Automated Decision System to Exploit Network Diversity for Connected Vehicles

Felipe Valle, Sandra Céspedes, *Senior Member, IEEE,* and Abdelhakim Hafid, *Member, IEEE*

***Abstract*—In this work, we introduce a methodology that takes advantage of the inherent network diversity present in vehicular communications to improve the performance of safety applications. This methodology is based on a framework that simultaneously exploits the strengths of each individual network by using a set of decision rules. The implementation begins with a manual approach in which a typical, hierarchical decision tree characterizes the decision process of a single application when sending data to other users in the network. Analytical and simulation results validate the decision system approach when diversity is exploited as demonstrated by a boost in application performance, achieving an average latency under 100 ms and a 40% increase in throughput due to the increased packet delivery ratio. We then apply an ensemble learning technique, Random Forests (RF), to automatically reproduce the performance of the manually built tree system. Simulations under realistic traffic scenarios show the RF approach can replicate manually-built tree performance with up to 98% precision. A comparison with another state-of-the-art hybrid method also shows the RF scheme improves performance under a different application scenario without additional manual adjustments. With our methodology, we can add different application requirements and network characteristics to obtain a fully automated and adaptable decision system to optimize vehicular safety applications.**

***Index Terms*—Connected Vehicles, Decision system, Decision tree learning, Heterogeneous networks, Random Forest.**

I. Introduction

THE large investments that government, academia, and industry sectors have dedicated to the evolution of Intelligent Transportation Systems (ITS) have led to the development of safety and traffic management applications ready to be deployed in vehicles and road infrastructure. Several wireless access technologies have been developed to support the communication flows in vehicular networks given the growing information exchange expected among vehicles (V2V communications) and between vehicles and the infrastructure (V2I communications). Among the available technologies, cellular 4G/5G and IEEE 802.11-OCB (formerly known as IEEE 802.11p) are front-runner candidates, and both are considered well suited for providing ITS services [1], [2].

Nevertheless, the high mobility of vehicles and the dynamic topology changes of the vehicular communications networks make it difficult to provide satisfactory ITS services only through a single wireless network. In fact, it is widely accepted that the supporting infrastructure and communications technologies for vehicular networks will be heterogeneous in nature, hence providing for network diversity [3], [4]. Therefore, future vehicular networks should consider systems designed to exploit the multiple access technologies in what is called a Heterogeneous Vehicular Network (HetVNET). This network model is illustrated in Fig. 1.

It must also be noted that both IEEE 802.11-OCB and mobile cellular networks have their own limitations when used in vehicular environments. In particular, IEEE 802.11-OCB was mainly designed for short-range communications without the need of pervasive roadside infrastructure, but it can hardly provide reliable connectivity between vehicles as the network density increases [1], [5]. On the other hand, although mobile cellular networks can provide wide geographical coverage, they cannot efficiently support real-time information exchange for local areas [6].

It is important to mention an emerging technology, called C-V2X [7], that is currently challenging 802.11-OCB for supremacy in the field of V2X communications. C-V2X refers to the family of cellular technologies designed for automotive applications and standardized by 3GPP. It is, in essence, an LTE variant defined in REL14 [8] of the standard that added direct car-to-car capability. The main issue between C-V2X and 802.11-OCB is that, while both technologies use the same spectrum, they may not be inter-operable. Because they both use different physical layers and MAC protocols, their coexistence could potentially result in harmful co-channel interference issues. Therefore, our contribution focuses on studying the interworking of 802.11-OCB and standard LTE given the already proven compatibility between the two technologies and the potential for boosting the performance of critical safety applications both in terms of throughput and latency.

It is clear that each access network has both advantages and disadvantages. In the long run, this should not be a race among the different options; instead, multiple options will need to combine for a robust communications system to operate within a heterogeneous infrastructure. In this work, we leverage multiple options by creating an intelligent framework that integrates a set of *decision rules*. The approach allows data packets to flow through the network with the most favorable conditions in terms of throughput and delay. On the applications side, we focus on a single category, i.e., cooperative awareness applications, since they share similar reliability and latency requirements. More specifically, depending on the application requirements, the control and signaling flows of the cooperative awareness application may, for example, travel via IEEE 802.11-OCB, while the data flow may be transmitted

F. Valle is with the Department of Electrical Engineering, Universidad de Chile, Santiago, Chile.
S. Céspedes is with the Department of Electrical Engineering, Universidad de Chile and with the NIC Labs, Universidad de Chile, Santiago, Chile.
A. Hafid is with the Department of Computer Science and Operational Research, University of Montreal, Montreal, Canada.

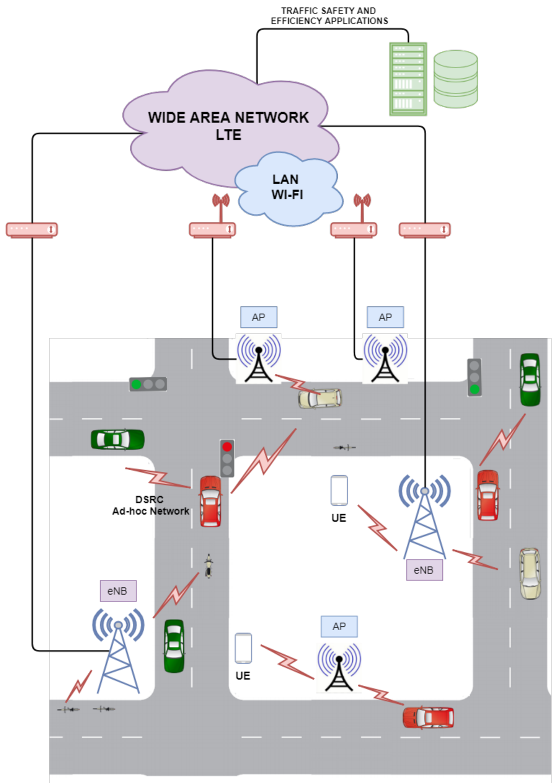


Fig. 1: Heterogeneous Vehicular Network Environment

over the cellular infrastructure.

Furthermore, we demonstrate that the decision tree approach helps to solve the task at hand by improving the applications' performance. However, many different decision trees would have to be designed and implemented to obtain a truly robust and generalized decision system. Thus, in this work, we also *automate* the system and demonstrate that a more robust and *generalized* decision system can be obtained with the proposed automation. Finally, we show through comparative analysis that the decision system approach can outperform a state-of-the-art hybrid architecture in regards to both the network latency and the packet delivery ratio (PDR).

To the best knowledge of the authors, the proposed decision system approach is the first to attempt a solution to the network selection problem for each individual user at the moment of message generation without the need to perform excessive computations or execute complex algorithms (e.g. clustering and bio-inspired mechanisms). The decision system considerably increases the probability of meeting the strict requirements of safety applications. Furthermore, our approach has the added advantages of flexibility and the potential to integrate with other state-of-the-art hybrid schemes.

The remainder of this paper is organized as follows. In Section II, we show a review of the related work. In Section III, we introduce our decision system framework and analyze its performance via analytical models and simulations. Section IV deals with the automation of the decision system and introduces the Random Forest algorithm. A comparative study based on simulation experiments is discussed in Section V. Finally, Section VI concludes this paper with a summary of our main results and findings.

II. Related Work

Initially, research on HetVNETs concentrated on demonstrating how the integration of a particular network to an existing one (either cellular into vehicular or vice versa) can be used to improve the performance of the joint network for a particular application or scenario. For example, the authors in [9] provide an analytical study in which they quantify and evaluate how much vehicular ad-hoc networks (VANETs) can offload from the cellular infrastructure while considering the constraints related to the capacity and stability of vehicular links, the infrastructure features, and the quality of Service (QoS) flow constraints. Meanwhile, the use of V2V communications to partially relieve the cellular infrastructure from Floating Car Data traffic is explored in [10].

More recently, the focus has turned into the development of hybrid architectures, i.e., multi-network architectures, whose distinguishing feature is usually their proposed *network selection* scheme. For example, Li et al. [11] developed a cellular-VANET heterogeneous network architecture to disseminate data more efficiently. They introduce a cooperative protocol based on coalition game theory that combines both networks to improve the dissemination of safety messages. Meanwhile, Zhu et al. [12] approach to network selection in heterogeneous networks is based on information theory. They model the selection problem as a Bayesian game with incomplete information under the assumption that each user has only partial information regarding the preferences of other users.

