RFRA: Random Forests Rate Adaptation for Vehicular Networks

Oscar Puñal*, Hanzhi Zhang*, James Gross§
*UMIC Research Centre, RWTH Aachen University, Germany
§School of Electrical Engineering, Royal Institute of Technology, Sweden
punal@umic.rwth-aachen.de, james.gross@ee.kth.se

***Abstract*—Rate adaptation in vehicular networks is known to be more challenging than in WLANs due to the high mobility of stations. Nevertheless, vehicular networks are subject to certain recurring patterns particularly if stations communicate to roadside units. This has lead to the proposal of learning-based rate adaptation schemes which are trained for a certain propagation environment. In general, these schemes outperform other approaches at the price of being specific for a particular environment. In this paper we present RFRA, a novel rate adaptation scheme for vehicular networks. It is based on the machine-learning algorithm Random Forests which is known to be superior to most other learning approaches. Firstly, we show that RFRA outperforms other learning-based methods significantly. We also study the question how sensitive RFRA is to changes of the learned environment, especially with respect to the propagation characteristics. We show that, although this reduces the gain of our scheme, RFRA still provides a much higher performance than state-of-the-art rate adaptation schemes.**

I. Introduction

Over the last years vehicular ad-hoc networks (VANETs) have attracted interest from industry and academia. This is driven by demands for higher traffic safety [1], [2] and demonstrated by several standardization efforts such as IEEE 1609 WAVE [3] and IEEE 802.11p [4]. Although safety applications constitute the driving force in VANETs, support for multimedia applications is also expected [2]. For instance, users could initiate video downloads and large data transfers, or listen to online music directly from the car by exploiting the Internet access provided by roadside units (RSU). All these applications benefit in one way or the other from rate adaptation. Safety-critical messages can be conveyed reliably without wasting extensive air-time if higher rates are also sufficiently stable. On the other hand best-effort and multimedia applications benefit from rate adaptation due to faster download times, lower jitter, and a better medium utilization which is to the advantage of all users in the network.

The effectiveness of a rate adaptation scheme depends on its ability to accurately adapt to a *predicted* future state of the wireless channel. High order modulations achieve high data rates but have worse error rates than less efficient modulations. This trade-off is not trivial and adaptation algorithms are not standardized. In the context of 802.11 networks, intensive research activities have led to a vast number of rate adaptation schemes, see [5]–[8] and references therein.

The majority of the proposed schemes target at indoor propagation environments, which are characterized by slow channel changes. Vehicular communications, however, are subject to fast topology and channel changes, as well as short connectivity spans. Therefore, rate adaptation schemes for VANETs need to be robust to achieve an acceptable performance under such challenging conditions. Fortunately, vehicular communications feature certain characteristics that can be exploited to adapt the rate. Firstly, the movement of the nodes is limited in space by the length and width of the roads, and is regulated by traffic lights and speed limitations. Furthermore, moving cars do in general first approach a roadside unit and later leave the communication area, which results in a certain pattern of the signal strength over time. Finally, GPS devices provide valuable information, such as position, speed, and acceleration of a node.

These characteristics motivate the application of learning approaches to rate adaptation in vehicular networks. A few works contributed over the last years [9]–[11] show that indeed a learning approach can improve the performance of vehicular networks. However, these schemes have significant drawbacks. On the one hand, they rely on simple learning approaches. Better approaches, especially from machine learning, are known and leave room for higher performance. On the other hand, the schemes are not evaluated in circumstances different than the environment considered for training. Furthermore, the chosen approaches do not exploit the temporal evolution of the channel state to infer future states. This motivates us to study and propose Random Forest Rate Adaptation (RFRA), a novel rate adaptation scheme that employs Random Forests [12], a state-of-the-art machine learning algorithm, to learn the propagation environment and select the data rate for transmission. In addition, random Forests is known for its robustness against incomplete input data. In detail, our contributions are the following:

1) This work is, to our best knowledge, the first to apply Random Forests for rate adaptation. This algorithm is more accurate than the model-tree schemes in [10], [11] and the linear-regression model in [9]. Our approach exploits SNR samples obtained over a time window to characterize the propagation environment. We also use GPS information such as position and speed to increase the prediction accuracy. This design is significantly different and more robust than previous work.
2) We conduct extensive evaluations based on detailed

978-1-4673-5828-6/13/$31.00

simulations and show that RFRA outperforms state-of-the-art rate adaptation schemes (by 30% and more). This is even observed under quite different conditions than the ones present when learning the environment, as RFRA exhibits an extraordinary robustness to changes in the propagation conditions.

The remaining paper is structured as follows. In Section II we give an introduction to Random Forests. Section III provides a description of the design of RFRA as well as the training and prediction phases. Next, in Section IV we evaluate the accuracy of the chosen design and the impact of different parameterizations. In Section V we compare, by means of simulations, the performance of RFRA against other rate adaptation schemes. In Section VI we discuss implementation issues of RFRA. Related work is presented in Section VII and we conclude our work in Section VIII.

II. Introduction to Random Forests

Random Forests [12] is a learning-based heuristic that recognizes and exploits statistical dependencies in multi-dimensional decision problems. It belongs to the broader class of machine-learning algorithms and is known to be superior to most other machine-learning methods [12].

The operation of the algorithm is as follows. Assume we are given a set of J input variables $\mathbf{x} = \{x_j\}$ representing an instance of a decision problem. This set of variables is also referred to as *input feature vector* and we are interested in a corresponding output variable y. Random Forests inserts the input feature vector into N_t binary random decision trees (or random forest). Every tree has a depth of N_d and consists of one *root* node as well as *interior* and *leaf* nodes, see Figure 1. Root and interior nodes represent classification features of the decision problem and are composed of decision thresholds for a (randomly chosen) subset of the input variables. Hence, at each node of each tree different input variable subsets are considered. The leaf nodes represent the value of the output variable, i.e., the proposed solution to the decision problem, for the corresponding path through the decision tree. By pushing the current instance down each decision tree in the forest (that means, by traversing the tree according to the classification features), we obtain N_t output variables which are also called *votes*. Finally, the individual votes are aggregated into a single output variable.

