was 4s, the mean was about 8s. From the instant to the end of simulation (a total of 200s long) 95% of the nodes observed $0.16 \leq \hat{\Delta}_v \leq 0.18$, which is very close to the minimum achievable (0.17s) as seen in Figure 6. Similar simulations for 200 node networks also saw good convergence.

B. Evaluation on the ORBIT testbed

The ORBIT grid [17] is a 400 node, 800 radio grid that hosts Atheros and Intel radios and allows for emulation of real world wireless network experiments. The 400 nodes, as shown in Figure 10, occupy a 20m x 20m area and hence provide an excellent platform for testing network algorithms for high density networks.

Experiment setup on ORBIT: We evaluate our algorithm using 298 nodes, all containing the Atheros chipset AR5212[14]. The chipset supports the 802.11a standard, which though not the same as 802.11p, is very similar to it and is suitable for evaluating our rate control algorithm as we are only interested in MAC broadcasting and CSMA aspects of 802.11a. We use the Atheros Linux Wireless driver ath5k[15], which we modify to disable beaconing, reduce the buffer space (queue size) to the minimum possible, increase the maximum number of allowed STAs to 512 and set the CW size to a fixed value of 15. The default transmit power of the cards is about 20dBm and PHY layer rate is fixed at 6Mbps (default for broadcast), and so all the nodes are in communication range of each other. We set the frequency to 5.22Ghz. The grid is an environment of much greater node density than the road network we used for simulations and may lead to different capture and collision characteristics than those we observe in simulation. However, the change in Δ with T was very similar in character.

The state application is run as a UDP broadcast application and, for the results shown in Figure 9, uses $\beta = 1.1$ and $|I| = 5$s. We used settings of $\beta = 1.2$, $|I| = 2, 10$s too. While $|I| = 2$s seemed too short an interval for getting an average delay estimate on the grid (probably because of the high density of nodes), convergence was observed for the other settings. Last but not the least, the Network Time Protocol is used to synchronize the clocks at the nodes.

Evaluation of results: The dotted staircase (top of Figure 9) denotes the number of nodes (plotted on the axis to the right) that were a part of the network at a given time. At the beginning, $t = 0$, we have 298 nodes, 63 of which (about 3 rows on the grid) are switched off at $t = 250$s. The nodes in the network are reduced every 300s. At $t = 1520$s only 50 nodes remain in the network. At $t = 0$ all nodes have their broadcast periodicities set to 0.05s.

The start period of 0.05s and 300 nodes leads to a large system age of ≈ 0.5s at the beginning of the experiment (see Figure 9). The nodes, however, soon start descending to a smaller system age of ≈ 0.13s. After every instant at which cars leave the network, the remaining nodes converge to a smaller $\hat{\Delta}$, at around which they stay till the next change in the network. The spikes in delay that coincide with cars leaving the network are observed in the interval $\mathcal{I}$ during which the cars leave. This is because, the remaining cars expect the cars to be a part of the network, but do not receive any packets from them. In the interval that follows lesser cars remains and can achieve a smaller age, towards which the system age begins to descend. At the end of the experiment the 50 car network achieves an age of 0.02sec (close to optimal) in contrast to the 0.13s the 298 nodes at the beginning had converged to.

[12] Note that in 40s, $\widehat{T}$ goes from 0.03 to 0.2, which is an order of magnitude larger. Convergence time for larger start periods, shown later, is much smaller.

[13] In general $\mathcal{I}$ may need increasing to obtain *accurate* estimates in larger networks for a given period T, or for smaller T in a given network.