Another discerning aspect to consider between hybrid architectures is whether or not they take into account the online network status. Some schemes only consider basic static information about the available networks [9]–[11]. However, more complex schemes integrate online information about the network state (e.g., bandwidth allocation and channel congestion) to solve the problem of network selection [12]–[14]. For example, Ucar et al. [13] propose a IEEE 802.11p and LTE hybrid architecture for message dissemination. They combine clustering of vehicles and the cellular architecture with the goal of achieving a high PDR and low latency, while keeping usage of the cellular infrastructure at a minimum level. Tian et al. [14] propose a bio-inspired network selection solution designed to guarantee the QoS of mobile users as well as efficient utilization and fair allocation of network resources. Their solution is based on an Attractor Selection Model (ASM), which is used to describe the self-adaptive response of a cellular gene network to varying environmental conditions.

However, one common issue that many approaches have, especially those focused on the use of more abstract mathematical tools (e.g game theory, information theory, and optimization) [9]–[12], is that they usually measure performance in terms of cost/utility functions and probability distributions. While this is useful to demonstrate their respective advantages, it does not allow us to visualize their impact in terms of more concrete quantities, such as PDR and latency.

Furthermore, most schemes report achieving low latency as a goal but do not specifically target (much less achieve)

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TVT.2020.3046413

Ref	Focus	Simultaneity	Network State	App Requirements
[9]	QoS Aware Approach for Cellular Offloading	Only for offloading in one direction (Cellular to V2X not viceversa)	Not considered	Not considered
[10]	Floating Car Data offloading	Only for offloading in one direction (V2X to Cellular not viceversa)	Not considered	Not considered
[11]	Cooperative Data Dissemination using Game Theory	Yes, via cluster heads	Not considered	Partially (throughput but not latency)
[12]	Network selection in Heterogeneous Networks using Information Theory	Yes, allows for network selection	Considered	Not Considered
[13]	Cluster-based hybrid LTE-802.11p architecture	Yes, via cluster heads Minimizes cellular infrastructure usage	Considered	Considered but latency values too high
[14]	Self-adaptive network selection	Yes, allows for network selection	Considered	Considered but not for safety applications
This	Automated Decision System for VANETs	Yes, allows for network selection	Considered	Considered

TABLE I: Existing dissemination schemes and their approach to key areas of HetVNETs

the 100 ms critical threshold required for safety applications. Moreover, the particular beacon frequency in which the application operates is usually never mentioned. The same can be said for their specific reliability and communication mode requirements.

Considering the related existing dissemination schemes, we identified three key areas while developing our proposed system: First, *simultaneity*, i.e., the ability of a scheme to exploit multiple networks in parallel; second, the *network state*, i.e., the ability of a scheme to consider current network status (e.g., congestion and number of users) and adapt to changes of this status; and third, the *application requirements*, i.e., the ability to consider the requirements of a particular application (e.g., latency, beacon frequency, etc.) and satisfy these requirements. Table I shows a comparison of related existing approaches concerning these key areas.

Safety applications and more specifically, cooperative awareness applications, must meet strict reliability and latency requirements over a wide variety of network conditions. One key advantage of our approach is that we start from these strict requirements to build our decision system from the perspective of a single vehicle. This system, then, incorporates knowledge about the state of the network and the inherent advantages of the distinct available access mechanisms to fulfill those strict requirements, resulting in a scheme that boosts the performance of applications in their most critical areas.

III. Intelligent system based on Decision Analysis

As previously mentioned, in the heterogeneous vehicular network shown in Fig. 1, there are typically two types of communications links: V2V and V2I. V2V allows for short and medium range communications, offering low deployment costs and supporting short message delivery with low latency. V2I, besides extending the coverage via base stations or road side units (RSU), also enables external connectivity to infotainment applications via Traffic control centers and the Internet.

Typically, IEEE 802.11-OCB is considered more suitable for V2V communications than the cellular network, in which interference is a major issue since device-to-device (D2D) links share the same radio resources with other links in the LTE network. Channel congestion, in IEEE 802.11-OCB, used to be an important issue because the probability of collisions, related to the CSMA protocol, increases with the number of neighboring vehicles. This resulted in high end-to-end latency and low channel utilization [1]. However, recent research focused on 802.11-OCB has brought forth several standard congestion control algorithms that manipulate important parameters (e.g., transmission power and beacon frequency) before the conditions become critical [15], [16]. Nevertheless, although the impact of channel congestion has been reduced, it can never be fully eliminated. A large number of users (and higher data loads) will always eventually result in higher latency values due to the nature of the channel access mechanism. Meanwhile, LTE is more suitable for V2I communications since it provides wide coverage, a robust mechanism for mobility management, high uplink and downlink capacity, a centralized flat architecture, and high efficiency for broadcasting [5].

To exploit the characteristics of the different access networks, we propose an Intelligent Decision Framework, preliminarily introduced in [17] and illustrated in Fig. 2. This framework is expected to improve the performance of the network both in terms of total throughput and end-to-end delay by allowing a single application to take full advantage of all the individual radio access networks (RANs) working in parallel. Our first approach to implement the decision-rule module of the framework in Fig. 2 is based on a decision tree, illustrated in Fig. 3.

Note that each data flow has its own rules, which depend on the network type, current network conditions, application requirements, and link direction (i.e., uplink or downlink). The differentiation of link direction is necessary since a single vehicle typically has less information available than the network infrastructure at the moment of making a decision. The hierarchical tree characterizes the decision process of a single application when sending data to other vehicles in the network in such a way that, for each data flow, the sender attempts to minimize the end-to-end delay and improve the throughput of the system without compromising the reliability requirements of the application.

A. Use case with IEEE 802.11-OCB and LTE

Consider a typical safety application in which every vehicle continuously sends beacon messages to all its neighbors. The most critical requirement is that end-to-end delay for a transmission must not exceed 100ms, otherwise the receiver

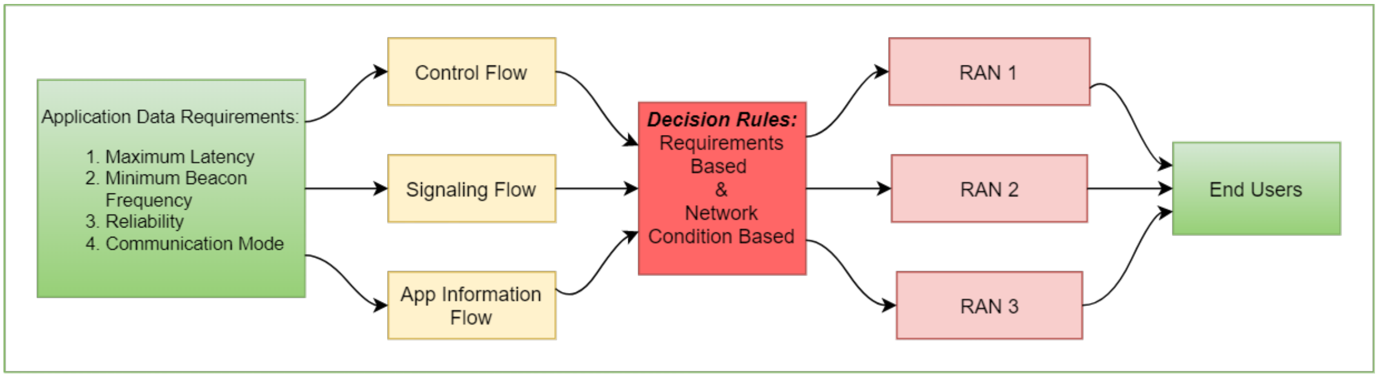


Fig. 2: Proposed Framework

does not have time to react, especially in the case of emergency applications. For most scenarios, a sending frequency of 10 Hz is required by the ETSI standard, but there are also scenarios requiring only 2 Hz [18], [19].

The access networks available are: IEEE 802.11-OCB in ad-hoc and infrastructure modes (i.e., there are RSUs available), and the LTE cellular network. For any access mechanism, the total end-to-end delay can be generalized as follows:

$$T = T_a + T_{tx} + \delta + T_p. \tag{1}$$

T_p is the processing time, which can be assumed as a constant value because it depends on the processing capabilities of the end device. The propagation delay is δ, which depends on the length of the physical link, and the transmission delay is T_{tx}, which depends on the transmission rate and packet size.

Therefore, we only need to focus on the difference between the access mechanisms (T_a) for each type of network. For 802.11p ad-hoc communication, we simplify the model presented in [20] for the Distributed Coordination Function (DCF) delay to obtain the access time in basic mode (i.e., without RTS/CTS). Then, given the workings of the DCF mechanism, if the channel is detected idle for a period of time (T_{DIFS}), a station can transmit immediately. Otherwise, a collision is detected, and the station will defer until the end of transmission while a random backoff interval is selected. Taking that into account, we use the following models for $T_{Success}$, the average time the channel is sensed to be busy due to a successful transmission, and $T_{Collision}$, the average time the channel is sensed busy by each station during a collision:

$$T_{Success} = T_{DIFS} + \frac{H+P}{C_d} + \delta + T_{SIFS} + \frac{ACK}{C_c} + \delta, \tag{2}$$

$$T_{Collision} = T_{DIFS} + \frac{H+P}{C_d} + \delta, \tag{3}$$

where H is the packet header, P is the payload in number of bits, C_d is the capacity of the link (in bits per second) for data channel, and C_c is the capacity for control channel. The header H consists of $H = PHY_{hdr} + MAC_{hdr}$, and the payload includes the IP_{hdr}. The symbol δ denotes the propagation delay inside the end device, depends on the PHY layer, and accounts for the time required to signal the state of the channel to the MAC layer.

