This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Model-Tree-based Rate Adaptation Scheme for Vehicular Networks

Qiuyan Xia, Jian Pu, and Mounir Hamdi
Department of Computer Science and Engineering
Hong Kong University of Science and Technology
Hong Kong, China
{xiaqy, pujian, hamdi}@cse.ust.hk

***Abstract*—Rate adaptation techniques have been extensively studied for traditional wireless LANs as way to adjust the data transmission bandwidth as a function of the channel quality. However, existing solutions which are mostly based on statistics collection inherently experience a large delay in response to the wireless channel fluctuations, which is unsuitable for the rapid changing of the channel conditions in vehicular networks. In this paper, we propose an efficient self-adaptive model-tree-based rate adaptation scheme termed MTRA that can predict the packet error rate (PER) and adapt the data rate in real time. We also present a detailed methodology for the PER model tree building and update. We have performed comprehensive experimentations using ns-2 simulations which demonstrate that MTRA can achieve much better performance than the traditional rate adaptation approaches under various scenarios in vehicular networks.**[*]

I. Introduction

Data transmission rate adaptation techniques have been extensively studied for traditional wireless LANs [1], [2], [3]. However, few rate adaptation algorithms have been designed particularly for vehicular networks. Because of the high mobility in such a vehicular environment, the data rate must adapt much faster in order to be effective. While the rate adaptation algorithms can be derived from those for wireless LANs, there are still some major challenges that need to be addressed in vehicular networks [4]: (1) When a vehicle enters the communication range of another vehicle or infrastructure, how to choose the initial data rate? The initial rate used by the legacy rate adaptation algorithms may be too high and may lead to instability; (2) The traditional rate adaptation schemes do not change the data rate until collecting sufficient link quality statistics. When the WAVE (Wireless Access in Vehicular Environments) [5] Basic Service Set (WBSS) provider and the user get closer or further from each other, the sender cannot adapt the transmission rate as quickly as necessary, which causes the under-estimation or over-estimation of the potential throughput; (3) Most current rate adaptation algorithms cannot distinguish between frame-collision errors and wireless errors, causing wrong data rate decisions. In summary, existing solutions based on statistics collection inherently experience a large delay in response time, which is unsuitable for the rapid changing of the channel conditions in vehicular networks.

Besides the above challenges, we also have the following observations in vehicular networks: (1) The movements of vehicles usually follow certain routes (for example, a car repeats driving in a familiar route from home to work) so that similar environmental patterns exist and can be learnt. (2) Certain context information of the environment is highly correlated to the data rate selection. This kind of context information includes: the location information of both the sender and receiver and their neighbors as well, the relative distances and speeds, the system configurations of the roadside units, etc. (3) The traditional rate adaptation algorithms based on statistics collection are not responsive to fast changing channel conditions in vehicular networks. Given these considerations, our aim is to propose a self-adaptive learning-based algorithm for responsive rate adaptation.

In this paper, we investigate the use of model-tree-based learning method to improve the throughput performance of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) applications. We want to find a rate adaptation algorithm that is able to handle the fast changing of the wireless channel conditions with the assistance of the environmental context information. This algorithm needs to maximize the data transfer rate during a single connection and should be effective under a variety of network and environmental situations.

II. Model-Tree-based Rate Adaptation

In this section, we propose a model-tree-based learning model for data rate adaptation using the context information in vehicular networks. First we briefly review the model-tree learning method for continuous classes. Then we model the effect of context information on the packet error rate (PER) for different data rates. Next we apply this model to the rate adaptation in vehicular networks so that it can accurately predict the evolving channel quality and select the right data rate which best satisfies the performance requirements.

A. Model Tree

Model tree M5 [6] was developed by Quinlan for inducing trees of regression models. It is an extension of the well-known decision-tree learning algorithm. Standard regression is not a very potent way of representing an induced function because it imposes a linear relationship on the data. The M5 algorithm builds model trees combining conventional decision-tree learning with the possibility of linear regression functions at the leaves. The resulting decision structure is clear and the regression models are normally easily interpretable.

Model trees, being analogous to piecewise linear functions, have certain advantages compared to several other learning techniques that also predict numeric values, such as the standard regression and neural networks. The accuracy of M5 trees is similar to that of neural networks. But model trees are more transparent and hence acceptable by decision makers. They are very fast in training and always converge, and they

[*] This work was supported by Grant HKUST RPC07/08.EG09.

are generally much smaller than regression trees. The actual learning algorithm used in this paper for rate adaptation purposes is M5' [7]. The improvement of M5' over M5 is that M5' can effectively deal with enumerated attributes and missing values which are typically encountered in practice.