Before this process can be applied, appropriate random decision trees need to be built. This requires a representative amount of *training data* consisting of I training examples. The i-th training example consists of input feature vector $\mathbf{x}^{(i)}$ and output $y^{(i)}$. To build a random decision tree, I training examples are randomly chosen with replacement. Hence, some training examples are considered multiple times while others are not considered at all. As a rule of thumb the latter approximately correspond to 30% of the total data and are also known as *out-of-the-bag* (oob) examples [12]. Then, for every node of the tree a random subset of input variables is considered. For the finally selected variable, decision thresholds need to be determined which lead either to traversing the tree at this node to the left or right. The thresholds for this splitting, as well as the input variable at the node, are carefully selected such that the so called *purity* of the subsequent child nodes is maximized with respect to the chosen training examples. Purity refers to the degree to which the resulting child nodes consist of cases with the same output variable value. Hence, an ideal threshold at a node would divide the training data into two distinct values for the output variable. For splitting the training data to maximize purity the CART algorithm [12] is a suitable choice (and the one used in this work).

The generation of a random forest provides two important quality measures. Firstly, there is the *prediction accuracy* that the newly created forest achieves. Either the data used for training (training accuracy) or newly collected data (test accuracy) is used to determine how accurately the forest predicts the actual outcome. This can be used as an indicator for the subsequent achievable performance without requiring further analysis or measurements, which substantially eases the design and evaluation process. Secondly, Random Forests provides a notion of the *importance* of the selected input variables. The methodology used to obtain the importance is briefly described in the following. Firstly, the oob examples are pushed down all trees in the forest and the number of correct votes is computed. Then, the values of the input variable $x_j^{(i)}$ under consideration are randomly permuted in the oob examples and these new examples are pushed down the trees. The resulting accuracy is compared with the original one and can be expressed by several different metrics, see [12].

Finally, Random Forests features several advantages which stem from the two randomization steps in the generation of the decision trees. First of all, for each tree only a subset of the training data is used for training, which prevents overfitting. Furthermore, due to the random selection of the subset of feature variables at each node of the tree, the method can cope with missing input variables [12].

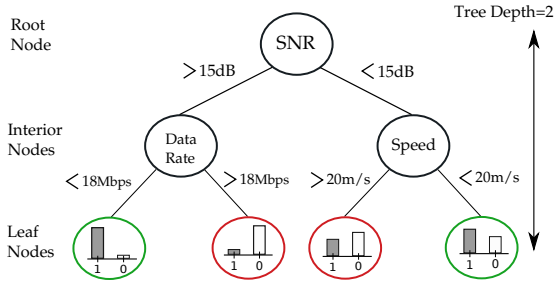


Fig. 1. Example of a random decision tree. In the example, the output variable is binary (either *failed* or *correct* transmission) and the tree has a depth N_d=2. The leaf nodes count the number of training examples that, having followed a certain path along the tree, result in a certain value of the output variable. The counting is represented as a histogram. Depending on the input variable at a node and the threshold considered for splitting, a different degree of purity is achieved at the leaf nodes.

III. Basic Design of RFRA for 802.11p Networks

The contribution of this work is the investigation of the suitability of machine-learning approaches in general – and Random Forests in detail – for the problem of rate adaptation in

vehicular networks. The intuition for this work is the following. Vehicular networking will be based, at least partially, on roadside units (RSUs) serving as access points. Such a RSU is installed at a fixed location and the movement patterns of cars associated to this RSU are potentially similar. The question arises if these patterns could be exploited together with the specifics of the propagation environment of the RSU. Essentially, this breaks down to predicting the signal-to-noise ratio γ (SNR) at the time of transmission depending on the propagation environment of the RSU as well as the mobility pattern and position of the corresponding station. We are interested in applying the method of Random Forests to this problem and benchmarking the corresponding performance against state-of-the-art rate adaptation approaches. Note that this implies a somewhat site-specific rate adaptation approach which may not be suitable for other sites. We call our scheme RFRA - *Random Forests Rate Adaptation* and consider IEEE 802.11p as underlying vehicular networking technology. In this section we provide background information about 802.11p and later discuss the design of RFRA.

A. IEEE 802.11p Background

IEEE 802.11p [4] adopts the physical layer from 802.11a with some modifications intended to increase the robustness of the signal to the highly dispersive vehicular environment, see Table I. Communication takes place in the 5.9 GHz band, where several channels are defined for network operation.[1] The medium access in 802.11p is governed by the CSMA/CA protocol. If a node wants to transmit a packet, it must first sense the medium idle for a certain time interval. The medium is idle if the detected energy is below the carrier sensing threshold. Every transmitted 802.11p packet consists of a preamble, PLCP, MAC and WSMP (WAVE Short Message Protocol) headers, and the payload. The PLCP header is transmitted using the most robust modulation and contains information about the data rate (i.e., modulation and coding) used for MAC and WSMP headers and payload. In 802.11p there are eight different data rates to choose from.

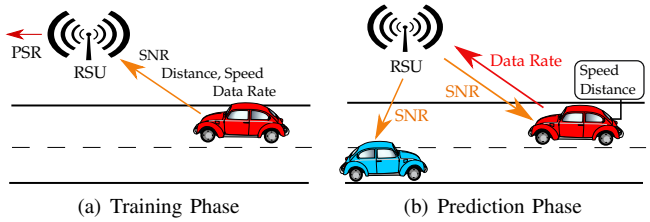
TABLE I. DIFFERENCES BETWEEN 802.11P AND 802.11A.

Parameter	**802.11p**	**802.11a**
Channel Bandwidth	10 MHz	20 MHz
Subcarrier Spacing	156.25 kHz	312.5 kHz
Data Rates	3 to 27 Mbps	6 to 54 Mbps
Slot/SIFS/DIFS time	13/32/58 μs	9/16/34 μs

B. RFRA Implementation Sketch

We consider in this paper the application of RFRA for uplink rate adaptation, i.e., for the transmission of packets from cars to the RSU. This is, however, not a limitation of the presented approach, as RFRA can be applied to downlink and bidirectional traffic as well. As input feature vector we select the variables SNR, current speed, current position, as well as data rate employed. The output of Random Forests is the prediction if the packet transmitted under the conditions described by the input variables will be received successfully by the RSU. As discussed in Section II, Random Forests requires an initial phase for collecting training data (training phase). Once this is completed, the generated random forest can be applied.

Figure 2(a) illustrates the training phase. The RSU keeps track of the transmission success of incoming packets together with the input variables (partially provided by the cars). The collected training data is later used to generate the random forest and is stored in form of a *configuration file*. This configuration file is provided to the cars which base their rate adaptation decision on it. Figure 2(b) exemplifies this prediction phase. Cars keep track of the input variables and predict, by using the configuration file, the success rate of a transmission for every available transmission rate.