Moreover, T_{SIFS} is the time the receiver waits to send the ACK package back to the transmitter. Because we are only interested in the broadcast scenario, we can ignore the sending of the ACK package. Thus, equation 2 is reduced to:

$$T_{Success} = T_{DIFS} + \frac{H+P}{C_d} + \delta. \tag{4}$$

In the broadcast scenario, the average time the channel is sensed to be busy due to a successful transmission is identical to the average time the channel is sensed to be busy during a collision. For a preliminary analysis we are only interested in getting a good approximation. To do this, we can measure the delay of every user in the network and then take the average of the measurements. If we know that a user senses that the channel is busy N times while attempting a transmission, then the average packet delay is given by:

$$T = N * T_{Collision} + \sum_{i=1}^{N} (\frac{CW_{Min}-1}{2}) * ST + T_{Success}, \tag{5}$$

where ST is the length of a time slot and CW_{Min} is the minimum size of the contention window. During the backoff procedure, the backoff time is uniformly chosen in the range $[0, CW_{Min}-1]$ interval. For our approximation, we take the value $\frac{CW_{Min}-1}{2}$ as it represents the average value over the distribution. Because the ACK cannot be used in the broadcast scenario, the mechanism only has one backoff stage and CW_{Min} does not increase after each retransmission attempt. Thus, each time the channel is sensed to be busy a user waits, on average, a time equal to $\frac{CW_{Min}-1}{2}$ before sensing the channel again.

Meanwhile, for infrastructure-based 802.11p, the EDCA mechanism includes the use of the AIFS differentiation and virtual collision mechanism specified in the 802.11e standard [21]. In the same work, the authors provide a simplified delay model for the channel access time in basic mode:

$$T_{Success} = T_{AIFSmin} + \frac{H+P}{C_d} + \delta + T_{SIFS} + \frac{ACK}{C_c} + \delta, \tag{6}$$

$$T_{Collision} = T_{AIFSmin} + \frac{H+P}{C_d} + \delta, \tag{7}$$

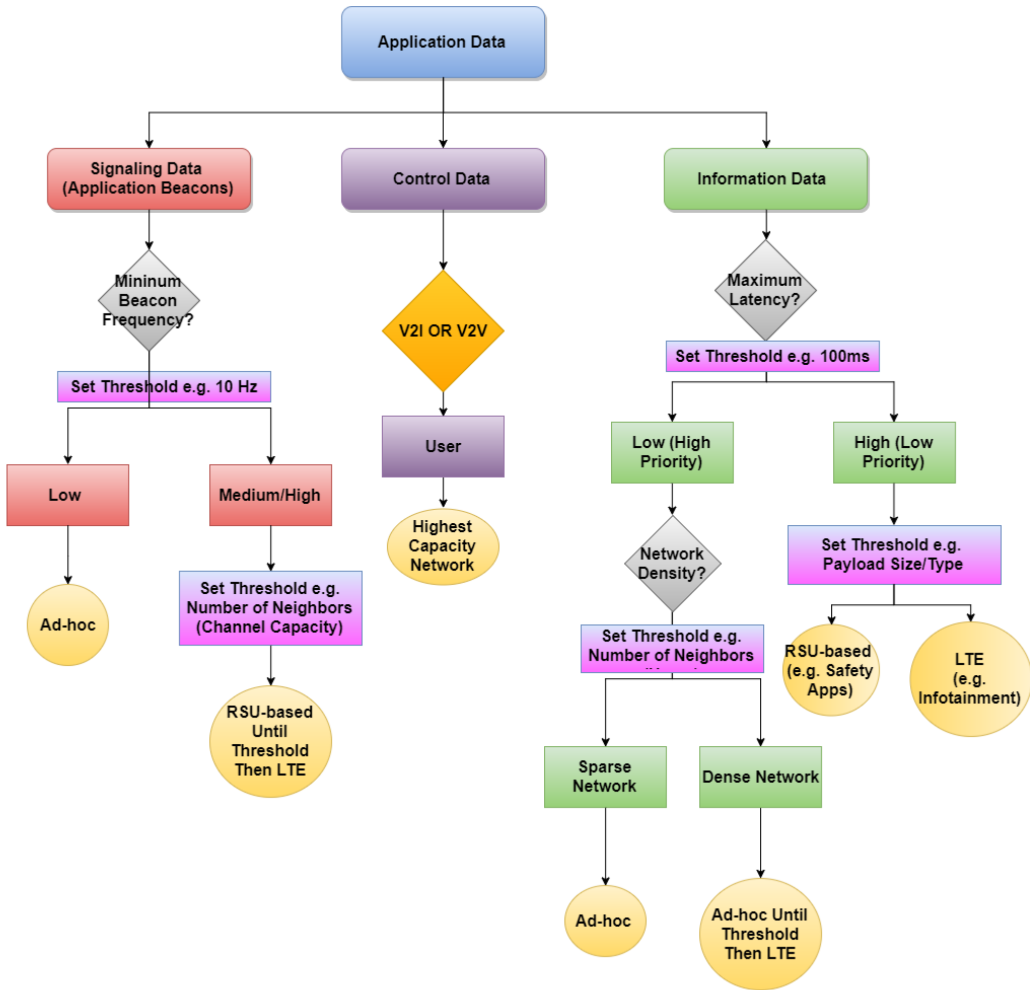


Fig. 3: Decision Tree

As in the DCF case, because we are only interested in the broadcast scenario, we can disregard the part relating to the ACK message, and the average time of a successful transmission is reduced to:

$$T_{Success} = T_{AIFSmin} + \frac{H+P}{C_d} + \delta. \tag{8}$$

Lastly, if a user experiences an average of N collisions before a successful transmission, then the average packet delay per user is again given by equation 5. Since we are using only one type of AC (also known as service class), the model is essentially the same as the ad-hoc mode if $T_{AIFSmin} = T_{DIFS}$ (they usually differ by a constant value). It should be noted that this model may be extended to account for the RTS/CTS mechanism and different AC.

In the case of LTE, the main difference between particular delay models arises from the underlying scheduling mechanism. CAM exchanges in LTE involve transmissions from vehicles to the infrastructure, followed by a message distribution from the infrastructure to the vehicles concerned. Unicast is always used for uplink transmission; in such a case, the challenge is to select the most appropriate channel type without congestion risks. The RACH is a common uplink transport channel usually selected for signaling and for transmitting small data amounts, such as CAM and DENM messages [22]. On the other hand, the PUCCH-based transmission is not susceptible to collisions and does not include backoff periods. In [23], the authors compare the performance of PUCCH with the RACH mechanism. Among the advantages of scheduling via PUCCH are high reliability and nearly deterministic data delay values. Assuming a data packet transmission takes 1 sub-frame of 1ms, the mean packet delay can be obtained as follows:

$$E[\tau] = T/2 + T_0 + 1, \tag{9}$$

where T_0 (8ms) is the PUCCH procedure duration and T (10ms) is the PUCCH scheduling request periodicity. It is worth highlighting that the use of PUCCH is only for uplink communications from a vehicle to the base station. Afterward, it is the LTE-based multicast mechanisms that deliver the message to the other vehicles as needed.

B. *Proof of concept of Decision Tree*

In this section, we evaluate the effectiveness of the decision tree via simulations. The simulation platform is based on *Veins* [24], an open source framework for running vehicular

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TVT.2020.3046413

network simulations, and more specifically on the extension *Veins LTE* [25]. Veins LTE includes a basic decision-making template inside its application development module. Thus, one can develop a custom message dissemination scheme inside the simulator. Using this module, we did deployed our decision system in each vehicle in the simulation framework. This allowed us to achieve the desired behavior within the simulation.

It is important to note that *path loss models* are central to accurately modeling information propagation in a vehicular network. To achieve this, we used the Two-Ray Interference model that captures ground reflection effects [26], [27]. We also assumed that the interference due to inter-street beacon messages can be mitigated by controlling the *maximum interference distance* parameter provided by the simulator. Note that the simulator also includes a simple obstacle shadowing model that has been calibrated and validated against real-world measurements [28], [29].

In the simulated scenario, vehicles are moving on a given path through a road that contains one intersection, as shown in Fig. 4. A typical collision warning application was implemented for data dissemination. In the application, every vehicle consistently sends beacon messages (CAM) to all its neighbors; when a collision is detected, the collision warning information is relayed to all vehicles in the vicinity.

