CARS: Context-Aware Rate Selection for Vehicular Networks

Pravin Shankar*, Tamer Nadeem†, Justinian Rosca† and Liviu Iftode*
*Department of Computer Science, Rutgers University. {spravin,iftode}@cs.rutgers.edu
†Siemens Corporate Research. {tamer.nadeem,justinian.rosca}@siemens.com

***Abstract*—Traffic querying, road sensing and mobile content delivery are emerging application domains for vehicular networks whose performance depends on the throughput these networks can sustain. Rate adaptation is one of the key mechanisms at the link layer that determine this performance. Rate adaptation in vehicular networks faces the following key challenges: (1) due to the rapid variations of the link quality caused by fading and mobility at vehicular speeds, the transmission rate must adapt fast in order to be effective, (2) during infrequent and bursty transmission, the rate adaptation scheme must be able to estimate the link quality with few or no packets transmitted in the estimation window, (3) the rate adaptation scheme must distinguish losses due to environment from those due to hidden-station induced collision. Our extensive outdoor experiments show that the existing rate adaptation schemes for 802.11 wireless networks underutilize the link capacity in vehicular environments. In this paper, we design, implement and evaluate CARS, a novel Context-Aware Rate Selection algorithm that makes use of context information (e.g. vehicle speed and distance from neighbor) to systematically address the above challenges, while maximizing the link throughput. Our experimental evaluation in real outdoor vehicular environments with different mobility scenarios shows that CARS adapts to changing link conditions at high vehicular speeds faster than existing rate-adaptation algorithms. Our scheme achieves significantly higher throughput, up to 79%, in all the tested scenarios, and is robust to packet loss due to collisions, improving the throughput by up to 256% in the presence of hidden stations.**

I. INTRODUCTION

In the near future, the number of vehicles equipped with computing technologies and wireless communication devices is poised to increase dramatically [1]. The Federal Communications Commision (FCC) has recently allocated the Dedicated Short Range Communications (DSRC) licensed spectrum aimed at enhancing bandwidth and reducing latency to support vehicle-to-vehicle and vehicle-to-infrastructure communication [2]. Safety applications, such as collision avoidance and braking detection, were the initial focus of vehicular networking research. Since then, various novel applications that make use of vehicular networks have been proposed, ranging from traffic management and urban sensing to multimedia sharing. Unlike vehicular safety applications, such as collision avoidance and braking warning, these novel applications are not that stringent about delay and reliability requirements, but, instead, demand high bandwidth and throughput in hostile environments with intermittent link conditions.

This work has been supported in part by the NSF grant CNS-0520123, and by Siemens Corporate Research.

Rate adaptation is a critical component to ensure optimal system performance in these dynamic mobile environments. The IEEE 802.11 protocol specifications allow multiple transmission rates at the physical layer (PHY), which use different modulation and coding schemes. For example, the 802.11p PHY offers eight different bitrates, ranging from 3 to 27 Mbps, from which transmitters can choose. Higher data rates allow high quality links to transmit more data, but have a higher loss probability on low quality links. On the other hand, a low data rate is more resilient to low quality links, but fails to achieve a high throughput in a high quality link. *Rate Adaptation* is the problem of selecting the best transmission rate based on the real-time link quality, so as to obtain maximum throughput at all times.

Several rate adaptation algorithms ([3],[4],[5],[6],[7],[8]) have been proposed in the literature. However, all the existing work in rate adaptation is based on traditional indoor wireless networks. Vehicular networks have vastly different characteristics from indoor wireless networks, as the link conditions in these networks change more rapidly due to the high mobility of the nodes. We perform a series of outdoor experiments to understand better how the existing rate adaptation algorithms in wireless networks perform in a highly mobile vehicular environment. These experiments (described in Section IV) indicate that existing schemes for rate adaptation significantly underutilize the wireless link capacity in vehicular networks.

The main problem faced by the existing rate adaptation algorithms is the delay in estimation as a result of the estimation window. All existing state-of-the-art rate adaptation algorithms consists of an estimation phase, in which an estimation window of link layer and physical layer metrics, such as frame errors and received signal strength, is maintained. The estimation window approach is *reactive* in nature, as it relies on past history for future link quality prediction, hence all the existing schemes inherently experience a delay in adaptation, which cannot be eliminated. In highly mobile environments, such as vehicular networks, the link conditions change rapidly, owing to high vehicular speeds. A second problem faced by existing rate adaptation algorithms is that they depend on packets being continuously transmitted in order to calculate the packet loss estimate. When a client periodically transmit packets in short bursts and then stay quiet, there are no recent samples available in the estimation window, causing anomalous behavior in rate adaptation algorithms. Finally, yet another challenge faced by current rate adaptation algorithms

1-4244-2507-5/08/$20.00

is the need for differentiation of various types of losses. A frame loss induced by environmental factors should trigger rate adaptation to recover from the loss. However, a hidden-station induced loss should not trigger rate adaptation, as reducing the transmission rate cannot solve the contention problem.

In summary, rate adaptation in vehicular networks faces the following key challenges: (1) due to the rapid variations of the link quality caused by fading and mobility at vehicular speeds, the transmission rate must adapt fast in order to be effective, (2) during infrequent and bursty transmission, the rate adaptation scheme must be able to estimate the link quality with few or no packets transmitted in the estimation window, (3) the rate adaptation scheme must distinguish losses due to environment from those due to hidden-station induced collision.

In vehicular networks, each node already possesses context information about the environment, in the form of the location and speed of itself and its neighbor. Almost all vehicular applications make use of location and neighbor information, so it is reasonable to assume that this information will be available on any vehicular node. This information can be easily obtained by means of a GPS device and an application such as TrafficView [9], or a VANET middleware [10]. The key insight we make is that this context information is a direct, predictable and real-time indicator of the link quality. and can be used to perform *fast* rate adaptation. In this paper, we propose Context Aware Rate Selection (CARS), a novel rate adaptation mechanism for VANETs that uses the context information to *learn* the real-time link quality. We show how the algorithm can be realized with no modifications to the 802.11 MAC protocol or hardware, by simply exchanging cross-layer context information between the application layer and the MAC layer. We implement CARS in the Linux Madwifi driver, and evaluate its performance by means of extensive outdoor experiments in real vehicular environments.

The key contributions of this paper are:

1) We show by means of real experiments why existing rate adaptation algorithms underutilize the link capacity in vehicular networks;
2) We design, implement and test CARS, a novel rate adaptation mechanism that uses context information to adapt to fast changing link conditions specific to vehicular networks.

The rest of the paper is organized as follows: section II introduces the background and related work and in section III, we describe the experimental methodology and setup. Section IV presents the challenges faced by rate adaptation algorithms in vehicular environments. In section V, we explain the CARS algorithm. In section VI, we present experimental results evaluating CARS's performance, and in section VII, we present results from our simulation study. We discuss some open issues and limitations of the CARS approach and present scope for future work in section VIII, and conclude in section IX.

II. Background and Related Work

Auto Rate Fallback (ARF) [3] was the first rate adaptation algorithm proposed way back in 1996 and it is also the simplest algorithm. Since then, numerous other rate adaptation algorithms have been proposed. Some of the popular existing rate adaptation algorithms in wireless networks are Adaptive Auto Rate Fallback (AARF) [6], ONOE [4], Receiver-Based Auto Rate (RBAR) [5], SampleRate [7], Collision Aware Rate Adaptation (CARA) [11] and Robust Rate Adaptation Algorithm (RRAA) [8]. Current rate adaptation algorithms in literature are all designed for wireless networks in general. To the best of our knowledge, there is no rate adaptation algorithm specifically designed for VANETs.