(a) Training Phase (b) Prediction Phase

Fig. 2. Figure 2(a) illustrates the training phase carried out by the RSU. The collected training data is stored and later used by RFRA for learning. Figure 2(b) exemplifies the prediction phase carried out by a communicating car. The car combines SNR information (obtained from packets originated at the RSU) with speed and propagation distance to predict the success probability for each data rate and decide on the most appropriate rate for transmission.

C. Input and Output Variables

Input Feature Vector Earlier works [13], [14] have shown that SNR-based rate adaptation approaches are most suitable for fast changing channels as it is the case in vehicular communications. In this work we collect SNR samples per communication link over a certain time window, which we then group into several time slots. If multiple SNR samples are gathered over the period of one slot, the median of those samples is considered as the *representative SNR* value. By using the median, the SNR characterization is more robust against fast fluctuations of the channel and captures the mid-term channel state evolution better. The time slot length has been set to 5 ms based on [15], where the authors show that for urban environments the vehicular channel has a fairly constant behaviour within this time range. Note that if no packets are received over the period of one time slot, the rate decision can still be performed, as Random Forests can cope with missing input features. We set the total observation window to 100 ms, hence, RFRA considers a maximum of 20 SNR values for every data rate selection. It can be assumed that within this

[1]Note that in 802.11p there exists a predefined hopping pattern between a control channel and a service channel. This mandatory operation is not considered in the following, but we discuss the implications in Section VI.

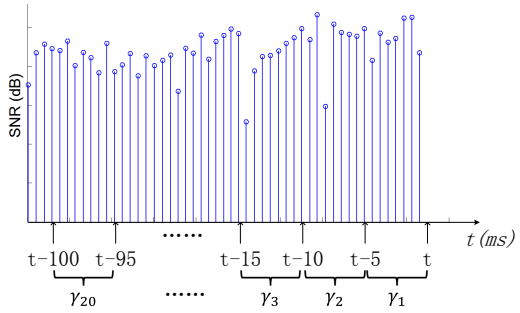


Fig. 3. SNR samples collected over a 100 ms window, which is divided into time slots of 5 ms each. If multiple samples are received during a time slot, the median of the samples (γ) is considered as the representative SNR.

time the speed and position of the cars remain constant. In fact, we show in Section IV that this time window results in a better performance than shorter or longer ones.

Further input variables for RFRA are the current GPS-position and the speed of the car. The current position is a good estimate of the propagation conditions between RSU and the car as, in general, short communication distances result in high signal strengths and vice-versa. It is, however, only a coarse indicator due to the variable signal propagation conditions created by multipath propagation and shadowing from other cars. Previous work [9], [10] considers the speed of a car as input variable as higher speeds result in faster channel changes. RFRA also accounts for the speed of the car, although this feature has a relatively low prediction importance as discussed in Section IV. Finally, RFRA takes the data rate of payload frames into consideration for predicting the success probability of the packet transmission[2].

Output Variable As response to a new input feature vector, every random tree predicts (with either one or zero[3]) the success of a transmission for every available data rate. The votes of all trees are aggregated to build a *Packet Success Rate* (PSR) which is defined as the number of trees voting for *one* divided by the total number of trees. Based on the predicted PSRs the most suitable data rate is selected. This final step is discussed further below.

D. Training Phase

It is known that the accuracy of Random Forests increases with the amount of training data [12]. We collect about two million training examples and use 60% for training and 40% for evaluating the accuracy. The data is obtained from ns3 simulations by letting one car move at different speeds within the coverage area of a RSU while constantly transmitting packets at different data rates. We assume that the car attaches position and speed information to every data packet[4]. Out of every incoming frame, the RSU generates a 23 elements input feature vector $\mathbf{x}^i$=($\gamma_1, \cdots, \gamma_{20}$, speed, position, data rate) and stores it to be later used by the learning algorithm. As the data is obtained with only one car, the training examples are free from packet errors due to collisions. This avoids inconsistencies in the data, which could lead to lower accuracies.

E. Prediction Phase

During communication, and before transmitting a new packet, the transmitter collects the required input variables. The different available values for the data rate are then iteratively pushed (together with all other input variables) down the random forest to obtain a PSR for each data rate. The authors in [9] argue that most SNR-based algorithms require a *build-up* phase to accurately predict the channel behaviour. However, RFRA does not wait until all SNR variables are collected as it can cope with missing input variables. It should be noted that the accuracy of Random Forests can be further improved by online training: With every new transmission the distribution of samples at the leaf nodes can be continuously updated. This is helpful in order to adapt to changes in the propagation environment (e.g., different attenuation caused by deciduous trees in winter and summer).

F. Rate Selection

The design of RFRA does not directly output a rate, but simply the PSR. Hence, to select a rate, a further processing step is required. We assume that for all available rates (or an appropriate subset) Random Forests is invoked. As we are interested in maximizing the goodput, the rate is afterwards selected according to a further criterion that reflects the goodput maximization. In principle, the goodput of the system can be computed in various different ways. Given the PSR per data rate, we have studied three such decision criteria:

- **Threshold:** Select highest data rate that predicts a PSR larger than a threshold $PSR > \theta$, where $\theta \in [0,1]$.
- **Raw Goodput:** Select data rate that maximizes Goodput $=$ Rate $\cdot$ PSR$^\theta$, where $\theta \in [0,\infty)$.
- **MAC Goodput:** Select data rate that maximizes Goodput $= f(Rate, PSR^\theta)$, where $\theta \in [0,\infty)$. The mapping function $f(Rate, PSR)$ is adopted from [16] and accounts for the impact of 802.11 protocol specialties, such as contention, packet collision, and inter-frame spacing.

IV. Design Parameter Evaluation

In this section we characterize the prediction accuracy of RFRA depending on the dimensionality of the random forest[5]. We further evaluate the impact on the goodput of the chosen random forest variables and other design parameters of RFRA.

[2]Note that we do not consider packets of variable size in this work, hence, the packet size is not part of the input feature vector.

[3]In this paper, every tree performs a binary decision about the success probability of a transmission (either failed or successful). One could, instead, exploit the information provided by the samples distribution at the leaf nodes (see Figure 1) to appropriately weight the decision of the trees.