Table II lists the parameters used in the simulation scenario. The most critical requirement to meet, for each simulated access network, is not to exceed 100 ms for end-to-end delay. Otherwise, the receiver may not have enough time to react, especially in the case of emergency messages.

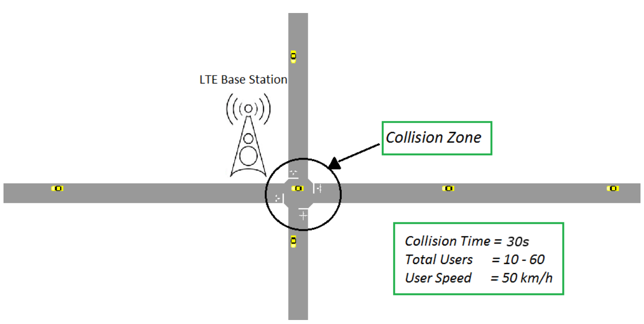


Fig. 4: Simple Simulation Scenario

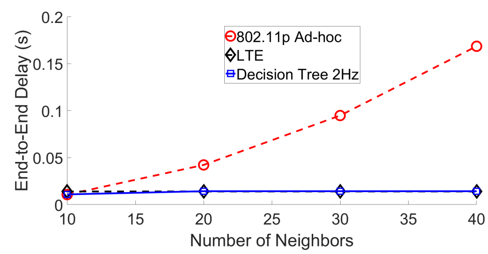
Figures 5a and 5b show the average MAC layer delay for frequencies 2 Hz and 10 Hz, respectively. We observe that when access networks work in an isolated manner, both are able to deliver packets in less than 100 ms for a certain number of neighboring vehicles. However, beyond the value of 12 neighbors for 10 Hz transmission (30 neighbors for 2Hz resp.), the average delay for the IEEE 802.11-OCB network becomes higher than the critical value. In both cases, the IEEE 802.11-OCB network is capable of achieving a lower average delay in low density scenarios (10 neighbors or less) compared to its LTE counterpart. However, as the density increases, the LTE network is able to maintain a more stable average delay due to its advanced multicasting capabilities.

In both frequency scenarios, when the decision tree is introduced to exploit the heterogeneous network, the system's

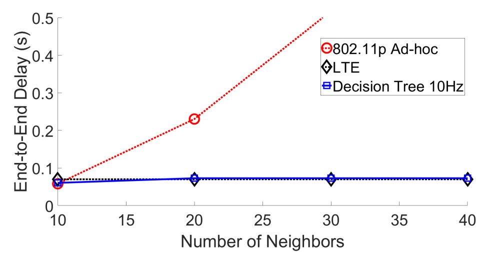
Parameter	Value
Number of Vehicle Users	10 - 60
Beacon Frequency	2Hz or 10Hz
Beacon Size (Bytes)	250
Notification Message Size (Bytes)	500
LTE Base Stations	1
Vehicle Speed	50Km/h
Traffic Collision Duration	30s

TABLE II: Simulation Parameters

delay is slightly higher than that of the faster network for a given traffic density. This is the expected behavior because, in an ideal scenario, the value for the delay using the decision system should be the minimum value among all the networks, with the IEEE 802.11-OCB exploited for short-range transmission and LTE employed as an extension to achieve long-range coverage of data dissemination.



(a) 2Hz Scenario



(b) 10Hz Scenario

Fig. 5: Average End-to-End Delay

The packet delivery ratio (PDR) for the collision notification messages is shown in Fig. 6. We observe that in the low frequency case, the IEEE 802.11-OCB network outperforms its LTE counterpart. However, as the beacon frequency increases, the high capacity nature of the LTE cellular network allows it to maintain a consistent performance in the delivery of messages while the ad hoc network performance degrades due to its contention-based nature.

In both frequency cases, once a threshold of 15 neighbors is reached and joint-network use starts, the system experiences a boost in packet delivery. For the 2Hz frequency scenario, a 32% increase in total PDR is achieved using the proposed decision tree, whereas a 42% improvement is achieved for the 10 Hz scenario. The packet delivery improvement is proportional to the difference in the number of neighbors that can now be reached under the critical threshold of 100 ms.

Finally, Fig. 7 shows a comparison between the delay obtained using simulations and those delays calculated using the analytical models introduced in section III-A. We observe that the simulation results follow a pattern similar to those

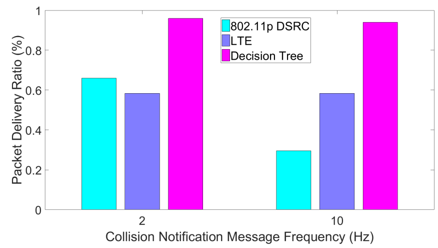


Fig. 6: Packet Delivery Ratio (PDR) versus packet frequency (Collision Notification Messages)

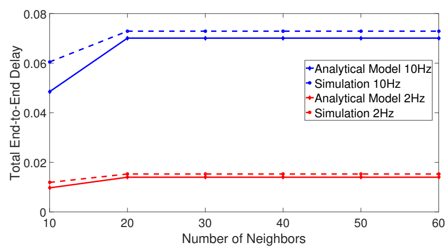


Fig. 7: Decision Tree Total Delay Analytical Model vs Simulations

produced by analytical models. In fact, the only difference is a slight increase in the delay (2-3ms approx.) that can be attributed to the time it takes the decision tree to process each packet. It follows that the preliminary results can be considered as valid for both real traffic models and simulations. In the next section, more realistic scenarios are used to test the decision system.

C. Realistic Scenario Simulations

To obtain more realistic results, real city traffic data were used from the TAPAS Cologne simulation scenario [30]. This scenario describes the traffic within the city of Cologne, Germany for a complete day.

The advantage of using this data is that the SUMO configuration files are already provided in the project repository; thus, they can easily be adapted to the application under consideration. The disadvantage is that it is impossible to identify the exact density at any point in the simulation, which means the results cannot be compared with those shown in section III-B. However, it is not necessary to simulate the entire network as one scenario. Because our aim is to test the system under different network conditions, we can study different sections of the network separately, and each of these sections can be seen as a different simulation scenario (downtown, highway, suburban areas, etc.). Studying each scenario separately also reduces the computational time and resources required to run each scenario.

D. Discussion of Simulation Results

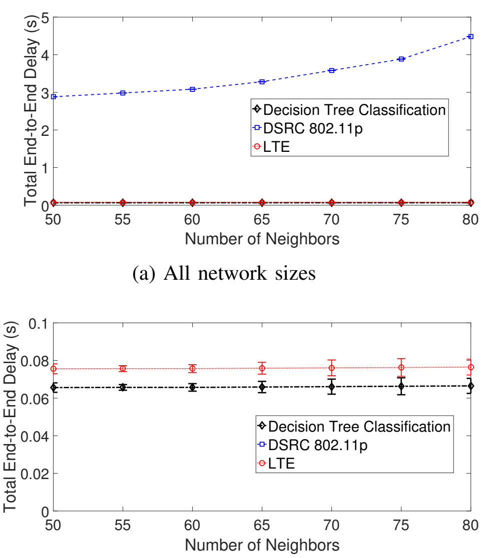
In this section, the simulation results for the three individual scenarios selected from TAPAS are presented and analyzed separately. Also, the results for different network conditions are tabulated into a single graph, and the results are then compared with the preliminary results. Unlike the preliminary results, where both 2 Hz and 10 Hz beacon frequency scenarios were studied, in this section we focus only on the 10 Hz scenario–the value used by most safety applications defined in the ETSI (Europe) and SAE (USA) standards. All three simulation scenarios were repeated a total of 30 times, 10 times for each access technology analyzed.

E. Downtown Cologne

Fig. 8a shows the results for the downtown Cologne scenario. Since 802.11-OCB is unable to perform at acceptable values under these conditions, the focus will be on analyzing the performance of the Decision Tree against LTE.

Fig. 8b shows that the decision to use LTE over 802.11-OCB dominates given the higher channel capacity and superior broadcasting capabilities of LTE. This is expected given it is a high density scenario.

Even though the performances of LTE and the decision tree follow the same pattern, latency values of the decision tree are lower (by 8 ms on average) because a fraction of the messages have been sent using 802.11-OCB.



(a) All network sizes

(b) Zoomed View

Fig. 8: TAPAS Cologne High Density Scenario Results

F. Suburban Cologne

Fig. 9 shows the results for the suburban/residential Cologne scenario. As expected, since this is a low density scenario, the decision to use 802.11 over LTE dominates given that the 802.11-OCB standard was designed precisely for low latency/high throughput communications over short distances, and the number of neighbors is not high enough to make the channel access delay an important factor in the decision.