[14] www.atheros.com

[15] http://linuxwireless.org/en/users/Drivers/ath5k

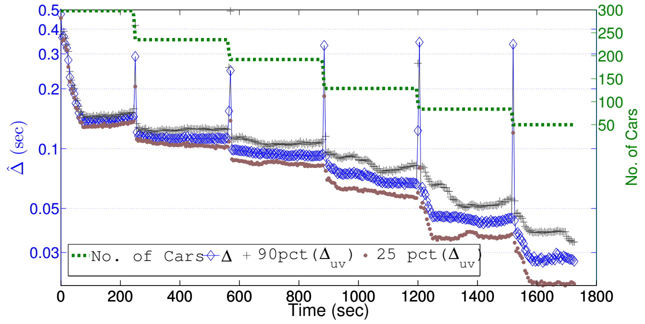

Fig. 9: Experiment on ORBIT. $\mathcal{I} = 5$ and $\beta = 1.1$. The max difference between the shown percentiles is ≈ 0.02. The left y-axis is in log-scale. The dashed staircase at the top shows number of nodes in network. The three other curves starting from top are the 90th percentile of $\{\hat{\Delta}_v\}$, $\hat{\Delta}$ and the 25th percentile of $\{\hat{\Delta}_v\}$ percentile.

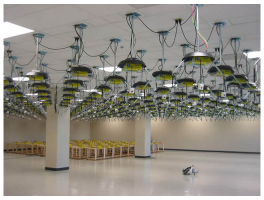

Fig. 10: The 400 node ORBIT grid. Nodes hang from the ceiling.

V. Limitations and a Few Observations

The system age metric assumes that all vehicles interested in each others' state know of each others presence. Even though different nodes may see different average information age from a given vehicle, the average system age estimated by them must be *similar*. This will be true if the nodes are in communication range of each other. However, this may not hold true in the presence of interference due to hidden nodes.

As a result, groups of nodes in the network may not decode messages from all nodes of interest, assume that the network is smaller than it is, estimate a smaller system age and converge to a smaller broadcast period. To avoid the same may require the nodes to add to their messages, less frequently than their state information, their neighborhood information, encoding the approximate region over which they hear nodes and the number of nodes they hear from the region. This way, even if nodes cannot decode from all nodes they want to listen to, they can know of the nodes' presence from other nodes' messages and account for them in their estimate of the system age.

On Power Control: The rate control algorithm will ideally operate in conjunction with a power control algorithm which sets the communication range of nodes. In the absence of power control and a very large number of cars, the rate control algorithm will push the broadcast period at the nodes to close to a very large value, a highly undesirable scenario. Once the size of network is chosen, the rate control algorithm can then minimize the system age seen by nodes in the network.

Other congestion control strategies: TCP congestion control uses multiplicative decrease when a packet delivery fails and 802.11 multiplies its CW size by 2. The strategies are agnostic to network size and application requirements, however, and simply respond to packet errors. Packet error rate (PER) at a given broadcast period T is a function of number of nodes N in the network and any PER thresholds chosen to vary T would need to be a function of N. Also, one would need to know a priori the PER corresponding to minimum system age for all possible N. Same holds true vis-a-vis packet delay based thresholds.

VI. Related Work

Controlling congestion in the DSRC channel used to convey vehicle-to-vehicle safety messages has been the subject of several papers in recent years [4]–[6], [18]. The authors of [4] use transmit power as the primary control. The power is computed locally at each sender in an approximation of a global max-min fair allocation, based on the location of the other vehicles within the sender's range. Transmit power is also the principal means of controlling congestion in [5]. In that paper the control is a linear function, within minimum and maximum power constraints, of the channel load, as measured by the Clear Channel Assessment (CCA) function defined in [19]. Message transmission opportunities are also chosen advisedly, using a concept that the same authors explore in more detail in [18]. In that paper the transmit events are chosen statistically as a function of CCA, perceived packet error ratio (PER), and vehicle dynamics, in an attempt to control a receiver's modeling error and to use the channel efficiently. In [6] a vehicle transmits safety messages at a rate that is adapted as a function of the local CCA and the CCA measured by neighboring vehicles. In [20] beaconing rate is adapted based on estimated channel capacity and message priority. The goal is delay sensitive distribution of traffic information to optimize routing of vehicles. In [21] the authors look at the effect of beaconing rates on channel load and the accuracy of the estimated position at beacon recipients. However, none of the above works use an average end-to-end application delay based metric.

In [11] the authors provide broadcast congestion control strategies in vehicular networks to maximize the average rate at which packets are received by a vehicle from its neighbors, assuming a Rayleigh faded channel. However, the analysis does not include queuing delays.