[4]In a real 802.11p network deployment, this information is exchanged via beacon frames every 100 ms.

[5]We use the OpenCV libraries v2.4 to implement Random Forests.

	Training Accuracy		Test Accuracy	
	TP	TN	TP	TN
$N_d = 7,\ \ N_t = 30$	90.3	91.7	90.2	91.9
$N_d = 10,\ N_t = 30$	92.0	91.2	91.8	91.3
$\mathbf{N_d = 10,\ N_t = 50}$	92.9	90.9	92.8	91.0
$N_d = 10,\ N_t = 100$	93.3	90.6	93.2	90.7
$N_d = 10,\ N_t = 200$	93.1	90.8	93.0	90.9

Fig. 4. Accuracy of Random Forests for different dimensionalities.

A. Random Forests Dimensionality

The dimensionality of the random forest (tree depth N_d and number of decision trees N_t) has an impact on the learning and prediction accuracy of the algorithm. We characterize the best dimensionality as illustrated in Figure 4. The figure presents the true positive (TP) and true negative (TN) rates, where a high TP (or TN) rate indicates high accuracy by correctly predicting successful (or failed) transmissions. Training accuracy refers to the level of prediction obtained on the training data, while test accuracy refers to the prediction accuracy obtained on data not considered for learning. It can be observed that up to a certain dimension the accuracy slightly increases. However, further increasing the size of the random forest (e.g., $N_d = 10, N_t = 200$) can even deteriorate the performance due to over-fitting. As bigger trees are associated with further disadvantages, we select $N_d = 10$ and $N_t = 50$ in our experiments.

B. Variable Importance

Random Forests provides the prediction importance of the input features as quality measure, which we summarize in Table II. Due to space constraints we only show the values for the first ten SNR features (i.e., $\gamma_{i=1..10}$) and the *oldest* SNR feature (i.e., $\gamma_{i=20}$). As expected, the data rate has the highest relevance for predicting the PSR. The SNR samples collected over the last 25 ms (i.e., $\gamma_{i=1..5}$) are more important than the propagation distance. Interestingly, the speed is the input variable with the lowest relevance.

TABLE II. VARIABLE IMPORTANCE PREDICTED BY RANDOM FORESTS

Feature	Importance [%]	Feature	Importance [%]
γ_1	15.64	γ_8	2.53
γ_2	8.38	γ_9	2.43
γ_3	5.88	γ_{10}	2.27
γ_4	5.45	γ_{20}	1.42
γ_5	4.17	**Data Rate**	**28.83**
γ_6	3.35	Distance	3.84
γ_7	2.53	Speed	1.28

C. Goodput Parameter Study for RFRA

Next, we study different design aspects of RFRA by evaluating the impact in the achieved goodput. These evaluations are done by ns3 simulations. For instance, we study SNR collection windows of variable size and the contribution of GPS-based information, among others. For these evaluations we define a scenario where five cars move along a straight road (as in Figure 5) and transmit packets continuously to the RSU in saturation mode. That is, the cars have always a data packet buffered (with a fixed size of 500 Byte) ready to be transmitted to the RSU. For every scenario we perform 20 simulation runs with different seeds to experience different channel realizations. All our results show, together with the average performance, the 95% confidence intervals. Further parameters are summarized in Table III.

TABLE III. SIMULATION PARAMETERS.

Parameter	Value	Parameter	Value
Transmit power	40 mW	Carrier frequency f_c	5.2 GHz
Packet size	500 Byte	Background Doppler	50 Hz
Loss exponent α	3	Reference loss $\lvert H_{(d_0=1m)}\rvert^2$	46.67 dB
Noise power	-97 dBm	Energy detection threshold	-96 dBm
Shadowing deviation	8 dB	Carrier sense threshold	-96 dBm

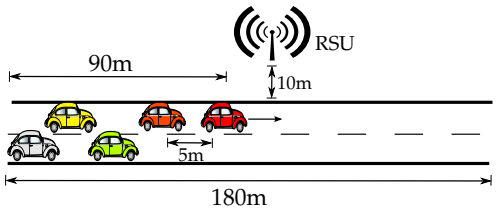
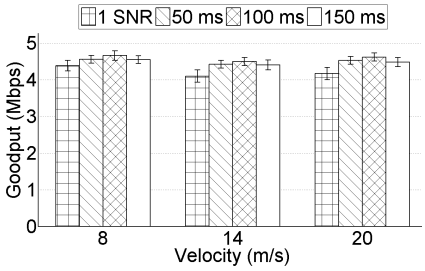


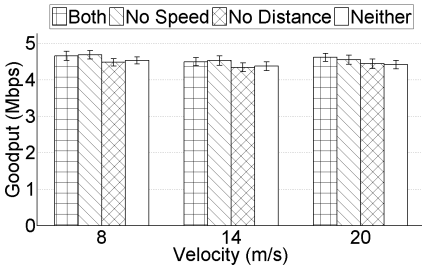
Fig. 5. Basic scenario sketch and topology details.

For the simulation of the wireless channel behavior, we consider the following setting. Path loss is obtained with the *log-distance* model $|H_{pl}|^2 = |H_{(d_0=1m)}|^2 + 10 \cdot \alpha \cdot \left(\frac{d}{d_0}\right)$, where d refers to the propagation distance (other variables are defined in Table III). Shadowing is based on the Gudmundson's model [17], which accounts for spatial correlation in urban environments and is characterized by a certain standard deviation. Small time-scale variations of the channel are modelled as time-correlated Rayleigh fading with a Doppler shift given by $f_d = v \cdot \frac{f_c}{c}$, where v corresponds to the speed, f_c to the carrier frequency and c to the speed of light. Given an SNR, a modulation and coding scheme, and a packet length, the error model in ns3 returns an expected packet error rate. The effects of convolutional coding are considered as described in [18].

Figures 6 and 7 highlight that different parameterizations of RFRA result in a variable goodput performance. However, the differences are only small and, in most cases, within the confidence intervals, which demonstrates the robustness of RFRA. In Figure 6(a) we build a new random forest for every different time window size. On the other hand, in Figure 6(b) we use the default random forest that accounts for both GPS inputs and evaluate the resulting performance when a certain variable (or both) is missing. It should be noted that in the evaluation in Section V we parameterize RFRA so as to achieve the best performance.