The ARF scheme [3] consists of dropping the transmission rate on successive packet losses, and increasing the rate on successive successful packet transmits or timeout. AARF [6] improves the stability of the scheme by using dynamic instead of fixed frame error thresholds to decrease or increase the rate. ONOE [4] is a popular rate adaptation algorithm whose implementation is available in the MADWifi driver code. It decreases the bit-rate when packets need at least 1 retry on average, and increases the bit-rate when less than 10% of packets require a retry. The problem with such schemes is that most realistic scenarios exhibit randomly distributed loss behaviors. Any deterministic pattern of consecutive transmission successes/failures may not occur with large probability in all cases.

Receiver Based AutoRate (RBAR) [5] is a signal-to-noise ratio (SNR) based scheme that uses feedback from the receiver to select the sender's optimal rate. In this scheme, the sender sends an RTS frame before every packet, and receiver measures the SNR and compares it with SNR thresholds from an apriori calculated wireless channel model, calculates the optimal rate, and sends it back to the sender as part of the CTS frame. SNR-Guided Rate Adaptation (SGRA) [12] performs a measurement study of SNR as a prediction tool for channel quality. CHARM [13] is a SNR-based scheme which uses channel reciprocity to obtain channel information, thus not incurring RTS/CTS overhead. The problem with using SNR as an estimate of channel quality is that in a rapidly changing channel, SNR can periodically fluctuate, leading to misleading predictions. Averaging the SNR values over a window will lead to the same estimation delay problem as in the frame error based schemes.

SampleRate [7] is a throughput-based scheme that aims to minimize the mean packet transmission time. It chooses the bit-rate that it predicts will provide the most throughput based on estimates of the expected per-packet transmission time for each bit-rate. SampleRate uses the idea of probing, in which it periodically sends packets at bit-rates other than the current one to estimate when another bit-rate will provide better performance. Using probe packets has two problems, as shown in [8]. Firstly, a successful probe can be misleading and trigger incorrect rate increase. Secondly, a statistical update based on probes is too sensitive to failure of probe packets;

an unsuccessful probe can incur severe penalty on future rate adaptation. This problem becomes particularly significant in highly dynamic environments, such as vehicular networks.

Robust Rate Adaptation Algorithm (RRAA) [8] does not use probe packets. Instead, it calculates packet loss using an estimation window and measures it against empirical low and high packet loss thresholds for different bitrates, in order to choose the correct bitrate. RRAA's approach is to minimize the delay due to the estimation window by using a short-term loss ratio and making the estimation window's size adaptive. The problem with a small estimation window size, especially during contention from multiple active clients, is that the client needs to get enough frame transmission statistics in this time, or else a statistically small number of samples may yield inaccurate rate adaptation.

Another problem with rate adaptation based on frame transmission results is that they cannot distingush frames lost due to collisions from frames lost due to channel errors. This problem is further aggravated in the presence of hidden terminals. Collision-Aware Rate Adaptation [11] proposes a scheme that makes use of RTS/CTS frames to differentiate frame collisions from frames lost due to channel errors [14]. Since these control frames introduce an overhead, they are not always used but are turned on upon a frame loss and turned off upon a frame success. However, in the presence of hidden terminals, this scheme can result in RTS/CTS oscillations. RRAA [8] uses an adaptive RTS scheme to adapt to the dynamic collision level incurred by hidden stations. Our approach to rate adaptation in the presence of collisions is orthogonal to existing RTS/CTS based approach. We completely eliminate the need for RTS/CTS control frames while keeping the rate adaptation robust to collisions caused by hidden terminals.

Very recently, a concurrent study conducted by Camp and Knightley [15] proposes a cross-layer framework for rate adaptation in vehicular networks. Similar to our work, the study concluded that ”in-situ” training can help overcome the sensitivity problem of the existing rate adaptation schemes. They also indicated the importance of differentiating between collision and channel noise for rate adaptation schemes in vehicular networks.

III. Experimental setup

The experimental results presented in this paper have been obtained using real experiments conducted in outdoor vehicular environments. In this section, we describe our experimental setup, platform and methodology.

A. Hardware and software configuration

The experiments were performed with two vehicles equipped with computing devices, GPS devices, and wireless interfaces augmented with omni-directional antennae. We use off-the shelf hardware for conducting our measurements instead of using sophisticated channel sensing equipment because of two reasons. Firstly, such equipment is prohibitive in cost. Secondly, using off-the-shelf hardware makes our results comparable to realistic scenarios that users of 802.11 equipment in vehicles can expect. The hardware and software environment used in the experiments is described in Table I.

TABLE I
Default experimental parameters

Parameter	Value
Hardware	HP Pavilion laptop with Intel Pentium M 1.8 GHz processor
Operating System	Linux Kernel 2.6.19
Wireless Card	Atheros 5212 chipset
Driver	MadWifi-ng
PHY and MAC Protocol	802.11a
Frequency	5.805 GHz
Transmit Power	40 mW
Antenna Type	folded dipole
Antenna Gain	3dBi
Data Payload Size	1000 bytes
Transmission Frequency	1 packet every 20 ms (50 packets every second)

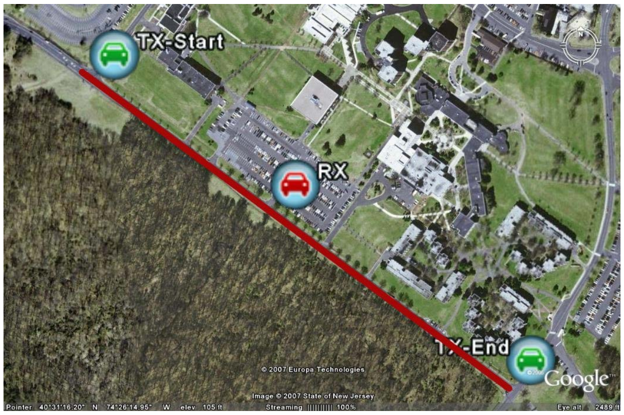
Our measurements are based on 802.11a, since these radios are more readily available, as compared to “pre-standard” 802.11p radios. In addition, 802.11a MAC and PHY protocols are similar to those under consideration in the emerging 802.11p standard. Both 802.11p and 802.11a support the same modulation and coding schemes as well as training sequences. We disable the 802.11a association protocol by operating the nodes in a special *pseudoIBSS* mode. This mode eliminates beacon synchronization and IBSS merging problems that exist in IBSS implementations in an easy way by completely removing beaconing. We chose a low transmit power of 40mW, to reduce the amount of space needed for our experiments. The results could be scaled to higher transmit powers considered in DSRC.

B. Design of experimental scenarios

The experiments are conducted on different outdoor environments ranging from a campus parking lot to campus roads to a inter-state freeway. The trajectories of the parking lot and the campus roads scenarios are shown in Figure 1(a) and Figure 1(b). The freeway scenario consists of an inter-state freeway with average speed of 65 mph. Two cars are used in all experiments, one configured as a transmitter and the other as a receiver, except when noted otherwise.

C. Workload

Each experiment consists of the transmitter sending a frame every 20 ms (50 frames every second) and the receiver logging the received signal strength indicator (RSSI) and the packet error rate (PER). We modified the iperf program to log individual packet transmission statistics, and we control the duration of time between two outgoing frames, each a 1472 byte payload ICMP packet, to be on the order of hundreds of microseconds and assign it the highest run-time priority. Using this approach, we were able to consistently generate packets at millisecond granularity, without noticeable packet loss in indoor tests. The packet error rate is computed by making use



(a) Scenario 1 consists of a stationary sender in an empty parking lot and moving receiver driving from one end of the road to the other with LOS.