B. MTRA Methodology

In this section we introduce the model-tree-based rate adaptation for vehicular networks (MTRA).

System Model

Let's consider a highway scenario for both V2I and V2V communications. Although currently the majority of the vehicular applications are for safety purposes, there is a fast emerging trend that increasing vehicle users are going to use non-safety applications such as traffic querying, mobile infotainment, and multimedia content sharing among vehicles. These massive unicast traffics require fast and accurate rate adaptation to exploit the dynamic bandwidth in vehicular communications. Each vehicle is equipped with a Global Positioning System (GPS) device which reports the location information to the user. The IEEE 802.11p [5] draft standard, also referred to as Wireless Access in Vehicular Environments (WAVE), adapts the IEEE 802.11a [8] standard to the vehicular environment. It is a high data rate, short range, multi-channel wireless standard based on 802.11a PHY and 802.11e MAC. The transmission power can provide at least 1000 meter communication range.

Operation Modes

There are two operation modes: the idle mode and the busy mode. In the idle mode, the vehicle has no application data to transmit. However, the rate adaptation module may send out probing packets so as to gauge the channel condition. Thus, in the idle mode, the rate adaptation module collects training samples which are multidimensional <distance, SNR, speed, data rate, PER> (we do not include other factors such as packet sizes, weather, acceleration speeds here as they are left for future work). The values of the distances and speeds can be calculated by the location and movement information exchanged between the sender and the receiver. The SNR and data rate values for each received frame are obtained from the PLCP header of the frame. The training dataset (containing many samples) is built, populated and maintained during the vehicle's idle periods. The process of building the PER model tree is also carried out in the idle mode. The outcome of the training algorithm is a PER model tree that computes the PER for the outgoing packet given current distance, SNR, relative speed and one of the data rates. Also, multiple training datasets and/or the PER models for rate adaptation can be uploaded to a central server that forms a shared database. A user of the vehicle may either download relevant datasets to assist in the building of its own PER model, or even download the already built PER model if it matches the current environments and configurations. For the busy mode, the vehicle has application data to transmit, and it refers to the available PER model for rate adaptation in real time. Meanwhile, <distance, SNR, speed, data rate, PER> samples of communicated frames are also logged in the training database. The user may run the training algorithm M5' using the accumulated training dataset periodically, as well as remove stale samples from the dataset. The relationship between the inputs and outputs of the model-tree-based rate adaptation is shown in Fig. 1. We will next describe the construction of the PER model tree using the training algorithm M5'.

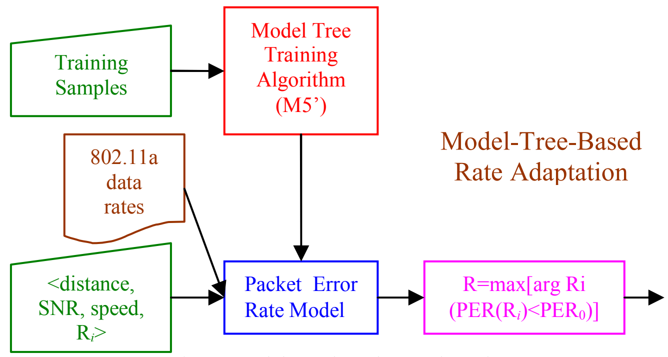

Fig. 1. Model-tree-based rate adaptation.

PER Model Tree Construction

The construction process consists of three iterative phases: data collection, model training, and model update.

Data collection: To obtain the initial dataset, we run the vehicle transmitting constant bit rate (CBR) traffic multiple times through the same route. Specifically, we can fix a data rate but vary the relative speed to obtain the training samples (i.e., the <distance, SNR, speed, data rate, PER> tuples). Then we change to another data rate while driving the vehicle at different speeds until we collect sufficient training samples. This data collection process can actually be carried out online by the roadside base stations and stored as the default configurations. Later on when the vehicle transmits packets either by active probing in the idle mode or serving the upper applications in the busy mode, the training dataset is updated by removing old samples and/or adding new samples. In active probing, the vehicle can send out probe requests periodically to the base station on the roadside and record the results of the probe response (as training samples). The data packets transmitted by applications in the busy mode can also be used to update the training dataset. In this case, these data packets are randomly selected as the probing packets.