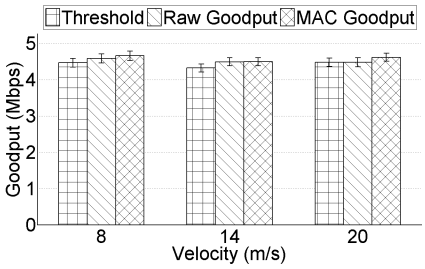


(a) Impact of SNR window length.

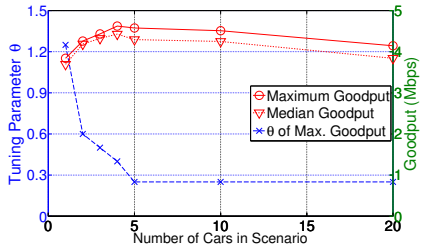


(b) Importance of GPS variables.

Fig. 6. Figure 6(a) shows that different window sizes for collecting SNR samples result in slightly different goodput. If only the most recent SNR sample is used (approach considered in most related works), the performance is significantly reduced. Figure 6(b) shows that speed and propagation distance provide a modest improvement. In addition, the distance is more important than the speed and the latter is only helpful in highly mobile environments.



(a) Different rate selection criteria.



(b) Best θ for the *MAC Goodput* criterion.

Fig. 7. Different methods for selecting the data rate based on the predicted PSR result in different goodput performance (Fig. 7(a)). Accounting for the MAC protocol issues provides a modest benefit. Figure 7(b) shows that the best parameterization θ for the MAC Goodput criterion depends on the number of active users. A lower θ leads to higher rate selections, which reduces the network congestion. We tested different values for θ (between 0.1 and 1.5) and show the maximum (and median) goodput achieved by the set of considered values for a varying number of vehicles.

V. Performance Evaluation

After presenting the design of RFRA, we study the performance of the scheme in comparison to other rate adaptation algorithms. We are also interested in the robustness of RFRA especially in cases where the propagation environment changes with respect to the training data. We study these issues in two typical vehicular scenarios. The first one is a straight road, while the second one models a urban grid with a cross-road. All results are obtained by simulations.

A. Comparison Schemes and Metrics

We consider the following comparison schemes:

- **AARF:** The approach presented in [19] adapts the rate based on packet success statistics and dynamic decision thresholds.
- **CARA:** The authors in [6] combine ARF with the RTS/CTS exchange to differentiate packet losses due to collisions from losses due to channel errors.
- **SNR:** This scheme assumes perfect channel knowledge at the transmitter (obtained from ACK frames). Assuming that the current channel state holds for the next transmission, the best rate is selected accordingly. This SNR scheme is used in [10] to benchmark MTRA, a decision tree-based scheme. In [10] MTRA and SNR achieve a comparable performance, and as not all implementation details of MTRA are available, we consider the SNR scheme as a good approximation for the former[6].

For evaluating the schemes described above we rely on the following performance metrics. **Goodput** is defined as rate (in bits/second) of correctly received data frames excluding headers. Unless differently specified, we show the goodput measured by the RSU aggregated over all transmitting nodes. **Packet Error Rate (PER)** is the ratio of erroneous data packets to the total data packets transmitted. **Data Rate** refers to the average data rate (in bits/second) used by the transmitting nodes during a simulation run. **Accuracy** represents if the data rate has been precisely selected. To evaluate this, we obtain (off-line) the performance of every data rate as function of the SNR. For every SNR point there is always a rate that maximizes the goodput. At runtime, for every received packet we obtain both its SNR and data rate, and decide if the rate is optimal, over-, or under-selected.

B. Straight Road Scenarios

The first scenario is characterized by a straight road and therefore simple movement patterns. In all our evaluations we assume that the RSU has finished learning the training data. Furthermore, we assume that the cars have already downloaded the configuration file and that the latter, and not the RSU, perform the rate adaptation.

[6]In the evaluation we have also studied the performance of OFRA [8]. Due to space constraints we do not show the results, however, the goodput achieved by SNR and OFRA was comparable.

a) Perfectly Trained Evaluation:
We start evaluating the different schemes in a scenario that matches the conditions used for training, however, we consider five cars instead of a single one. In the scenario, the cars move from the left to the right end of a road at the same constant speed, see Figure 5. We find that training with multiple cars slightly reduces the performance of RFRA, as packet collisions lead to lower throughput (even under good channel conditions) and, hence, distort the training data. All other simulation parameters are the same as in Section IV-C

The goodput achieved by RFRA is the highest among all schemes (Figure 9(a)) independent from the speed at which cars are moving. The performance gain is on average 80% higher than AARF, 24% higher than CARA, and 20% higher than SNR. Furthermore, the goodput gain of RFRA increases at high speeds (20 m/s). This can be explained with the help of Figure 8, where RFRA exhibits the most balanced rate selection accuracy. It achieves a good performance even at high speeds, where almost no deterioration in accuracy is observed. In contrast, the SNR algorithm suffers from a very high error rate of about 35% due to an aggressive rate selection (Figures 9(b) and 9(c)). It can be further observed that the error rate of RFRA increases only slightly with the speed, which is not the case for the SNR scheme (higher over-selection). The error rates of AARF and CARA are noticeably lower as these schemes tend to under-select the data rate, which in mobile environments results in a reliable communication yet a poor goodput performance.

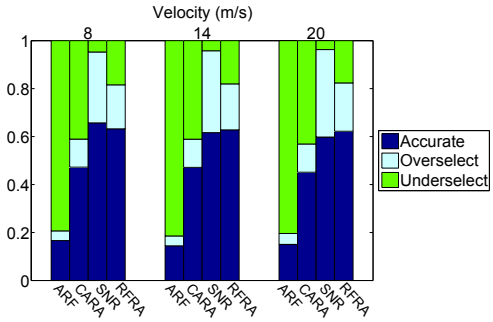
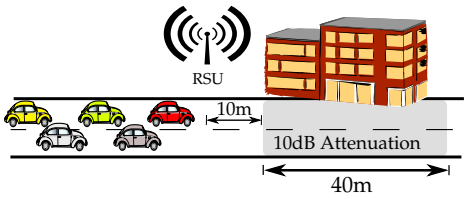


Fig. 8. Prediction accuracy comparison.