(b) Scenario 2 consists of moving sender and receiver driving in loops across the trajectory shown

Fig. 1. The experimental scenarios for the distance and mobility experiments

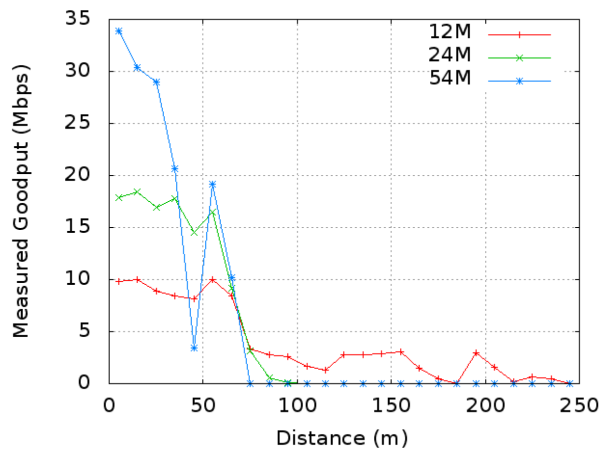


Fig. 2. Measured goodput at different distances for different fixed rates.

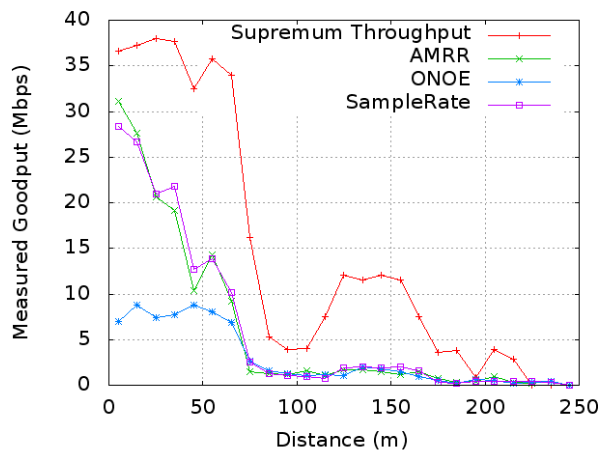


Fig. 3. Measured goodput averaged over distance for different state-of-the-art rate adaptation algorithms compared against the supremum goodput.

of 32-bit sequence numbers, which are incremented by the transmitter for every packet transmitted. The sequence number is transmitted as part of the ICMP payload.

In addition, the nodes continuously log their location and speed information using a GPS device once per second. The system time on each node is set to the GPS time so that the system clocks between the nodes are synchronized. The transmitter includes its timestamp in the ICMP packets payload so that the receiver can correlate its GPS record with the corresponding GPS record of the transmitter.

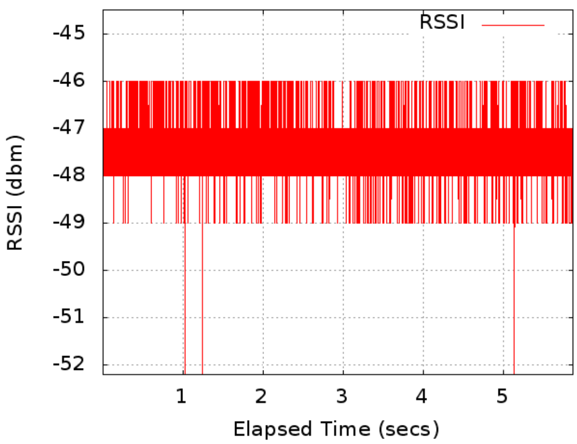
IV. Challenges Faced By Rate Adaptation In Vehicular Networks

We perform a series of outdoor experiment in order to understand the problems encountered by rate adaptation in vehicular networks. The experiments are conducted in a campus parking lot setting, where we measure average goodput received by a vehicular client at different distances, first by different fixed transmission rates, and subsequently by current state-of-the-art rate adaptation schemes.

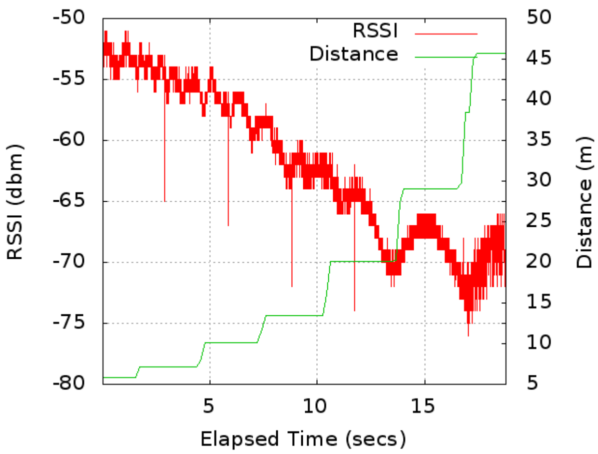
Figure 2 shows the average measured goodput at different distances, achieved by three different fixed PHY transmission rates. From the figure, we observe that rate adaptation is essential because at different distances, different rates achieve optimal goodput. In fact, the highest bitrate of 54 Mbps causes the transmission range to be 60m, whereas at the lowest bitrate of 6 Mbps, the transmission is 240m (four times the transmission range). Figure 3 plots the average goodput achieved by three current state-of-the-art rate adaptation schemes, AMRR, ONOE, and SampleRate, in the same experimental scenario. In order to provide a benchmark for comparison, we compute from our earlier experiment with fixed bitrates, the maximum goodput possible at each distance range, which we term the *Supremum Throughput* for each distance bin. The figure shows that there is a significant underutilization of link capacity, which deteriorates link goodput in vehicular clients.

These results show that there are some important challenges faced by rate adaptation algorithms in vehicular environments, which we enumerate below:

- **SNR variation over time**: Physical-layer metrics, such as the signal to noise ratio (SNR), can be used for rate estimation, as they are a good indicator of channel quality. However, use of SNR encounters practical difficulties,



(a) RSSI trace in a stationery environment



(b) RSSI trace when the cars are moving

Fig. 4. Evolution of RSSI over time

because of the SNR variations over time. Figure 4(a) shows an RSSI trace when the two vehicles are stationary and close to each other. In our experiments, we notice spikes in RSSI over time for a stable link, with an average variation of 5 dB and peaks of upto 10-14 dB. In the case of a moving car, as shown in Figure 4(b), even though the RSSI spikes are larger, the large-scale path loss is much more significant than small-scale fading. The Y2 axis shows the distance between the two cars, and, as we would expect, the distance is a great estimator of the large-scale fading. In vehicular networks, nodes typically already possess their location information by means of a GPS device. We make use of this key observation in designing our rate adaptation algorithm.