Note, that in this scenario, the performance of the Decision Tree is actually 2 ms worse on average versus using the 802.11 network because of the added delay in computing the decision using the tree. However, this added delay is not high enough to impact the decision-system performance in low density scenarios.

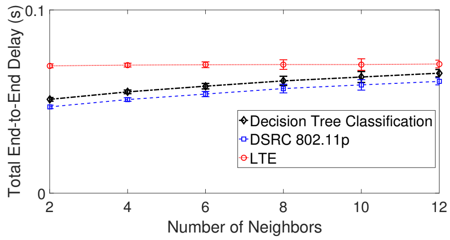


Fig. 9: TAPAS Cologne Low Density Scenario Results

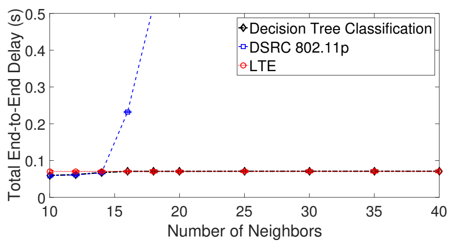


Fig. 10: TAPAS Cologne Mix Density Scenario Results

G. Mixed Area Cologne

Fig. 10 shows the results for the mixed suburban/commercial Cologne scenario. As expected, since this scenario contains both low and high vehicle density moments during the simulation, the Decision Tree chooses between 802.11 and LTE accordingly using the advantages offered by both individual networks to boost the performance of the system as required.

H. Integrated Simulation Results

Fig. 11a shows the complete tabulated results for the TAPAS Cologne scenario. It can be observed that the Decision-Tree-system behavior mimics that obtained during the preliminary results. In low density scenarios, the Decision Tree chooses 802.11 due to its low latency and high throughput capabilities. Then, as the vehicle density increases, it chooses LTE to take advantage of its multibroadcast features.

The major difference that can be observed with respect to the preliminary results is that, at a value of approx. 50 neighbors, the performance of the Decision Tree is superior to the individual networks even when taking into account the small delay in computing the actual decision. This occurs because the simulations are now using a more realistic model for the LTE Base Stations (or eNBs), which takes into account the use of Resource Blocks (RB).

An RB is the smallest unit of resources that can be allocated to a user by the scheduler in an LTE Base Station. Considering that most of the LTE network service providers use 10 MHz channels, the simulations also assumed 10 MHz channels for the LTE eNB. Then, if 100 RBs are available during each Transmission Time Interval (1 ms) according to 3GPP specifications, and the minimum resources that can be allocated to a user as per standard are 2 RB in time domain, then at most, an eNB can schedule 50 users per interval if it uses an ideal scheduler, which is commercially unfeasible. At that density value, the LTE network will, therefore, necessarily show decreased performance.

However, at a network vehicle density value of 50, the Decision Tree is able to respond to the LTE eNB limitation by using the 802.11 network to service some of the vehicle users in the network. This response overcomes the performance limitation shown in the preliminary results, in which at every point the system had a higher latency than the best performing network. Then, even while considering the small delay in computing the decision, the latency performance of the decision system eventually beats the best performing network by taking advantage of both networks in parallel.

Fig. 11b shows the PDR ratio obtained for the complete TAPAS Cologne simulation scenario. As expected, the Decision System performance is superior to both individual networks showing a large gap in performance against both 802.11 and LTE. In this case, the decision system's PDR advantage over the LTE network is approx. 39%, falling just a little short of the preliminary results of 42% shown in Fig. 6. It is important to remember that this boost in packet delivery is proportional to the difference in the number of vehicle users that can be reached under 100ms.

Finally, Fig. 11c shows the average packet losses incurred by each network. Here, the difference between considering the LTE Resource Blocks in the simulation can also be observed. When the network reaches a density value of approx. 50 neighbors, the LTE network starts to lose fewer packages than before but maintains the same average PDR due to resource constraints, i.e., the number of application messages that are generated but never sent actually increases because the eNB is incapable of assigning more resources to serve all the users.

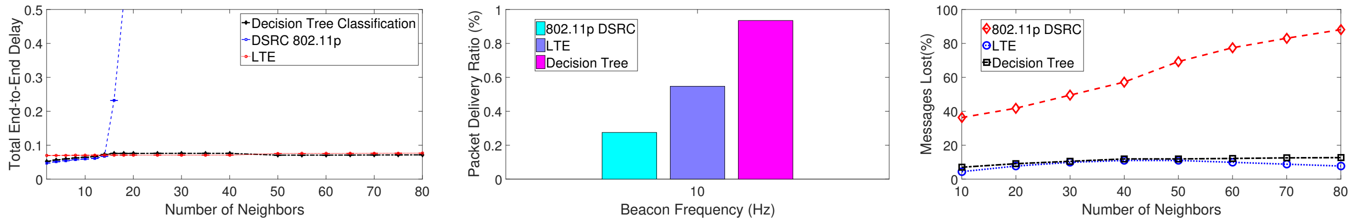
These analytical results validate the decision system approach in a realistic simulation scenario using real traffic data to test a variety of network conditions and more accurate mathematical/computational tools to model the different entities present in the network. The decision tree system has been proven capable of boosting the performance of a standard safety application by taking advantage of the multiple networks it has at its disposal. Then, using the decision system approach, not only the latency but also the throughput values of an application can be improved because more users are reachable under the critical time threshold of 100 ms.

IV. Decision System Automation

A. Decision Tree Learning

In the previous section, the use of a decision tree was validated using both analytical and realistic simulation results. We manually designed the tree for a specific class of applications using what is commonly known as *decision analysis* to develop a set of *if-else* type of rules for every particular type of data generated by the application. If the communication scenario changes, or if more information needs to be incorporated to make the decision, then there are two possibilities:

1. Develop a new tree: This could be done, for example, in the event of a change in application family. Among safety applications, there are different classes that vary both in terms of requirements and functionality, so that a



(a) Average End-to-End Delay (b) Packet Delivery Ratio (c) Average Packet Losses

Fig. 11: Full Scenario Results

change in the application family may require an entirely new tree.

2. Extend the existing tree: This approach could be used to incorporate different communication scenarios into one tree. This would require the development of more rules that would increase the size and complexity of the decision tree.

The problem with both these approaches is that the tree must be constructed manually. Given that the requirements change for every family of applications and for every different communication scenario, a large number of different trees would have to be designed and implemented to obtain a truly robust and generalized decision system. However, while doing this manually is unfeasible, the decision tree approach has proven itself capable of solving the task at hand by improving the performance of the applications. Thus, we now focus on automating the decision tree approach to create a more robust and adaptable system.

To automate the decision tree creation process, we used a machine learning approach called *decision tree learning*. This technique, typical in the field of data mining [31], uses a decision tree as a predictive model. *Decision tree learning* requires the construction of a decision tree from class-labeled training tuples and, as such, it falls under the class of supervised machine learning techniques. A decision tree is a flow-chart-like structure, in which each internal (non-leaf) node denotes a test on a feature (attribute), each branch represents the outcome of a test (a decision rule), and each leaf (or terminal) node represents an outcome (categorical or numerical value). Growing (creating) a tree involves deciding which features to choose, what conditions to use for splitting, and when to stop.

This work makes use of CART to build the decision trees [32]. The GINI impurity metric used by CART is designed to minimize classification error, and thus it was selected over other algorithms because it minimizes the probability that the system chooses the wrong network.

Table III shows an example of the features that can be used to classify messages. We observe that the features are a combination of the application requirements (e.g., latency via priority), the communication scenario (e.g., who the sender of the message is) and the network conditions (e.g., network density). These features are the input of the proposed system shown in Fig. 12. The intelligent decision system is the classifier and the different networks through which a particular message can travel within the system (e.g. RAN1, RAN2 and RAN3) correspond to the target or class of the message i.e. the output of the classifier.

Features	Domain
Message Type	Signaling, Control, Data
Sender	Vehicle User or Core Network
Priority (Latency Tolerated)	Safety, Information, Entertainment
Network Density	Number of One-Hop Neighbors
Beacon Frequency	Low (2Hz) or High (10Hz)

TABLE III: Feature Space

As with the decision trees, in this framework, there is no centralized architecture. Each user has an instance of the classifier installed in the on-board-unit (OBU). When a message is generated, it gets processed by the classifier. The output corresponds to the wireless communication technology through which the message will be sent. The training delay is not relevant in our proposal because the classifier is trained offline; thus, the only added delay comes from processing the input through the classifier.

Now that the methodology has been established, the next step is to recreate the manually created tree shown in Fig. 3 using decision tree learning. To achieve this, we create a training set that reflects only those features containing values (numerical or categorical) that would be encountered in a real life cooperative awareness communication scenario.