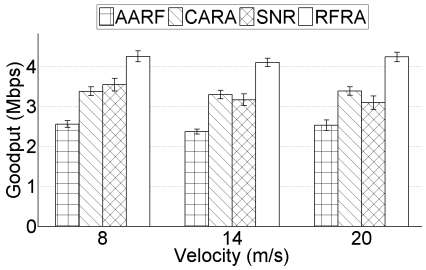
b) Changing Propagation Conditions:
Next we consider two slight variations to the above scenario.

Big Object Attenuation: First, we introduce a constant attenuation of 10 dB over a road segment of 40 m, see Figure 10(a). Figure 10(b) shows that the goodput of all schemes decreases compared to the basic scenario due to the larger attenuation. However, RFRA still outperforms all other schemes with a 25% gain over the SNR scheme.

Larger Path Loss Attenuation: Next, we change the path loss exponent to $\alpha = 3.5$. We also add more cars to the scenario (having now 10 cars). We split them in two groups of 5 cars, where each group starts at one end of the road. Both groups meet in the proximity of the RSU. Note that when the two



(a) Scenario sketch: Attenuation caused by building.



(b) Goodput performance.

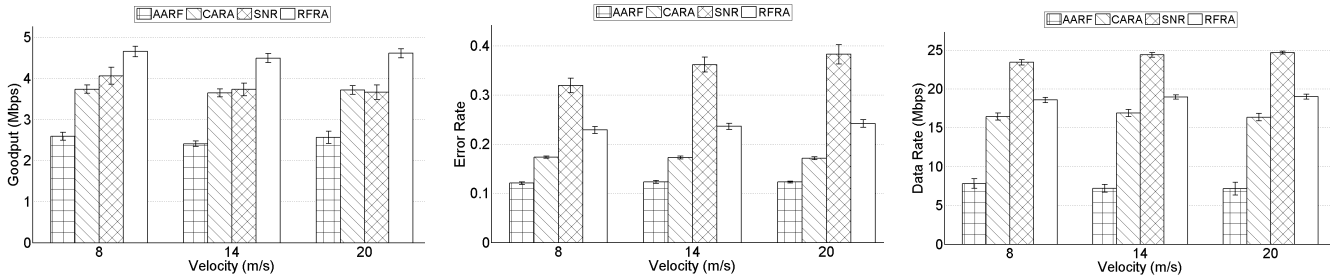
Fig. 10. Changing the propagation conditions of the basic scenario (5 cars) by adding a large attenuation area caused by a building.

groups are at the ends of the road they are sometimes out of each other's hearing range (depending on the shadowing and multipath fading realizations). Hence, packet collisions due to hidden nodes may happen. Figure 11(a) shows that the goodput of all schemes is further reduced due to the increased attenuation. The performance obtained by the CARA algorithm comes close SNR's. Note that CARA transmits RTS/CTS frames to protect against hidden nodes, which is beneficial in this case. Still, RFRA outperforms all schemes (on average 11% and 17% more goodput than CARA and SNR, respectively) and achieves a good performance even in scenarios that differ significantly from the basic training data.

However, if every scenario is characterized by a specific random forest, the performance gain of RFRA is expected to increase. To confirm that, we train RFRA in this new scenario and show the resulting performance in Figure 11(b). Interestingly, there is only a slight improvement for low speeds and a noticeable one, of about 10%, obtained at larger speeds.

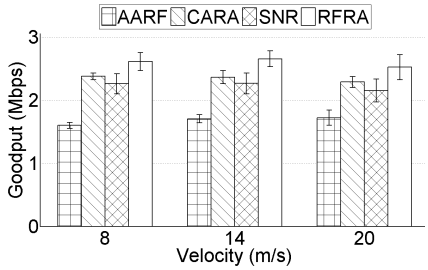
C. Urban Scenario with Cross-Road: Manhattan Grid

Nex,t we consider a urban scenario with many cars. Specifically, we select the Manhattan Grid scenario, which is the most common vehicular scenario in literature [20]. We use OpenStreetMap [21] to model a cross-road and the contiguous four blocks of the area of Manhattan (Figure 12). The roadside unit is located in the center of the cross-road. Along the vertical and horizontal streets (300 m long) with origin at the cross-road, line-of-sight (LOS) communication is possible. For the other streets (shaded area) we add a fixed attenuation of 10 dB caused by buildings. All streets consist of two lanes with a single driving direction. We place 70 cars in the scenario and use the SUMO mobility tool [22] to model acceleration (3.5 m/s^2), deceleration (4.0 m/s^2), length of a car

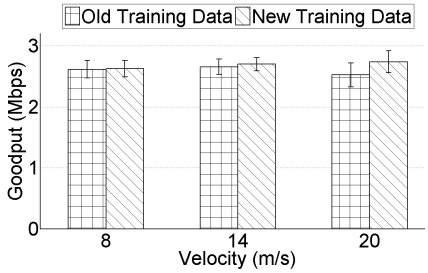


(a) Goodput (b) Error Rate (c) Average Data Rate

Fig. 9. Performance comparison for the 5 cars scenario, where RFRA provides the highest goodput.



(a) Goodput performance: Path loss exponent $\alpha = 3.5$ and 10 cars.



(b) Performance gain from specific training in this scenario.

Fig. 11. Changing the propagation conditions of the basic scenario by increasing the path loss exponent and including more cars (10) in the scenario.

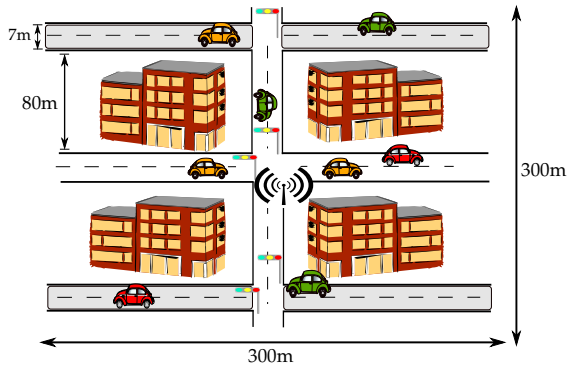


Fig. 12. Manhattan grid: Scenario details.

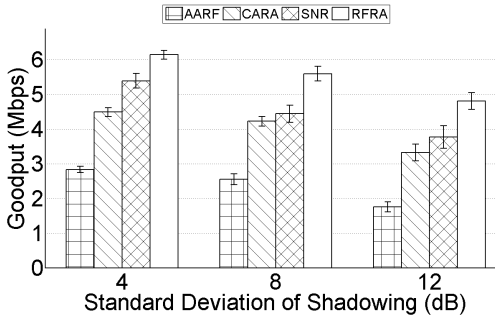


Fig. 13. Goodput results for the Manhattan Grid scenario.