- **Estimation window induced delay**: All existing state-of-the-art rate adaptation algorithms consist of an estimation phase and an action phase. In the estimation phase, an estimation window of link layer and/or physical layer metrics, such as frame errors and received signal strength, is maintained. Packet error estimation window introduces delay in rate adaptation, as the estimation window approach is *reactive* in nature, relying on past history for future link quality prediction. In highly mobile environments such as vehicular networks, the link conditions change rapidly, owing to high vehicular speeds. As a result, the estimation window could contain a lot of stale samples.
Prior work [8] used mutual information analysis to determine the maximum time interval up to which there is some correlation across transmissions. The authors found the value for indoor wireless environments to be 150 ms. Vehicular environments being more dynamic than indoor wireless environments, this value is expected to be even smaller for vehicular environments. An estimation window larger than this value could contain stale samples that affect the accuracy of rate adaptation. On the other hand, a very small estimation window suffers from the problem of statistical errors due to insufficient samples. Further, in the presence of multiple active stations, since the transmissions of the stations are multiplexed, the number of samples may further decrease, making it harder for the rate adaptation algorithm to infer the channel conditions.
- **Idle station problem**: With an idle station, since there has been no transmission in the recent past, there is no way estimation window based schemes can work. In vehicular networks, a MAC-layer packet scheduling algorithm [16] can cause clients to periodically transmit packets in short bursts and then stay quiet. As a result, when the client starts transmitting, there are no recent samples available in the estimation window, which introduces anomalous behavior in rate adaptation. There is no prior known solution to the idle station problem, and this problem becomes even more significant in vehicular networks.
- **Hidden station problem**: When there are collisions as a result of hidden stations, the packet error statistics become polluted with collision-induced losses. However, a hidden-station induced loss should not trigger rate adaptation, as reducing the transmission rate cannot solve the contention problem. Rate adaptation in such a scenario actually aggravates channel collisions, since a lower transmission rate prolongs the transmission time for each frame. Existing algorithms ([11], [8]) suggest adaptive RTS/CTS schemes to alleviate the hidden station induced losses. RTS/CTS, however, incurs an overhead because of the additional control frames transmitted before every data frame.

V. Context Aware Rate Selection

This section describes the design and implementation of CARS, a novel context-aware rate adaptation algorithm for vehicular networks, designed to adapt to the high mobility vehicular environments.

A. Overview

The core idea of CARS is to make use of context information from the application layer, in addition to the frame transmission statistics received from the lower layers. Figure 5 shows the overall system architecture of the CARS scheme.

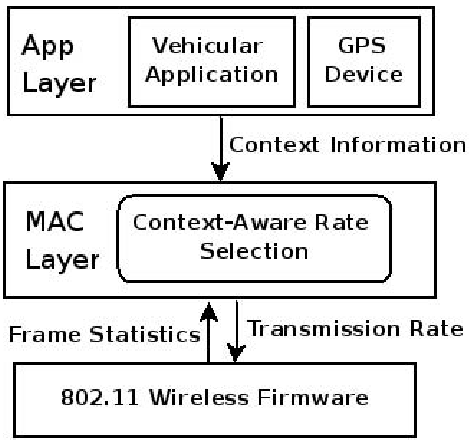


Fig. 5. The CARS system architecture

The context information used in CARS broadly consists of information about the environment that is available to the node and which has an effect on the packet delivery probability. Such information could include the position, speed and acceleration of the vehicle, the distance from the neighboring vehicle, and environment factors such as location, time of day, weather, type of road and traffic density. As a starting point, we choose the two most significant of these parameters, viz. distance from the receiver and and the vehicle's speed. Vehicles that are part of a vehicular network typically run either a safety application [17], [9] or a VANET middleware [18], [19]. These applications/middleware obtain the position and speed of the vehicle from a GPS device and use the wireless device to periodically broadcast their location and receive location information from neighboring vehicles. Thus, the vehicles possess information about their position and speed as well as the position and speed of their neighbors. These constitute the context information that CARS obtains from the application/middleware.

B. Rate Adaptation Algorithm

Algorithm 1 describes the CARS algorithm. The key idea of the CARS algorithm is to estimate the link quality using both context information as well as past history. The CARS rate selection algorithm estimates the packet error by means of a weighted decision function involving two functions, E_C and E_H (line 4 of algorithm). The function E_C uses the context information, transmission rate and packet length as input parameters, and outputs the estimated packet error rate. We describe in detail in the next subsection, how we derive this function. The function E_H uses an exponentially weighted moving average (EWMA) of past frame transmission statistics for each bitrate, similar to schemes such as SampleRate [7].

Context information is represented by the variable ctx. The weight α determines whether to give preference to the context information or to the EWMA. α is assigned based on the vehicle speed.When speed is zero, there is no opportunity for doing any prediction of link quality using context information, so EWMA is given preference. On the other hand, when vehicle speed is high, context information is given preference. More precisely, $\alpha = max(0, min(1, speed/S))$. We experimented with different values of the speed normalizer, S, and we select $S = 30$ (metres per second) as the best value, which corresponds to a vehicle speed of about 65 miles per hour. The algorithm calculates estimated throughput for each bitrate and selects the bitrate that it predicts will provide the most throughput. N is the maximum number of retransmissions, and $avg\_retries$ computes the average number of retransmissions (line 5 of algorithm). ρ is the weight that signifies the penalty given to unsuccessful packet transmission. We experimented with different values of ρ and we select $\rho = 8$ as the best value.

Algorithm 1 The Context Aware Rate Selection Algorithm

Function $CARS\_GetRate$
Input: ctx, α, len
Output: $rate$

1: $Max\_Thr \leftarrow 0$
2: $Best\_Rate \leftarrow MIN\_RATE$
3: **for all** $rate$ **do**
4: $PER = \alpha.E_C(ctx, rate, len) + (1-\alpha).E_H(rate, len)$
5: $avg\_retries = (N.PER^{(N+1)} - (N+1).PER^N + 1)/(1-PER) + N.PER^N$
6: $Thr = Rate/avg\_retries.(1-PER^N)^\rho$
7: **if** $Thr > Max\_Thr$ **then**
8: $Best\_Rate \leftarrow bitrate$
9: $Max\_Thr \leftarrow Thr$
10: **end if**
11: **end for**
12: Return $Best\_Rate$

The context information and the EWMA do not need to be recalculated for every transmission. Even at very high mobility (vehicular speed of 65 mph), in 100 ms, the vehicle moves less than 3 metres. Therefore, we set the recalculation frequency to be 100 ms.

The wireless Hardware Abstraction Layer (HAL) provides a multirate retry chain, which consists of four segments. Each segment is an advisement to the HAL to try to send the current packet at some rate, with a fixed number of retry attempts. Once the packet is successfully transmitted, the remainder of the retry chain is ignored. We describe our CARS retry chain in Table II. The reasoning behind our selected retry chain is that on first failure, the weight, α, assigned to context information is reduced, on a subsequent failure, a purely EWMA based scheme is used, and finally, the lowest possible rate is tried.

TABLE II
CARS RETRY CHAIN

Attempt	Value
1	CARS_GetRate(ctx, α, len)
2	CARS_GetRate($ctx, \alpha/2, len$)
3	CARS_GetRate($ctx, 0, len$)
4	Lowest baserate

C. CARS Model

The CARS scheme makes use of an empirical model to *learn* the effect of context information on the packet delivery probability. We introduce the function $E_C(ctx, bitrate, len)$ in the Algorithm 1, representing this empirical model, which uses the context information, transmission rate and packet length as input parameters, and outputs the estimated packet error rate. As explained before, the context information in CARS, represented by the variable ctx, refers to information about the environment that is available to the node and which has an effect on the packet delivery probability. In this paper, we consider ctx to consist of two parameters, viz. distance from the receiver and and the vehicle's speed.

Several analytical and empirical models for radio frequency (RF) propagation in free space have been proposed in the literature. To model the effect of distance, we could use the free space path loss model, the two ray propagation model or a more complex fading model. Models such as the delay tap model [20] or ray models with delay profiles [21] can be used to model the effect of speed. Instead of using existing RF propagation models, we choose to derive a simple empirical model for delivery probability using measurements from real outdoor vehicular experiments. This is because the goal of our work is not to build an elaborate model of vehicular channel conditions in the presence of small scale fading and large scale shadowing. Instead, we want to show that model-based schemes can improve rate adaptation, even with a simple model.