To generate the tree, a training set with 10,000 samples was created. This involved creating both the feature vectors themselves and their corresponding target class vectors. Fig. 13 shows a 3D representation of the samples in the feature space, a combination of numerical data (e.g., number of neighbors) and categorical values (e.g., message type). This ability to manipulate both types of data at the same time is one of the most important advantages of using CART. The different marker types illustrate the different labels or target classes for each sample.

One disadvantage of decision tree learning methods is they generally cannot guarantee the return of the optimal decision tree because of the use of *greedy methods* in the tree construction. However, this issue can be mitigated by training multiple trees in which the features and samples are randomly sampled with replacement.

Fig. 14 shows the decision tree created using CART that has a structure similar to the original manually created tree. In fact, the generated tree has a *misclassification error* under

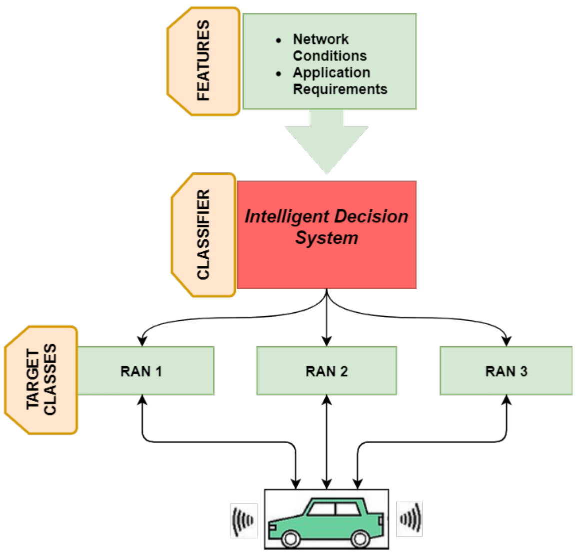


Fig. 12: System Framework

1%, which means that less than 1% of the messages would get sent through the incorrect network. This tree will then agree in 99% of the cases with the original tree.

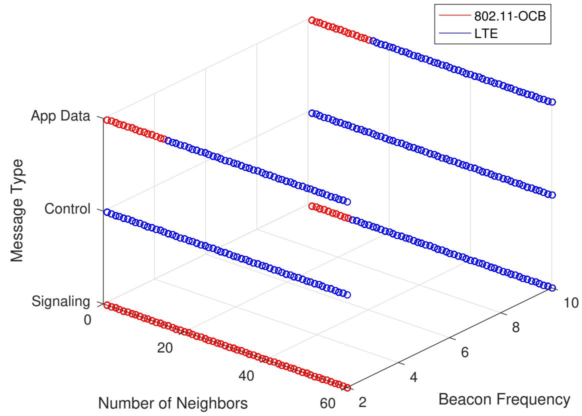


Fig. 13: Scatter Plot of the Training Set

However, this extremely low error underlines an important problem of using decision tree learning, which is the propensity of the models to *overfit* the training data. In this first example, it is not particularly important because the generated tree is designed to work for a single application type. Nevertheless, one of the objectives of this work is to obtain a generalized decision method. Thus, it is important to study the effect of increasing the training set to account for different application scenarios and types.

A training set with 30,000 samples was created to obtain a more generalized tree. This set includes samples for three different cooperative awareness applications with similar requirements but different modes of operation. The first application is a typical *V2X Cooperative Awareness* use case with a minimum beacon frequency of 10 Hz; this is the same type of application used in the previous experiments. The second application is based on a *time limited periodic broadcast on*

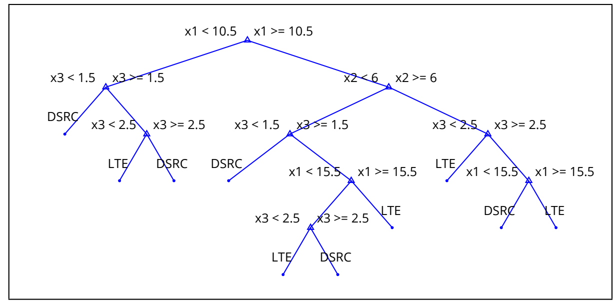


Fig. 14: 1st Decision Tree created using CART

event communication mode; this type of application only sends beacons when certain events are triggered e.g. when emergency brake lights turn on. The third application is *periodic triggered by vehicle mode* with a minimum beacon frequency of 2 Hz. Fig. 15 shows only a branch of the second decision tree created using CART. The main disadvantage of using decision tree learning can be observed almost immediately. The generated tree increases, in size and complexity, with the size of the training set.

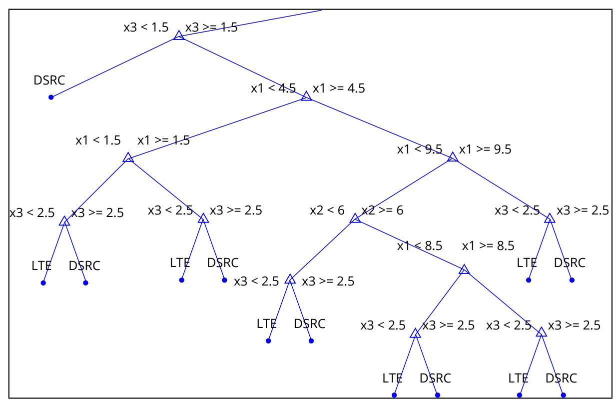


Fig. 15: Branch of the 2nd Decision Tree created using CART

The problem with *overfitting* is that it renders the decision trees unable to generalize data very well. Basically, trees that grow very deep tend to learn highly irregular patterns causing an overfit of their training sets, i.e. they will have low bias but at the cost of very high *variance*. Because of this, large scale trees tend to be unstable; thus, small variations in the data might result in a completely different tree being generated.

In terms of performance, the generated tree has a *misclassification error* of approximately 10%; then, initially about 10% of the messages would get sent through the incorrect network. This number is much higher compared to the initial tree model, which is to be expected because the initial model was concerned with only one type of application. Nevertheless, as previously mentioned, this issue can be mitigated by training multiple trees using sampling with replacements and will be addressed in the next section.

The findings of this section can be summarized in four key points: (1) decision trees have the classification power required to be used as building blocks for the decision system; (2) it is possible to obtain automated trees that are able to

Algorithm 1: Random Forest Algorithm for Classification

1 Generate Random Forest (X, Y, N);
Input : Training Set $X = x_1, ..., x_n$ with targets $Y = y_1, ..., y_n$
Output: A Random Forest Classifier with N decision trees
2 Bootstrapping:
Perform bootstrapping N times over the training set by sampling with replacement.
For $n = 1, ..., N$ sample, with replacement, to select m examples of the training set. The resulting subsets X_n, Y_n are called bootstrap samples
3 Fit Trees using feature bagging:
Use a random subset of features to perform CART procedure and train a classification tree T_n on each set X_n, Y_n from $n = 1, ..., N$
4 Tree Bagging:
After training N decision trees , classification is performed by using majority vote.

handle generalized data. (3) to obtain a fully *automated and generalized* system, it is necessary to improve the performance of decision tree learning models; and (4) most importantly, we have shown that the tree creation process can be *automated* using machine learning.

B. Random Forests

Decision trees have shown a great capacity to act as decision makers in heterogeneous vehicular scenarios. However, as shown in the previous section, their performance as classifiers must be improved to obtain a fully *automated and generalized* system. To achieve this, an ensemble learning technique, called Random Forest (RF), will be used.

Random Forest [33], [34] is a supervised learning algorithm that creates a forest of random decision trees and qualifies as an *ensemble learning* method. Random Forest averages multiple deep decision trees, trained on different parts of a training set, with the goal of reducing variance [35]. Reduced variance, however, comes at the expense of a slight increase in bias and some loss of interpretability; however, this method generally boosts the performance in the final model.

The current version of RF developed by Leo Breiman [34] combines the random selection of features with Breiman's own idea of *bagging* [36] (short for bootstrap aggregating). Algorithm 1 shows a summary of the Random Forest algorithm implemented for classification. Bagging is a two step process that involves bootstraping (step 2) and aggregating (step 4). Using this method, multiple versions of a predictor are generated to obtain an aggregated predictor.

This combination of techniques allows the model to limit *overfitting* without increasing the error due to bias. As the number of trees increases, the likelihood of overfitting the forest decreases. RF solve the three most important issues of decision tree learning presented in the previous section (overfitting, variance and bias). This makes it an ideal candidate to achieve the desired *automation* and *generalization* of the decision system framework.

C. Automated System Performance

Before analyzing the final results, there is one more concept that needs to be introduced: the out-of-bag (OOB) error. OOB allows the prediction error of machine learning models that use *bagging* to be measured so as to sub-sample data samples used for training. Because all trees in the forest are trained using a portion of the training set, there is a set of samples that has never been seen by each individual tree. This set is called *out-of-bag* examples. There are N such sets (one for each tree that is generated). The OOB classifier is the aggregation of votes ONLY over the trees that do not contain a specific sample (x_i, y_i). The OOB estimate for the generalization error is given by the error rate of the out-of-bag classifier on the training set. This provides empirical evidence to show that the out-of-bag estimate is as accurate as using a test set of the same size as the training set [36].