In [22] the authors propose congestion control by prioritizing messages based on their deadline, in vehicular networks. In [23], the authors propose prediction and efficient messaging to keep the number of transmitted messages in a vehicular network to a minimum.

A protocol that improves broadcast reliability in 802.11 networks is proposed in [24]. The protocol uses the collision avoidance of 802.11 and RTS/CTS and NACK frames for reliable delivery of broadcasts.

Rate and congestion control has been studied extensively in ad-hoc and sensor networks. In [8] the authors propose a rate control algorithm for wireless sensor networks that uses knowledge of available capacity and allocates rates to flows so that they achieve a lexicographic max-min fair optimum. In [25] the nodes send back-pressure messages once they detect congestion so that the senders can throttle their sending rate. In [26] the authors propose contention window adaptation to enable the different flows in an ad-hoc network to achieve delay guarantees. They assume that there is enough capacity to support all flows, however.

A survey on the work on congestion control in the internet, which is dominated by TCP traffic, can be found in [27]. In [10] the authors propose congestion control for the Satellite MAC, where on detection on congestion the satellite broadcasts to ground terminals to reduce their rate.

VII. Conclusions

We looked at the problem of congestion control in large wireless networks where nodes periodically broadcast time-critical information. Specifically:

- We introduced the system age metric to capture in an end-to-end manner the timeliness requirements of new applications that periodically broadcast their information.
- We show that in practice minimizing age is not the same as maximizing throughput. In fact, an increase in throughput can increase the age. This precludes the use of a saturated messaging load at the optimal contention window size, for a network to achieve the minimum age in a practical setting. Instead one wants to choose a small CW, a small queue size and a large T (larger than periods that lead to a saturated load).
- Changing CW sizes is an often used technique in CSMA wireless networks to achieve greater packet delivery, for example. We show that no clear strategy exists to reduce the system age by changing the CW size.
- The above observations and the presence of a unique broadcast period at which system age is minimized motivated the design of an application layer rate control algorithm. The algorithm is fully distributed, uses the local estimate of the system age at each node. It achieves fast convergence, a median time of 4s over many simulations. It is shown not only to settle well around the system age, once achieved, for extended periods of time, but also has an ability to adapt quickly to changes in network size.

Acknowledgment

The system information age, as a metric of interest, was suggested to us by Prof. Roy Yates (at WINLAB, Rutgers University). We would also like to thank the anonymous reviewers for their comments.

References

[1] "Vehicle safety communications - applications (VSC-A) first annual report, dec. 7, 2006 through dec. 31 2007," Tech. Rep., Sep. 2008. [Online]. Available: http://www.intellidriveusa.org/documents/2009/05/09042008-vsc-a-report.pdf

[2] "Dedicated short range communications." [Online]. Available: http://www.leearmstrong.com/DSRC/DSRCHomeset.htm

[3] A. Weinfield, "Methods to reduce DSRC channel congestion and improve V2V communication reliability," Oct. 2010.

[4] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-Vehicle communication: Fair transmit power control for Safety-Critical information," *Vehicular Technology, IEEE Transactions on*, vol. 58, no. 7, pp. 3684–3703, 2009.

[5] C. Huang, H. Krishnan, R. Sengupta, and Y. P. Fallah, "Implementation and evaluation of scalable Vehicle-to-Vehicle transmission control protocol," in *VNC 2010*.

[6] D. Jiang and Q. Chen, "VSC-A congestion control," Dec. 2009, vehicle Safety Communications – Applications project internal report.

[7] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware rate adaptation for IEEE 802.11 WLANs," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, 2006, pp. 1–11.

[8] A. Sridharan and B. Krishnamachari, "Explicit and precise rate control for wireless sensor networks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*. Berkeley, California: ACM, 2009, pp. 29–42.

[9] C. Lochert, B. Scheuermann, and M. Mauve, "A survey on congestion control for mobile ad hoc networks: Research articles," *Wirel. Commun. Mob. Comput.*, vol. 7, no. 5, pp. 655–676, 2007.