The goal of the model is to predict packet error rate (PER) as a function of the distance between the vehicles, the relative speed between the vehicles, and the transmission rate. We use multivariate linear regression as the learning approach. Measurements from extensive outdoor vehicular experiments were used to build this empirical model, in which we vary the distance between the vehicles, the speed, the packet size and the bitrate, as described in Section III.

Linear regression models can be too restrictive, so we also construct higher-order polynomial multivariate regression models that fit our data. We compared the least mean squared errors generated by the linear, quadratic and cubic regression models and found that the difference is not very significant. Further, a linear model has lesser computational overhead.

The CARS model estimates packet delivery probability based on just two parameters, distance between vehicles and relative speed. In a real VANET scenario, there are many other effects that cause packet loss, such as multipath effect, shadowing due to obstructing vehicles, environmental effects, such as rain or snow, background interference, etc. Therefore, the predicted packet loss will be inaccurate as a result of these effects. However, the reason why the CARS scheme works in practise is because there is scope for error in PER estimation. The possible bitrates in IEEE 802.11-based wireless cards are discrete. Even if the predicted packet loss is not accurate, it is sufficient if the estimated packet loss by the model allows CARS to selects the optimal bitrate. Further, even if we are off by one rate, the effect is not significant.

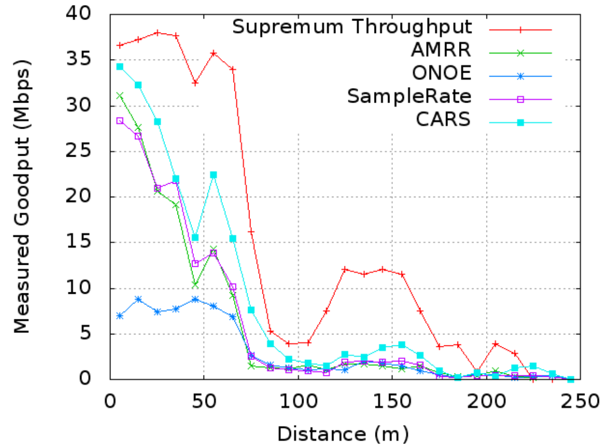


Fig. 6. Measured goodput averaged over distance for CARS compared with different state-of-the-art rate adaptation algorithms. We also show the the supremum goodput at each distance for comparison.

VI. EXPERIMENTAL EVALUATION

We implemented the CARS algorithm on the open-source MadWifi wireless driver for Atheros chipset wireless cards. The implementation consisted of 520 lines of C code. Context information required for CARS was obtained using GPSDaemon, a VANET application that interfaced with the wireless driver using a generic /proc interface. Any other VANET application can be extended to use this same interface, so CARS can be deployed with no change to the 802.11 protocol or to the hardware. In this section, we present results from our detailed evaluation of CARS by means of outdoor field trials, as well a simulation study.

A. Underutilization of link capacity

In section IV, we presented experimental results (Figure 3) that showed the underutilization of link capacity in outdoor vehicular environments by current state-of-the-art rate adaptation algorithms. We repeated the same experiment with our CARS implementation, and we show the results in Figure 6. The figure shows that CARS consistently outperforms all the existing rate adaptation algorithms at all distances. However, there is still a lot of scope for improvement compared with the supremum throughput.

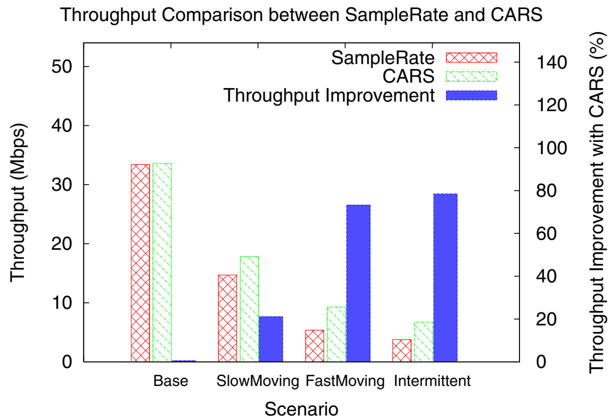


Fig. 7. Histogram showing the comparison of throughput achieved by CARS and SampleRate in different scenarios

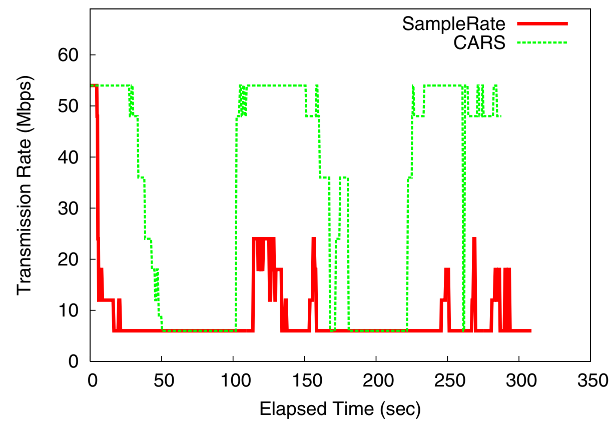


Fig. 8. Detailed comparison of the rate adaptation performed by CARS and SampleRate for the Intermittent scenario

B. Effect of environment

We compare CARS and SampleRate [7], the default rate adaptation algorithm used in the MadWifi driver codebase, by means of outdoor experiments lasting 5 minutes each in four different mobility scenarios, which are as follows:

- Base: The two cars are stationary next to each other.
- SlowMoving: Slow moving scenario - The two cars are moving around our campus at 25mph speed each, following each other.
- FastMoving: High speed moving scenario - The two cars are moving on an interstate highway at 70mph speed each in high car/truck traffic conditions, following each other.
- Intermittent: Intermittent connectivity - A more stressful intermittent connectivity scenario where cars are mostly out of range and periodically meet each other.

Figure 7 shows the comparison of throughput between CARS and SampleRate for 5 minute experiments in these five scenarios. In the Base scenario, both CARS and SampleRate give the same throughput. In all the other scenarios, CARS gives significantly better throughput, and the more stressful the condition in terms of varying distance and speed, the more is the throughput gain. In the SlowMoving, FastMoving and Intermittent scenarios, the throughput improvement is 21 %, 73 % and 79% respectively.

The reason CARS performs better in the stressful scenarios is because it adapts the bitrate faster as the conditions change. Figure 8 explains why CARS performs better using a detailed analysis of rate changes in the most stressful scenario - *Intermittent*. The figure shows a detailed plot of rate adaptation by both SampleRate and CARS in this scenario. As the distance increases, both algorithms reduce the bitrate, but when they come back in range, CARS quickly gets back to the highest bitrate, while SampleRate takes much longer time to converge to the optimum bitrate.

C. Effect of hidden-node collisions

We perform outdoor experiments comparing the performance of CARS with SampleRate in the presence of packet losses caused by hidden-node induced collisions. The experiment consists of 4 nodes that are stationary during the course of the experiment, viz. the transmitter (TX), the receiver (RX), a second receiver (RX2) and the interferer (IX). TX and IX are out of transmission and carrier sensing range of each other, so IX acts as a hidden node when TX is sending frames to RX. The distance between IX and RX is fixed such that when RX hears from IX during the time a frame is transmitted from TX, it results in packet error due to collision. We ensure that the capture effect does not come into play. i.e. $\frac{P_{TX}}{P_{IX}} < \gamma$ where P_{TX} and P_{IX} are the received power from TX's packet and IX's packet, respectively, and γ is the threshold ratio above which the packet is successfully received.