(4.0 m), minimum distance between cars (3.0 m), and maximal speed of the cars (19.50 m/s). The movement of the cars depends on the speed limits and the traffic lights located at the cross-roads. It should be noted that not all cars are within hearing range of the RSU at all times. On average, the RSU supports 20 cars simultaneously. Unless differently specified, the channel modelling and other simulation parameters used are summarized in Table III. Figure 13 shows the goodput comparison under different shadowing conditions (4, 8, and 12 dB standard deviation). Note that RFRA uses a scenario-specific training for 8 dB shadowing variance.

It can be observed that RFRA clearly outperforms all other rate adaptation schemes. Again this occurs also under different propagation conditions than the ones used for training. RFRA achieves up to 27%, 35%, and 170% more goodput than SNR, CARA, and AARF, respectively.

VI. Implementation Aspects

Clearly, learning the environment specifics for rate adaptation in vehicular networks brings performance gains. The downside to this approach is that for each RSU potentially a different random forest is required. Furthermore, the question arises how fast the rate can be computed.

A. Random Forests Configuration File

As shown in Table IV, a random forest dimensioned with $N_d = 10$ and $N_t = 50$ has a total data size of 6.6 MByte. Basically, the configuration file is a text file, hence, compression schemes reduce its size significantly. The compressed file could be downloaded when associating to an access point or perhaps

prior to a trip. In the case of regular driving routes (e.g., from home to workplace) the overhead is negligible.

TABLE IV. RANDOM FORESTS CONFIGURATION FILE SIZE.

Dimensionality (N_d, N_t)	**File Size [MB]**	**Compressed Size [MB]**
(10, 100)	13.40	0.68
(10, 50)	6.60	0.34
(10, 30)	3.90	0.19

B. Computational Complexity

In addition to considering the file size, the execution time is important. We measured the time for performing a PSR prediction. For this task we used a computer featuring a 3 GHz AMD Phenom processor and 8 GByte RAM memory. The average time required for predicting the PSR of a single data rate is 12.62 μs. Hence, predicting the PSR for all 8 data rates results in an average time of about 100 μs. This time is too high for a practical implementation, as in the worst case only 58 μs (DIFS time in 802.11p) are available to select the rate before transmitting. Nevertheless, the required speed-up can be obtained by reducing the search space such that only few data rates are considered. In addition, CPU parallelization can provide a further speed-up.

C. Channel Hopping between CCH and SCH

As mentioned in Section III-A, in 802.11p the medium is accessed in a FDMA/TDMA fashion, where a station switches between control channel (CCH) and a service channel (SCH) every 50 ms. The CCH is used to advertise services offered on a SCH and to broadcast safety messages. This has several implications to our proposed scheme: Firstly, during the CCH interval the number of gathered SNR samples would be reduced, as a low rate of transmitted packets is needed to not hinder the timeliness of safety messages. Although RFRA can cope with missing input variables, this would reduce the prediction accuracy. Alternatively, as proposed in [23], the SCH interval can be extended to improve the throughput. In addition, if cars feature dual-radio capabilities, 802.11p could be used for traffic security, while 802.11a/n could support multimedia applications. In both cases, we expect that RFRA improves the performance of the vehicular network.

VII. RELATED WORK

Rate adaptation for IEEE 802.11 networks has been extensively considered over the last years [6]–[10], [13], [14]. Most work considers indoor scenarios where channel conditions and network topology remain stable for long time spans. In typical vehicular networks, these conditions are naturally quite different: Channel states vary fast as well as network associations. Under these conditions, rate adaptation approaches based on packet success statistics [6], [19] are known to have serious problems to converge to the optimal rate [14] and are outperformed by SNR-based schemes.

A. SNR-Based Rate Adaptation

SNR-based rate adaptation can be either sender or receiver-based. In the sender-based case [5], [7], the transmitter computes the SNR from previously received frames assuming channel reciprocity. Afterwards, the data rate selection is performed. In receiver-based rate adaptation [13], [14], the receiver maps the SNR of received packets to a specific data rate and informs the sender about which rate to use. This can be done by modifying CTS or ACK packets [13] or using a customized feedback packet [8], among others.

In general, the authors in [13] reveal that SNR-based rate selection approaches tend to over-select the data rate. They further propose to empirically characterize (or *train*) the propagation conditions and obtain an accurate mapping between data rate and performance.

B. Rate Adaptation for Vehicular Networks

There have been a few works proposing rate adaptation for vehicular networks [9]–[11], [13], [24].

In [9] the authors propose the CARS algorithm. Based on empirically collected data, a linear-regression model is built that combines context information (speed and transmission distance) and packet success statistics to predict the packet error rate. Depending on the relative speed of two communicating nodes, the algorithm gives more importance to the GPS information or to the packet statistics. By means of simulations the authors show that CARS outperforms the AARF algorithm substantially. Furthermore, the proposed method is simple and can be easily implemented in commodity hardware. However, the algorithm relies at high speeds practically only on speed and distance as predictors for the error performance. These variables are likely to work well only in similar environments to the one used for collecting training data. As soon as the distance does not correlate as expected with the signal strength, the algorithm is prone to over- or underestimate the data rate. We have adapted and parameterized our simulator to accurately reproduce the vehicle-to-infrastructure scenario described in [9] (i.e., straight road scenario with 10 cars transmitting 1000 Byte long packets to the RSU at a rate of 100 packets/s). In the 10 cars case (see Table III in [9]), the CARS algorithm achieves about 80% more goodput than AARF. In the same scenario we compare RFRA with AARF and obtain an average goodput gain of about 290%, hence, RFRA is expected to outperform CARS significantly.

The authors in [10] propose a model-tree-based rate adaptation (MTRA) which is a different approach from machine learning. They build a M5 decision tree [25] to model the training data consisting of {distance, SNR, speed, data rate} input feature vectors. The decision tree is used to predict the error rate for different data rates and the highest data rate that yields an error rate below a predefined threshold is selected. The MTRA algorithm considers only the most recent SNR value for predicting the rate. In our work we show that, in fast changing channel conditions, accounting only for the most recent SNR sample results in a poor performance and propose a broader time window for collecting SNR samples. Furthermore, from machine learning it is known that Random

Forests is in general a superior approach to M5 decision trees. As discussed in Section V, we have implicitly shown that RFRA outperforms MTRA by up 25%.