Applying the above concepts of the RF algorithm, the next step is to test it using the same Cologne simulation scenario described previously and compare its performance to the decision tree. Next, the fully trained RF is put directly in the simulator to make decisions at the moment of message generation, replacing the decision tree. The rest of the simulation parameters are left untouched with respect to the experiments described in the previous sections.

Fig. 16 shows the OOB error obtained using RF on the TAPAS Cologne dataset analyzed in the previous section. We can observe that as the number of grown trees increases, the OOB error decreases. However, this comes at the cost of a higher computational time and resources because more trees need to be generated and tested. At a value of 10 trees, the OOB error is 1.2% (for 20 trees, the error is 0.8% resp.), indicating that the RF algorithm agrees with the manually built tree approx. 98.8% of the time (99.2% resp.).

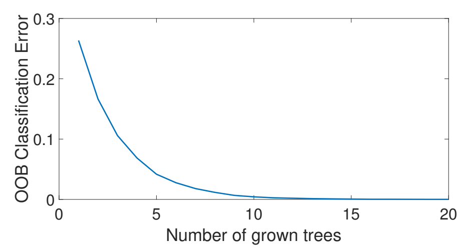
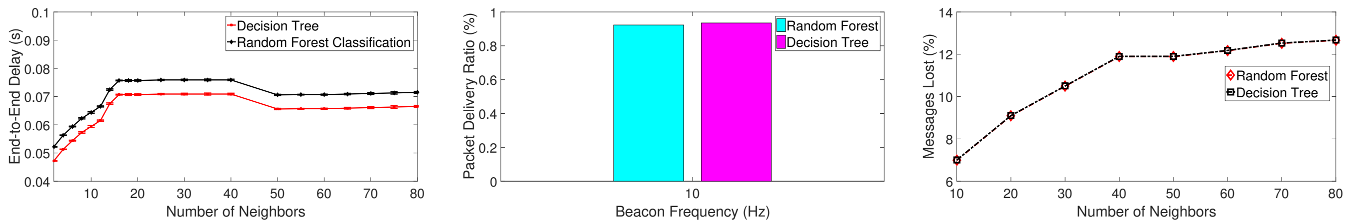


Fig. 16: Out-of-bag Error versus Number of Trees

Fig. 17a shows a comparison between the end-to-end latency values of the RF and the decision tree. This comparison uses an RF algorithm with 10 grown trees. We observe that the values obtained by the two methods differ only on a mostly constant value of approx. 5 ms. This 5 ms represents the computational time to make the decision by aggregating the 10 trees instead of using just one tree.

Fig. 17b shows a comparison between the packet delivery ratio of RF and the decision tree in which the results are practically identical. The difference is approximately 0.8% between both values with a slight advantage for the decision tree. This small difference is close to the value predicted by the OOB error as an estimation for the generalization error.

Finally, Fig. 17c shows the difference between the percentage of lost packets in the network. Here, RF shows a slight advantage of approx. 1% on average. Nevertheless, the values are essentially the same; this was expected given the highly similar results for the delay and PDR.



(a) Average End-to-End Delay (b) Packet Delivery Ratio (c) Average Packet Losses

Fig. 17: Cologne scenario comparison between Random Forests (10 Trees) and Decision Tree

We conclude that RF is capable of replicating the results obtained with the manually built tree but with the added advantages of not having to build the classifier manually and of being able to adapt to changing requirements. If another application needs to be added or if the network conditions change abruptly, a simple retraining of the classifier is all that is needed to keep the system running with high performance.

V. Performance Comparison

In this section, we compare the performance of RF against the state-of-the-art related contributions (e.g. [11], [13], [14]). We focus on the similarities between the performance metrics used in other methods and our system (latency and PDR) to decide which could be compared to our proposed scheme. For this reason we did not consider [11] and [14] to be suitable for comparison because they used different, noncomparable metrics. In Ucar et al. [13], PDR and average delay (among other metrics) were used to measure performance of cluster-based IEEE 802.11p and LTE hybrid architecture for message dissemination. The architecture was based on the selection of *gateway* vehicles (i.e. cluster heads) that forward (offload) the data of all cluster members towards the cellular network. We reproduced the scenario analyzed in [13] substituting the proposed VMaSC method (one hop variant) with our own decision system to generate a one-to-one comparison. It is worth noting that VMaSC outperforms most of the other classic VANET multihop hybrid architecture algorithms such as NHOP [37] and MDAC [38].

The scenario consisted of a five-kilometer, two-lane, two-way road. Vehicles were injected into the road according to a Poisson process with a rate equal to two vehicles per second. The vehicles had a maximum variable speed ranging from 10 to 35 m/s. Thus, the average number of neighbors for any vehicle ranged from 10 to 18 at different times for different scenarios. Because the communication scenario is similar to those we used to train and evaluate the RF scheme, we did not perform any additional adjustments to the classifier shown in the previous section.

Fig. 18a shows a comparison between PDR obtained using VMaSC and our decision system. Our decision system held a slight advantage for speeds up to 32m/s. Beyond this speed, VMaSC slightly outperformed our system.

Fig. 18b shows a comparison between PDR of VMaSC and the decision system with respect to vehicle density. In this comparison, the decision system outperformed VMaSC, an expected outcome given that our method incorporates the number of neighbors as a feature of the decision system.

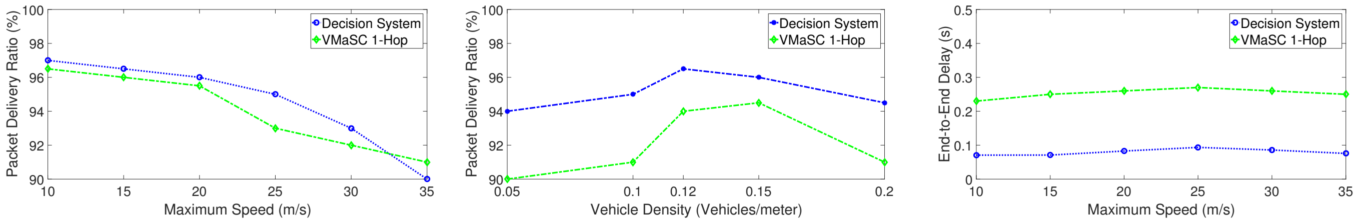
Finally, Fig. 18c shows a comparison between the end-to-end delay of VMaSC and the decision system with respect to the maximum speed of vehicles. In this comparison, both methods maintain a relatively stable performance as the maximum speed increases. However, the average delay of the network using VMaSC is over the 100ms critical threshold required for safety applications (the same holds true for the two and three hop variants of the algorithm). Our decision system achieves an average under this value for all speeds tested.

It is worth noting that VMaSC performs better when the number of hops between vehicles is greater than one, especially in regards to the packet delivery ratio. However, a key advantage of our scheme is that it can be incorporated into other schemes. Indeed, the decision-based system does not put any restrictions on whether the vehicles can be in a cluster or any other type of platooning mechanism. Moreover, our scheme is not limited to the use of a single decision-maker (classifier) for the entire network. Multiple RFs could be used within the system e.g. one for cluster heads and another for the remaining vehicles. Furthermore, the clustering information could even be incorporated into the RF training as a feature. Thus, our methodology could potentially be integrated with other hybrid architectures such as VMaSC to achieve a greater boost in performance by combining their advantages with our classifier approach.

One last thing worth noting is that our methodology can be easily integrated into an SDN-based architecture [39]. Indeed, having a centralized controller architecture means that we can handle the deployment of the classifier into the vehicles. Also, whenever the classifier needs to be updated, the training can be done in the cloud to avoid adding overhead and extra delay to the message dissemination. Then, using a scheme such as the one proposed in [40], an SDN architecture can be used to distribute the updated classifier to vehicles in the network.

VI. Conclusion

In this work, we introduced a methodology to improve the performance of safety applications deployed over vehicular networks. Our scheme exploits the inherent network diversity present in VANETs to create a decision system that takes into account the network conditions (e.g. latency, channel congestion and capacity) and the application requirements



(a) PDR vs Max Vehicle Speed (b) PDR vs Vehicle Density (c) Average Packet Delay vs Max Vehicle Speed

Fig. 18: Performance Comparison between Decision System and VMaSC 1-Hop

(e.g. maximum tolerable delay and message exchange rate) to choose the best available network at the moment of message generation.