When sending packets, we want to measure only packets lost due to collisions and not packets lost due to queueing. So, the application layer transmission rate at TX is determined based on the maximum actual throughput that is possible at the lowest rate. The maximum actual possible throughput is lesser than the actual transmission rate (eg. at 54 Mbps PHY transmission rate, the maximum possible throughput is approximately 33 Mbps). At the lowest transmission rate (6 Mbps), the maximum possible throughput is approximately 5.1 Mbps, so, to ensure no packet loss in the base case, we select 3 Mbps as the application layer transmission rate at TX for our experiments. We use a modified *iperf* program in which the following metrics are logged per frame transmitted: $sequence\_number, transmission\_rate, PER, RSSI$.

The experiment is performed once with SampleRate as the rate adaptation algorithm at TX and again with CARS as the rate adaptation algorithm at TX. There are two scenarios, viz. :

- *Base*: TX sends packets to RX using iperf at the application rate of 3 Mbps for 5 minutes (300 seconds).
- *Hidden-Node* The same step is repeated but this time IX sends packets to RX2 using iperf at the application rate

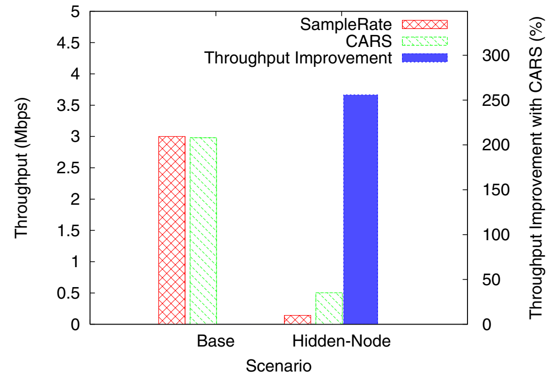


Fig. 9. Histogram showing the comparison of throughput achieved by CARS and SampleRate in the Hidden-Node experiment

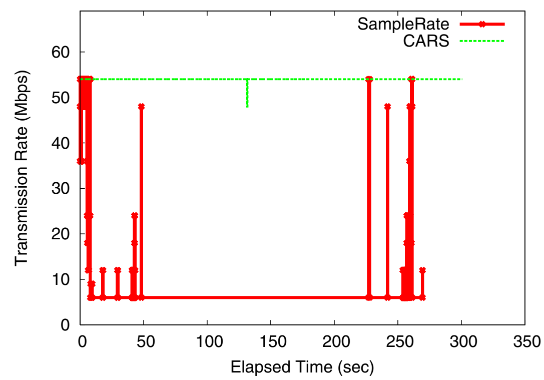


Fig. 10. Detailed comparison of the rate adaptation performed by CARS and SampleRate in the Hidden-Node scenario

of 54 Mbps during the time of the experiment.

Figure 9 shows the comparison of throughput between CARS and SampleRate for 5 minute experiments in these two scenarios. In the Base scenario, both CARS and SampleRate give the same throughput. In the Hidden-Node scenario, CARS gives significantly better throughput, with a throughput improvement of 256%.

The reason CARS performs so much better than SampleRate in the Hidden-Node scenario is because it does not adapt the bitrate when packet loss occurs due to collision. Figure 10 explains why CARS performs better in the Hidden-Node by showing a detailed plot of rate adaptation by both SampleRate and CARS. As the collisions increase, SampleRate reduces the bitrate, but CARS does not consider packet losses due to collisions while performing rate adaptation.

VII. Simulation Evaluation

In this section we present results from extensive simulation-based studies comparing the performance of CARS scheme with popular rate adaptation algorithms SampleRate [7] (we refer to it as Sample through this section) and AARF [6] under different large scale and high-density scenarios. The simulations are performed using the *ns-2* simulator [22] with IEEE 802.11a as the underlying link layer and a transmission range of 250 meters*. We extended the bit transmission error model in the physical layer of ns-2 to match our outdoor experiments with respect to the distance and relative speed between vehicles. The CARS, Sample and AARF schemes for rate selection are implemented in MAC layer of ns-2. CARS is implemented as described in Section V while Sample and AARF schemes are implemented as described in [7] and [6] respectively. None of our implemented schemes adopts the multirate retry chain described in Section V. In addition, we fix the maximum number of retransmissions of any scheme to 4. We make use of a microscopic traffic generator tool we developed, which can generate traffic scenarios for hotspots, highways, as well as cities with grid roads (e.g., Manhattan). This traffic generator accepts as parameters the simulation time, road length, number of lanes per road, average speed of the vehicles, and the number of vehicles on the road.

In our scenarios, each vehicle acts as either a client or a server. Vehicles periodically broadcast very short packets to update the neighbors about their new context (location and speed). Once a vehicle acting as a client detects a server in its neighborhood, it establishes a UDP connection to upload/download a video stream to/from the server. We experimented with various vehicle speeds and densities in different environments, as will be shown in the following subsections. For all these runs, a vehicle selects the next update period uniformly from $[1.75, 2.25]$ seconds. We assume each video stream consists of 1500 data packets of size 1000 byte. Packet transmission rate for UDP connection is 100 packets per second. No RTS/CTS is used and the number of MAC layer retransmission is set to 4 trials. Each value in the results is averaged over 10 runs for each scenario.

A. Metrics

We employ the following metrics in the evaluation:

- *Number of Packets* is the average number of packets transmitted/received per client-server connection. This metric counts each individual retransmission of the IEEE 802.11 MAC layer for a single data packet.
- *Airtime* is the corresponding airtime usage, measured in second, for the transmitted/received packets per client-server connection. This metric is critical in performance evaluation because different schemes will use different data rates for data transmissions.
- *Goodput* is the average number of bits delivered per second. It is measured as the total bits received from a connection divided by the corresponding transmission airtime.
- *Load* is the average transmission airtime needed to deliver a single packet. Ideally, selection of a low data rate for transmission yields high load compared to a high data rate. However, with bad links, selection of high data rate may add more load because of increased number of retransmission.

*Although the actual wireless transmission range may be less than 250 meters for outdoor application, this transmission range could be restored with the use of external antennas.

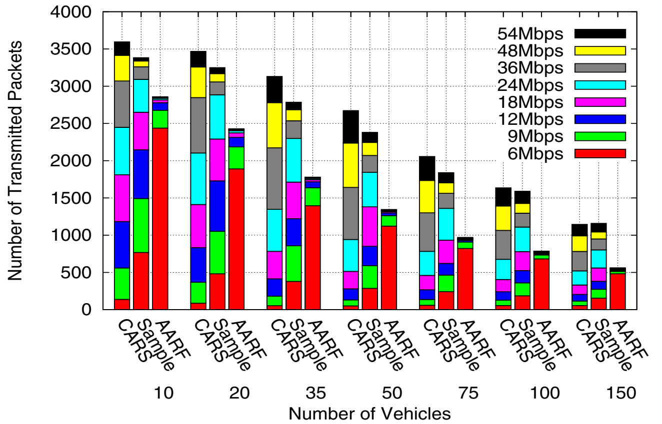


Fig. 11. Packet distributions with selected data rates for Hotspot uploading scenarios

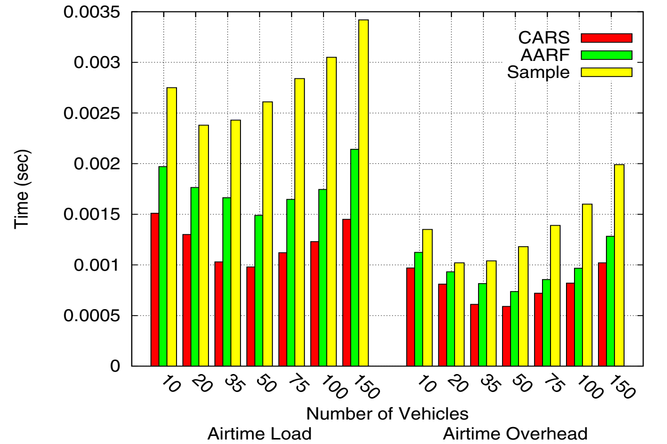


Fig. 12. Overhead and Load metrics for Vehicle-to-Infrastructure uploading scenarios

- *Overhead* is the average non-useful transmission airtime needed to deliver a single packet. It is measured as the difference between the reception airtime and the transmission airtime per delivered packet. Retransmissions over bad links are the reason for high overhead.