Model training: Once sufficient training samples are accumulated, we input them into the M5' model tree training algorithm. The basic idea of M5' [7] is as follows. In the first stage, a decision-tree induction algorithm is used to build a tree. A splitting criterion is used to minimize the intra-subset variation in the class values down each branch. In the second stage, the pruning procedure makes use of an estimate of the expected error that will be executed at each node for test data. When pruning to an interior node, it is replaced by a regression plane instead of a constant value. The third stage is to use a smoothing process to compensate for the sharp discontinuities that occur between adjacent linear models at the leaves of the pruned tree, particularly for some models constructed from a small number of training samples. Finally, M5' clarifies how enumerated attributes and missing values should be handled. An example of the model tree output by M5' is shown in Fig.

2. Using this model tree, we can predict the PER for any given <distance, SNR, speed, data rate> tuple.

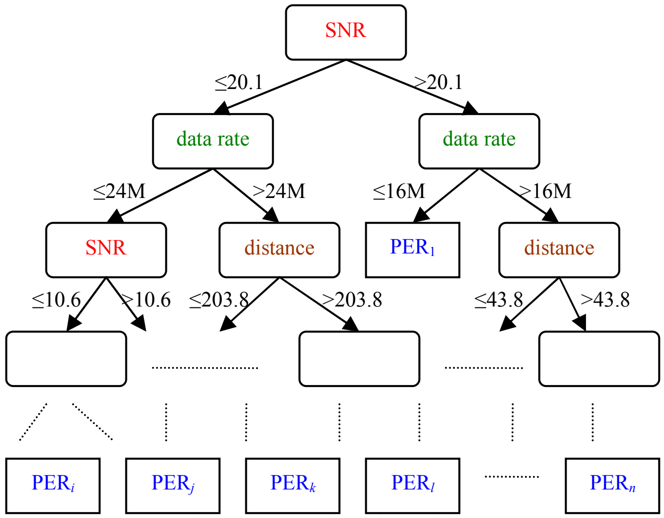

Fig. 2. An example of the model tree.

Model update: The model tree built by a certain training dataset is only applicable when it matches the current environment and settings (e.g., propagation characteristics, transmit power, speed, *etc*.). The vehicle can also upload the PER model tree to a base station to share with other vehicles. When a vehicle enters into the communication range of the base station, it checks its local database as well as the base station's to find a match. In this way, the vehicle can reduce its time and energy for training a new PER model. If the vehicle ventures into a completely new environment without any existing matched models, the PER model has to be trained in an online fashion. In this case, if the vehicle is in the idle mode, it can collect samples by active probing and perform training on them; otherwise, the vehicle can use linear interpolation of existing models to predict a proper data rate and meanwhile update the training dataset with transmissions at that data rate. As the availability of more and more accumulated learning data, the model can be improved to give more accurate predictions.

Rate Adaptation

Once the PER model tree has been built, it can be used for rate selection in real time. In our implementation, we select the maximum data rate that has bounded the PER performance by the target value PER_0. Specifically, we input the <distance, SNR, speed, R_i> tuple into the PER model for each data rate R_i, and get a set of PER values on the whole rate set. We then output the maximum data rate which has its calculated PER less than the target value PER_0. This data rate will be used for packet transmissions later, and new samples will be collected for updating the training dataset. The MTRA algorithm is sketched in Algorithm 1.

Algorithm 1. Model-Tree-based Rate Adaptation Algorithm

In the idle mode:
  Collect the training samples and build/update the training dataset *ds*
  PER_Model = M5'(*ds*)
In the busy mode:
  Find the PER Model that matches current environments and configurations (locally or from the base station)
  (the data rate R_i is in increasing order)
  for all R_i ($1 \le i \le M$) in the rate set do
    PER = PER_Model(<distance, SNR, speed, R_i>);
    if PER < PER_0 then
      Rate = R_i
    end if
  end for
  Use Rate for outgoing packets

C. *Discussion: Robustness*

Legacy rate adaptation methods for wireless LANs use packet error statistics and/or heuristics that cannot distinguish collision errors from wireless losses, and therefore, are not scalable in congested networks. For MTRA, as it utilizes the PER model which is trained without collision errors, it can successfully filter out the collision errors and guide correct rate selections. Moreover, the use of environmental information such as the signal strength and the relative distance can also reduce the impact of collision errors. Hence, it is expected that MTRA will behave robustly in high density networks. We conduct a simulation study in the next section to confirm this.