The first implementation of the decision system took the form of a Decision Tree. The study of its performance under a realistic traffic scenario confirmed the findings of the preliminary analysis and simulations. Using the capabilities of each network, the decision system was able to reduce latency and boost throughput. The Random Forest algorithm, an automated classifier, was then used to reproduce the performance of the manually-built Decision Tree. This automated system has the advantage of being able to adapt to different application requirements without the need to manually construct a new tree for each application. Comparing the performance metrics of our automated decision system with another hybrid architecture (VMaSC) that used the same metrics, we found that our approach outperforms most clustering based hybrid schemes in terms of latency, PDR per vehicle density and PDR for speeds up to 115 km/h.

Acknowledgment

This work was supported by the ANID Basal Project FB0008 and by the Cisco Research Grant #2019-199458 (3696).

References

[1] C. Han, M. Dianati, R. Tafazolli, R. Kernchen, and X. Shen, "Analytical study of the ieee 802.11p mac sublayer in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 873–886, June 2012.

[2] M. Kihl, K. Bür, P. Mahanta, and E. Coelingh, "3gpp lte downlink scheduling strategies in vehicle-to-infrastructure communications for traffic safety applications," pp. 448–453, 2012. [Online]. Available: http://dx.doi.org/10.1109/ISCC.2012.6249337

[3] F. Dressler, H. Hartenstein, O. Altintas, and O. K. Tonguz, "Inter-vehicle communication: Quo vadis," *IEEE Commun. Mag.*, vol. 52, no. 6, pp. 170–177, Jun. 2014.

[4] K. Zheng, Q. Zheng, H. Yang, L. Zhao, L. Hou, and P. Chatzimisios, "Reliable and efficient autonomous driving: the need for heterogeneous vehicular networks," *IEEE Communications Magazine*, vol. 53, no. 12, pp. 72–79, Dec 2015.

[5] Z. Hameed Mir and F. Filali, "LTE and IEEE 802.11p for vehicular networking: a performance evaluation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, p. 89, may 2014.

[6] A. Vinel, "3GPP LTE versus IEEE 802.11p/WAVE: Which technology is able to support cooperative vehicular safety applications?" *IEEE Wireless Communications Letters*, vol. 1, no. 2, pp. 125–128, 2012.

[7] G. A. Association, *"The case for cellular V2X for safety and cooperative driving", White Paper*, November 2016. [Online]. Available: http://5gaa.org/pdfs/5GAA-whitepaper-23-Nov-2016.pdf

[8] G. T. 21.914, "Technical specification group services and system aspects; release 14 description; summary of rel-14 work items (release 14)," V14.00 (2018-5).

[9] G. el mouna Zhioua, H. Labiod, N. Tabbane, and S. Tabbane, "A traffic QoS aware approach for cellular infrastructure offloading using VANETs," in *2014 IEEE IWQoS*, 2014, pp. 278–283.

[10] S. Ancona, R. Stanica, and M. Fiore, "Performance boundaries of massive Floating Car Data offloading," in *Proc. WONS*, 2014, pp. 89–96.

[11] Y. Li, K. Ying, P. Cheng, H. Yu, and H. Luo, "Cooperative data dissemination in cellular-VANET heterogeneous wireless networks," in *Proc. HSIC*, 2012, pp. 287–290.

[12] K. Zhu, D. Niyato, and P. Wang, "Network selection in heterogeneous wireless networks: Evolution with incomplete information," in *2010 IEEE Wireless Communication and Networking Conference*, April 2010, pp. 1–6.

[13] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based ieee 802.11p and lte hybrid architecture for vanet safety message dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, April 2016.

[14] D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, "A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 6, pp. 3033–3049, Dec 2015.

[15] D. Smely, S. Rührup, R. K. Schmidt, J. Kenney, and K. Sjöberg, "Decentralized congestion control techniques for vanets," in *Vehicular ad hoc Networks*. Springer, 2015, pp. 165–191.

[16] T. Etsi, "Intelligent transport systems (its); decentralized congestion control mechanisms for intelligent transport systems operating in the 5 ghz range; access layer part," *ETSI TS*, vol. 102, no. 687, p. V1.2.1, 2018.

[17] F. Valle and S. Céspedes, "Improving vehicular applications through network diversity," in *2016 8th IEEE Latin-American Conference on Communications (LATINCOM)*, Nov 2016, pp. 1–6.

[18] M.-a. Phan, R. Rembarz, and S. Sories, "A Capacity Analysis for the Transmission of Event and Cooperative Awareness Messages in LTE Networks," in *Proc. 18th ITS World Congress*, October 2011, pp. 1–12.

[19] T. ETSI, "102 638 intelligent transport systems (its); vehicular communications; basic set of applications; definitions," *ETSI, Sophia Antipolis Cedex, France*, 2009.

[20] A. M. Abbas and K. A. M. A. Soufy, "Analysis of IEEE 802.11 DCF for ad hoc networks: Saturation," in *Proc. IEEE IMSAA*, dec 2011, pp. 1–6.

[21] Tantra, Juki Wirawan, Chuan Heng Foh and A. B. Mnaouer, "Throughput and Delay Analysis of the IEEE 802.11e EDCA saturation," in *Proc. IEEE ICC*, 2005, pp. 3450–3454.

[22] C. Campolo, A. Molinaro, and R. Scopigno, *Vehicular ad hoc Networks: Standards, Solutions, and Research*. Springer International Publishing, 2015. [Online]. Available: https://books.google.cl/books?id=yC0RrgEACAAJ

[23] S. Andreev, A. Larmo, M. Gerasimenko, V. Petrov, O. Galinina, T. Tirronen, J. Torsner, and Y. Koucheryavy, "Efficient small data access for machine-type communications in lte," in *2013 IEEE International Conference on Communications (ICC)*, June 2013, pp. 3569–3574.

[24] C. Sommer, R. German, and F. Dressler, "Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, January 2011.

[25] F. Hagenauer, F. Dressler, and C. Sommer, "A Simulator for Heterogeneous Vehicular Networks," in *6th IEEE Vehicular Networking*

Conference (VNC 2014), Poster Session. Paderborn, Germany: IEEE, December 2014, pp. 185–186.

[26] C. Sommer, S. Joerer, and F. Dressler, "On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation," in *4th IEEE Vehicular Networking Conference (VNC 2012)*. Seoul, South Korea: IEEE, 11 2012, pp. 64–69.

[27] C. Sommer and F. Dressler, "Using the Right Two-Ray Model? A Measurement based Evaluation of PHY Models in VANETs," in *17th ACM International Conference on Mobile Computing and Networking (MobiCom 2011), Poster Session*. Las Vegas, NV: ACM, 9 2011.

[28] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A computationally inexpensive empirical model of ieee 802.11p radio shadowing in urban environments," in *2011 Eighth International Conference on Wireless On-Demand Network Systems and Services*, 2011, pp. 84–90.

[29] C. Sommer, D. Eckhoff, and F. Dressler, "Ivc in cities: Signal attenuation by buildings and how parked cars can improve the situation," *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1733–1745, 2014.

[30] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1061–1075, May 2014.

[31] L. Rokach and O. Maimon, *Data Mining with Decision Trees: Theory and Applications*, ser. Series in machine perception and artificial intelligence. World Scientific, 2008. [Online]. Available: https://books.google.cl/books?id=8B9pDQAAQBAJ

[32] L. Breiman, J. Friedman, C. Stone, and R. Olshen, *Classification and Regression Trees*, ser. The Wadsworth and Brooks-Cole statistics-probability series. Taylor & Francis, 1984. [Online]. Available: https://books.google.cl/books?id=JwQx-WOmSyQC

[33] T. K. Ho, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, vol. 1, Aug 1995, pp. 278–282 vol.1.

[34] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[35] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, ser. Springer series in statistics. Springer, 2001. [Online]. Available: https://books.google.cl/books?id=VRzITwgNV2UC

[36] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug 1996. [Online]. Available: https://doi.org/10.1023/A:1018054314350

[37] Z. Zhang, A. Boukerche, and R. Pazzi, "A novel multi-hop clustering scheme for vehicular ad-hoc networks," in *Proceedings of the 9th ACM International Symposium on Mobility Management and Wireless Access*, ser. MobiWac '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 19–26. [Online]. Available: https://doi.org/10.1145/2069131.2069135

[38] G. Wolny, "Modified dmac clustering algorithm for vanets," in *2008 Third International Conference on Systems and Networks Communications*, 2008, pp. 268–273.

[39] R. Dos Reis Fontes, C. Campolo, C. Esteve Rothenberg, and A. Molinaro, "From theory to experimental evaluation: Resource management in software-defined vehicular networks," *IEEE Access*, vol. 5, pp. 3069–3076, 2017.

[40] M. Azizian, S. Cherkaoui, and A. S. Hafid, "Vehicle software updates distribution with sdn and cloud computing," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 74–79, 2017.