B. Vehicle-to-Infrastructure Scenario

In this scenario, all vehicles act as clients. We use a fixed base station as server. This scenario is very typical in cities and highways having road-side units (e.g., kiosks and cafes) with wireless services. Our scenario consists of a road of length 5000 m with multiple lanes. The base station is located at the middle of the road. Vehicles select their speeds uniformly over the range $[Speed_{avg} * 0.75, Speed_{avg} * 1.25]$ Km/h, where $Speed_{avg}$ and number of vehicles are inputs to the traffic generator. All vehicles start at the same time at the beginning of the road and move towards the end of the road, crossing the server on the way. Once a vehicle is in the range of the server, it establishes a connection immediately. We experimented with different number of vehicles and speeds, as well as downloading versus uploading connections.

Table III shows the average number of transmitted packets by each client to the server, the average number of received packets by the server per client, the corresponding airtime usage, and the goodput when clients are uploading to the server versus different number of vehicles for the different schemes. The $Speed_{avg}$ is selected to be 55Km/h, which results in average duration of 33 second for each connection. We observe that, although CARS scheme transmits more packets than both Sample and AARF schemes, the transmission airtime of CARS is lower than both. This indicates that CARS selects higher data rates for transmission.

Interestingly, Table III indicates that CARS managed to select higher data rates, which results in better overall performance as indicated by the number of delivered packets. To verify this observation, we plotted the number of transmitted packets with respect to data rates used for the three schemes in Figure 11. As shown in this figure, CARS scheme exploits the available higher data rates better than the other schemes. As described earlier in the paper, CARS overcomes the challenges in vehicular environment, such as short duration of the connection and the fast change in link conditions, which prevent other schemes from selecting the optimum data rate. Note that Sample scheme outperforms AARF scheme in exploiting higher data rates. This is due to the enhancements in Sample scheme that uses data rate with the minimum average transmission time in addition to probing for higher data rates periodically.

Using optimum higher data rates allows CARS scheme to reduce the network load, as indicated in the transmission airtime in the table and the plotted *Load* and *Overhead* metrics in Figure 12. The figure shows that Sample and AARF add load per received packet up to 60% and 165% more than CARS, respectively. Although Sample and AARF use more frequent lower data rates, the overhead of both schemes are still higher than CARS. This is because in highly dynamic vehicular environments, using lower data rates cannot guarantee to eliminate the transmission errors completely. A single retransmission with lower data rates adds too much overhead to the network. Unlike Sample and AARF, CARS tries to reduce the load through the use of higher data rates and permitting more retransmissions with increased relative overhead. Figure 12 shows that the overhead increase in Sample and AARF with respect to CARS is not as high as load increase (e.g., AARF adds overhead up to 100% more than CARS compared to the 165% increase in load).

An interesting observation from Figure12 is that load and overhead decrease as number of vehicles increases before it increases again. This is because, as number of vehicles increases, the duration needed per vehicle to upload the stream is extended due to greater medium contention. This allows the vehicles to upload more packets using higher rates while they are closer to the server, which minimizes the transmission airtime and the number of retransmissions. As the number of vehicles continues increasing, the number of packets transmitted per vehicle when it's near the server reduces due to

	Number of Packets						Airtime Usage (sec)						Goodput (Mbps)		
	Send			Recv			Send			Recv					
Veh. No.	$CARS$	$Sample$	$AARF$	$CARS$	$Sample$	$AARF$	$CARS$	$Sample$	$AARF$	$CARS$	$Sample$	$AARF$	$CARS$	$Sample$	$AARF$
10	3595.35	3383.13	2858.37	1461.96	1453.23	1481.01	2.200	2.721	4.072	0.780	1.235	2.082	5.315	4.273	2.909
20	3468.24	3249.45	2427.59	1471.17	1459.05	1411.79	1.912	2.283	3.339	0.718	1.072	1.916	6.154	5.112	3.382
35	3131.10	2784.6	1779.87	1423.09	1200.80	1041.63	1.454	1.997	2.466	0.593	0.964	1.439	7.833	4.811	3.379
50	2672.84	2379.95	1344.85	1211.86	1034.33	741.31	1.163	1.540	1.900	0.472	0.778	1.054	8.336	5.373	3.121
75	2056.01	1838.85	969.64	867.02	720.16	503.61	0.941	1.186	1.372	0.348	0.570	0.723	7.368	4.857	2.936
100	1636.08	1590.00	785.67	654.66	574.09	378.98	0.776	1.002	1.116	0.270	0.447	0.551	6.745	4.584	2.716
150	1145.21	1158.71	561.89	412.00	355.20	247.41	0.579	0.761	0.789	0.178	0.305	0.360	5.689	3.736	2.509

TABLE III
THE NUMBER OF PACKETS, AIRTIME, AND GOODPUT FOR VEHICLE-TO-INFRASTRUCTURE SCENARIO WITH UPLOADING CONNECTIONS VERSUS DIFFERENT NUMBER OF VEHICLES.

the contention. Hence, vehicles continue their uploads while they are moving away from the server. As vehicle moves away from the server, vehicles switch to lower data rates to cope with the change in link conditions. This is illustrated in Figure 11, which shows that the number of packets transmitted with higher rates (i.e., 36Mbps, 48Mbps, and 54Mbps) is larger for scenarios with moderate number of vehicles (i.e., 35, 50, and 75 vehicles).

In summary, CARS exploits better the higher data rates, thus reducing the network load and allowing better data packet delivery ratio. Therefore, CARS scales better than other schemes with increasing the number of vehicles in the network as indicated in Table III. The table shows that both number of packets and airtime per vehicle are decreasing with the increase of the vehicles in the network for all schemes due to the contention over the shared bandwith. However, since CARS uses lower transmission airtime, the reduction rate in packets delivery is lower than Sample and AARF schemes. As an example, for scenario with 20 vehicles, CARS and AARF deliver almost identical number packets. However, when we increase the contention by increasing the number of vehicles to 75, CARS outperforms AARF and delivers about 72% more packets than AARF. The Goodput column in Table III shows that enhancements in CARS goodput could reach up to more than 60% over Sample and up to more than 165% over AARF, which coincides with the enhancement in airtime load as described above.

To evaluate the CARS behavior in the presence of hidden station induced collisions, we use the above hotspot scenarios but with stationary vehicles located close to the server, to guarantee high quality links between vehicles and the server. Figure 13 shows the distribution of the number of packet transmitted over data rates. As shown, CARS and Sample schemes are robust to collisions and maintain the highest possible data rate, while AARF gets misled and switches to lower data rates with high collisions as number of vehicles increases. Although Sample performs better than AARF, Sample is not persistent as CARS in using the top data rates especially with the increase in number of vehicles. We also experimented with Vehicle-to-Infrastructure scenarios with downloading streams from the server to the vehicles. Results show similar behavior.