Recently, the authors in [11] present a rate adaptation scheme based on decision tree, where SNR, speed, and *channel type* are the input variables. Channel type coarsely defines the propagation characteristics of the environment (e.g., pedestrian or vehicular channel) and it is significantly more important than SNR and speed for performing accurate rate selections [11]. However, in practice it is difficult to determine the characteristics of the channel (i.e., determine the channel type) without requiring extensive measurements campaigns or the use of complex and expensive devices. In addition, the channel characteristics are implicitly contained in the SNR measurements that we propose. Firstly, the impact of speed (i.e., channel variability in the time domain) is captured by the SNR evolution over the time window. Secondly, the impact of multipath dispersion (i.e., the channel variability in the frequency domain) leads to varying error rates for a given (average) SNR [26], which can be learned as well.

VIII. Conclusion

In this paper we have presented RFRA, a novel rate adaptation scheme for vehicular networks. RFRA collects training data and learns, using the Random Forests algorithm, how the communication performance varies as function of certain input features for a given data rate. In particular, SNR samples are collected over a time window together with the speed and the position of a car. Depending on these input values, RFRA predicts the success probability of the transmission and uses this information to select the data rate. By means of simulations we have shown that RFRA is suitable for vehicular communications as it outperforms related work, especially at high speeds, even if the propagation conditions differ from the ones present while collecting the training data. As future work we plan to implement RFRA on commodity hardware as preliminary analysis anticipates only a moderate computational complexity.

IX. Acknowledgements

This research was funded by the DFG Cluster of Excellence on Ultra High-Speed Mobile Information and Communication (UMIC), German Research Foundation grant DFG EXC 89. We would like to thank Georgios Floros for the extensive and fruitful discussions on machine learning and Moritz Werner for the help with the evaluation.

References

[1] E. Schoch, F. Kargl, M. Weber, and T. Leinmüller, "Communication Patterns in VANETs," *IEEE Communications Magazine*, 2008.

[2] Vehicle Safety Communication Project, "Task 3 Final Report: Identify Intelligent Vehicle Safety Applications," U.S. Department of Transportation, Tech. Rep., March 2005.

[3] *IEEE Standard for Wireless Access in Vehicular Environments (WAVE)–Multi-Channel Operation*, IEEE IEEE: 1609.4, July 2011.

[4] *IEEE 802.11p – Wireless Access in Vehicular Environments, Amendment 6 to 802.11*, IEEE IEEE: 802.11p, July 2010.

[5] G. Judd, X. Wang, and P. Steenkiste, "Efficient Channel-Aware Rate Adaptation in Dynamic Environments," in *Proc. of MobiSys*, 2008.

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

[7] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, "A Practical SNR-Guided Rate Adaptation," in *Proc. INFOCOM*, 2008.

[8] F. Schmidt, A. Hithnawi, O. Puñal, J. Gross, and K. Wehrle, "A Receiver-Based 802.11 Rate Adaptation Scheme with On-Demand Feedback," in *Proc. PIMRC*, 2012.

[9] P. Shankar, T. Nadeem, J. Rosca, and L. Iftode, "CARS : Context-Aware Rate Selection for Vehicular Networks," in *Proc. ICNP*, 2008.

[10] Q. Xia, J. Pu, and M. Hamdi, "Model-Tree-based Rate Adaptation Scheme for Vehicular Networks," in *Proc. ICC*, 2009.

[11] J. He, H. Liu, P. Cui, J. Landon, O. Altintas, R. Vuyyuru, D. Rajan, and J. Camp, "Design and Experimental Evaluation of Context-Aware Link-Level Adaptation," in *Proc. INFOCOM*, 2012.

[12] L. Breiman, "Random Forests," Tech. Rep., 2001.

[13] J. Camp and E. Knightly, "Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-Layer Implementation and Experimental Evaluation," in *IEEE/ACM Transactions on Networking*, 2010.

[14] X. Chen, P. Gangwal, and D. Qiao, "Practical Rate Adaptation in Mobile Environments," in *Proc. PERCOM*, 2009.

[15] L. Bernadó, T. Zemen, A. Paier, G. Matz, J. Karedal, N. Czink, C. Dumard, F. Tufvesson, M. Hagenauer, A. Molisch, and C. F. Mecklenbrauker, "Non-WSSUS Vehicular Channel Characterization at 5.2 GHz - Spectral Divergence and Time-Variant Coherence Parameters," *URSI General Assembly*, 2008.

[16] D. Qiao, S. Choi, A. Jain, and K. G. Shin, "MiSer: An Optimal Low-Energy Transmission Strategy for IEEE 802.11a/h," in *Proc. MobiCom*, 2003.

[17] M. Gudmundson, "Correlation Model for Shadow Fading in Mobile Radio Systems," *Electronics Letters*, 1991.

[18] M. Lacage and T. R. Henderson, "Yet Another Network Simulator," in *Proc. of WNS2*, 2006.

[19] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *Proc. MSWIM*, 2004.

[20] S. Joerer, F. Dressler, and C. Sommer, "Comparing Apples and Oranges?: Trends in IVC Simulations," in *Proc. of VANET*, 2012.

[21] E. Wolf, G. Matthews, K. McNinch, and B. Poore, "OpenStreetMap Collaborative Prototype, Phase 1: U.S. Geological Survey Open-File Report," Tech. Rep., 2011.

[22] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "SUMO: Simulation of Urban Mobility: An Overview," in *Proc. SIMUL*, 2011.

[23] S. Y. Wang, C. L. Chou, K. C. Liu, T. W. Ho, W. J. Hung, C. F. Huang, M. S. Hsu, H. Y. Chen, and C. C. Lin, "Improving the Channel Utilization of IEEE 802.11p/1609 Networks," in *Proc. WCNC*, 2009.

[24] P. Deshpande and S. R. Das, "BRAVE: Bit-Rate Adaptation in Vehicular Environments," in *Proc. VANET*, 2012.

[25] J. R. Quinlan, "Learning With Continuous Classes," in *Proc. of AUS-AI*, 1992.

[26] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 Packet Delivery from Wireless Channel Measurements," *SIGCOMM Comput. Commun. Rev.*, 2010.