[10] A. Iuoras and P. Yeung, "A broadcast congestion control scheme for obp satellites," *Space Comms.*, vol. 17, pp. 231–242, April 2001. [Online]. Available: http://portal.acm.org.proxy.libraries.rutgers.edu/citation.cfm?id=1498976.1498979

[11] F. Ye, R. Yim, J. Zhang, and S. Roy, "Congestion control to achieve optimal broadcast efficiency in VANETs," in *Communications (ICC), 2010 IEEE International Conference on*, 2010, pp. 1–5.

[12] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *Selected Areas in Communications, IEEE Journal on*, vol. 18, no. 3, pp. 535–547, 2000.

[13] B. S. Davie, L. L. Peterson, and D. Clark, *Computer Networks: A Systems Approach*. Morgan Kaufmann Publishers., 2000.

[14] Q. Chen, F. Schmidt-Eisenlohr, D. Jiang, M. Torrent-Moreno, L. Delgrossi, and H. Hartenstein, "Overhaul of ieee 802.11 modeling and simulation in ns-2," in *Proceedings of the 10th ACM Symposium on Modeling, analysis, and simulation of wireless and mobile systems*. Chania, Crete Island, Greece: ACM, 2007, pp. 159–168.

[15] H. Ma and S. Roy, "Contention window and transmission opportunity adaptation for dense IEEE 802.11 WLAN based on loss differentiation," in *Communications, 2008. ICC '08. IEEE International Conference on*, 2008, pp. 2556–2560.

[16] H. M. Alazemi, A. Margolis, J. Choi, R. Vijaykumar, and S. Roy, "Stochastic modelling and analysis of 802.11 DCF with heterogeneous non-saturated nodes," *Computer Communications*, vol. 30, no. 18, pp. 3652–3661, Dec. 2007.

[17] WINLAB, "Orbit webpage." http://www.orbit-lab.org/.

[18] C. Huang, Y. Fallah, R. Sengupta, and H. Krishnan, "Intervehicle transmission rate control for cooperative active safety system," *Transactions on Intelligent Transportation Systems, IEEE*, vol. [To Appear].

[19] "IEEE std 802.11TM-2007, IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area networks – specific requirements; part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications;," Jun. 2007.

[20] C. Sommer, R. German, and F. Dressler, "Adaptive beaconing for Delay-Sensitive and Congestion-Aware traffic information systems," *University of Erlangen, Dept. of Computer Science, Technical Report CS-2010-01*, 2010.

[21] R. Schmidt, T. Leinmuller, E. Schoch, F. Kargl, and G. Schafer, "Exploration of adaptive beaconing for efficient intervehicle safety communication," *Network, IEEE*, vol. 24, no. 1, pp. 14–19, 2010.

[22] M. S. Bouassida and M. Shawky, "Verification and performance evaluation of a congestion control approach within vehicular ad hoc networks," in *PM2HW2N'09*. Tenerife, Canary Islands, Spain: ACM, 2009.

[23] C. Robinson, D. Caveney, L. Caminiti, G. Baliga, K. Laberteaux, and P. Kumar, "Efficient message composition and coding for cooperative vehicular safety applications," *Vehicular Technology, IEEE Transactions on*, vol. 56, no. 6, pp. 3244–3255, 2007.

[24] K. Tang and M. Gerla, "Random access MAC for efficient broadcast support in ad hoc networks," in *Wireless Communications and Networking Conference, 2000. WCNC. 2000 IEEE*, vol. 1, 2000, pp. 454–459 vol.1.

[25] C. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: congestion detection and avoidance in sensor networks," in *Proceedings of the 1st international conference on Embedded networked sensor systems*. Los Angeles, California, USA: ACM, 2003, pp. 266–279.

[26] Y. Yang and R. Kravets, "Achieving delay guarantees in ad hoc networks through dynamic contention window adaptation," in *Proceedings IEEE INFOCOM 2006*, Barcelona, Spain, 2006, pp. 1–12.

[27] S. Ryu, C. Rump, and C. Qiao, "Advances in internet congestion control," *Communications Surveys & Tutorials, IEEE*, vol. 5, no. 1, pp. 28–39, 2003.