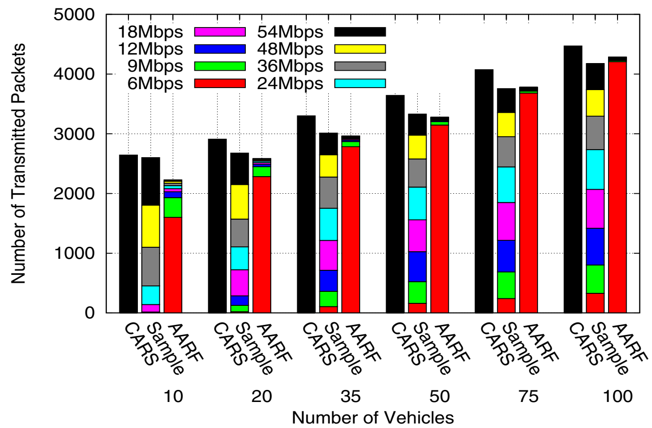


Fig. 13. Packet distributions with selected data rates for Hotspot uploading scenarios with stationary vehicles

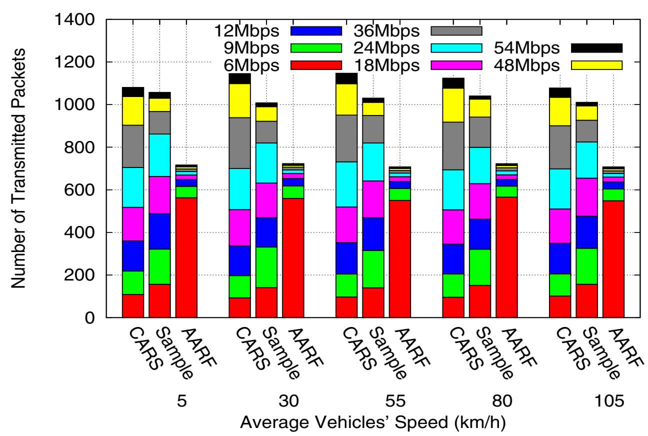


Fig. 14. Packet distributions with selected data rates for Highway scenarios

C. Vehicle-to-Vehicle Scenario

In this subsection, we describe the scenarios of Vehicle-to-Vehicle communication over a highway. Highway scenarios consist of a bi-directional highway of 10Km length, where each direction has the same number of vehicles. We set vehicles on one direction to be clients and vehicles on the other direction to be servers. On receiving a periodic update from a server, a client waits for a short random period with a mean value of 1.5 seconds before establishing a connection. The vehicle selects the lowest loaded server it hears from during this period. During the simulation, each client vehicle is limited to one connection only. We experimented with number of vehicles varying from 10 to 150 in each direction, and with vehicles' speed varying from 5Km/h to 105Km/h. Due to space limitation, we only show the distribution of transmitted packets with data rates in Figure 14 with different speeds values and 100 vehicles in each direction. As shown, CARS manages to use higher data rates for transmission more optimally than both Sample and AARF. For example, results show that CARS achieves better goodput than AARF up to 89%.

We also experimented with grid city scenarios, where we obtained similar results. We omit the results due to space constraints.

VIII. Discussion And Future Work

In what follows, we discuss some open issues and limitations of the CARS approach and present scope for future work.

A. Estimation Window Size Tuning

We performed preliminary experiments in different mobility scenarios, in which we tune the estimation window size to different values. Our initial results suggest that, in low mobility scenarios, existing estimation window based algorithms with very small estimation window size match the performance of CARS, but in high mobility scenarios, lowering the estimation window size deteriorates the performance. As future work, we are designing schemes to tune the estimation window size dynamically, using the context information.

B. Robustness To Errors in Context Information

During our experiments with GPS devices, we observed average localization errors of 5 meters with regular GPS, and 2 meters with differential GPS, which were small enough to not affect the performance of CARS. A more significant problem, however, is that of occasional GPS outage in tunnels, urban areas with tall buildings, etc. Our current implementation reduces the weight α when it detects a GPS outage, so that the rate selection is done purely using EWMA. We are studying the impact of context information update rate due to GPS outages on CARS performance.

IX. Conclusion

In this paper, we showed why existing rate adaptation algorithms underutilize the wireless link capacity in vehicular environments, and introduced Context Aware Rate Selection (CARS), a novel scheme that uses context information from the environment to perform *fast* rate adaptation in dynamic environments. Through extensive experiments in vehicular environments, we demonstrate that CARS consistently outperforms existing rate adaptation algorithms. We believe that our solution will inspire a new context aware approach to the problem of rate adaptation for traditional wireless networks as well.

References

[1] "Wireless on Wheels," 2005, MIT Technology Review.

[2] L. Armstrong, "Dedicated Short Range Communications(DSRC) home," http://www.leearmstrong.com/dsrc/dsrchomeset.htm.

[3] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," *Bell Labs Technical Journal*, 1997.

[4] "Onoe Rate Control," http://madwifi.org/browser/trunk/ath_rate/onoe.

[5] G. Holland, N. Vaidya, and V. Bahl, "A rate-adaptive MAC protocol for multihop wireless networks," in *Proceedings of Mobicom*, 2001.

[6] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: A practical approach," in *Proceedings of MSWiM*, 2004.

[7] J. C. Bicket, "Bit-rate selection in wireless networks," *Masters Thesis, MIT*, 2005.

[8] S. H. Wong, H. Yang, S. Lu, and V. Bhargavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of Mobicom*, 2006.

[9] T. Nadeem, S. Dashtinezhad, C. Liao, and L. Iftode, "TrafficView: A scalable traffic monitoring system," in *Proceedings of 2004 IEEE International Conference on Mobile Data Management (MDM)*, 2004.

[10] C. Robinson, L. Caminiti, D. Caveney, and K. Laberteaux, "Efficient coordination and transmission of data for vehicular safety applications," in *Proceedings of ACM VANET*, 2006.

[11] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs," in *Proceedings of IEEE INFOCOM*, 2006.

[12] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A practical snr-guided rate adaptation," in *In Proceedings of INFOCOM*, 2008.

[13] G. Judd, X. Wang, and P. Steenkiste, "Efficient channel-aware rate adaptation in dynamic environments," in *Proceedings of MobiSys*, June 2008.

[14] M. Loiacono, J. Rosca, and W. Trappe, "The Snowball Effect: Detailing Performance Anomalies of 802.11 Rate Adaptation," *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE*, pp. 5117–5122, 2007.

[15] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," in *Proceedings of ACM MobiCom*, September 2008.

[16] D. Hadaller, S. Keshav, and T. Brecht, "Mv-max: Improving wireless infrastructure access for multi-vehicular communication," in *ACM SIGCOMM Workshop on Challenged Networks (CHANTS-06)*, 2006.

[17] L. Wischhof, A. Ebner, H. Rohling, M. Lott, and R. Halfmann, "SOTIS - A Self-Organizing Traffic Information System," in *Proceedings of the 57th IEEE Vehicular Technology Conference*, 2003.

[18] R. Meier, B. Hughes, R. Cunningham, and V. Cahill, "Towards real-time middleware for applications of vehicular ad hoc networks," in *Proceedings of IFIP DAIS*, 2005.

[19] D. Dhoutaut, A. Regis, and F. Spies, "Impact of radio propagation models in vehicular ad hoc networks simulations," in *Proceedings of ACM VANET*, 2006.

[20] T. Rappaport, "Wireless communications: Principles and practice, 2nd edition," in *Prentice-Hall, Englewood Cliffs, NJ*, 2001.

[21] S. Gurunathan and K. Feher, "Multipath simulation models for mobile radio channels," in *IEEE 42nd Vehicular Technology Conference (VTC)*, 1992.

[22] "The Network Simulator ns-2," http://www.isi.edu/nsnam/ns/.