III. PERFORMANCE EVALUATION

A. *Simulation Settings*

In this section, we use ns-2 [9] simulations to study the performance of MTRA. We then compare it with both AARF [1] and SNR schemes. AARF is a popular rate adaptation algorithm for wireless LANs with adaptive success thresholds. We also implement the SNR method which can be treated as the quasi-optimal solution in this ideal environment since we assume the accurate channel information can be obtained at the transmitter. By comparing MTRA with these two algorithms, we can measure how much MTRA improves the existing algorithm and to which extent its performance is close to the optimal. The RTS/CTS exchange is disabled to reduce the communication overhead. It has been found that the Nakagami model [10] is well suited for outdoor propagation in vehicular environments and hence is utilized in our simulation study. The CBR traffic packet size is 1000 bytes. We use 802.11a parameters in the simulation study as they are readily available and very similar to 802.11p. Some of the parameters are summarized in Table 1. Vehicles can download or upload packets with the base station as well as communicate with its neighbors (other vehicles). As introduced below, we conduct simulations in different highway scenarios with various vehicle speeds and network densities; we also utilize a few metrics for performance evaluation.

Scenarios

1. In this scenario, there is a one-direction, single-lane highway with length 1 km. The base station is located (a) at the beginning, (b) in the middle, (c) and at the end of the highway. The speed of the vehicle is set at 20 m/s.

2. In this scenario, there is a bi-directional, four-lane

highway (1 km) with two lanes for each direction. The base station is located at the center of the highway. A total of *N* vehicles are active as either a client (downloading) or a server (uploading). The ratio of the upload/download connections is set to 0.3 to simulate real world scenarios where users download more than they upload. The speeds of the vehicles vary over the range [10, 20] m/s.

3. This scenario is similar to the second one with a bi-directional, two-lane highway except that there is no base station, and each vehicle communicates with another vehicle.

Table 1. Vehicular Network Simulation Parameters

CW_{min}	15	Frequency	5.9 GHz
CW_{max}	1023	Transmit Power	0.1 W
RxTresh	-95 dBm	Noise	-99 dBm
CSThresh	-96 dBm	Data Rate	6–54 Mbps
CPThresh	4 dB	CBR Packet Size	1000 bytes

Metrics

1. Average/aggregate number of TxSucc (Transmission Success) packets: this is the average/aggregate number of successfully transmitted packets per vehicle/of all vehicles.

2. Average transmission time: this is the average airtime required to deliver a packet, including retransmissions.

3. Rate distribution of transmitted packets: this metric calculates the number of packets that are sent at a specific data rate. It studies the algorithm's ability to exploit high data rates opportunistically.

B. *Effect of Environment*

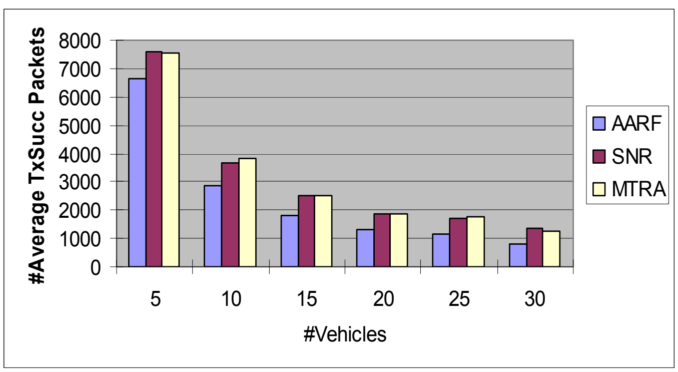

Fig. 3. Scenario 1(a): Average number of successfully transmitted packets when the base station is located at the beginning of the highway.

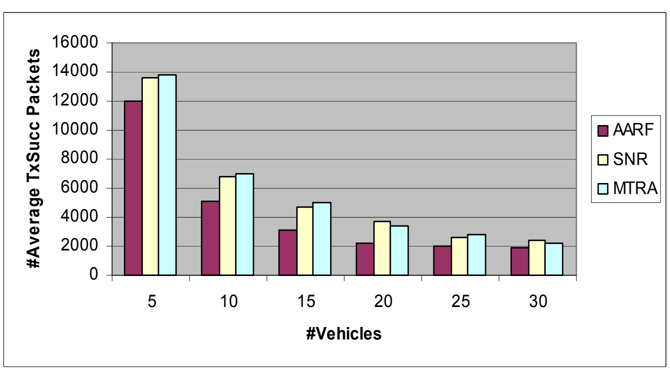

Fig. 4. Scenario 1(b): Average number of successfully transmitted packets when the base station is located in the middle of the highway.

In this set of simulations we use scenario 1 where we vary the location of the base station to get different propagation environments. As shown in Fig. 3, 4 and 5, the average number of transmitted packets (throughput) using MTRA is consistently higher than AARF (where the largest throughput improvement is 60% over AARF). Generally speaking, as the number of vehicles gets larger, the throughput gain is higher. This is due to the fact that in high-density networks, AARF falls back to lower data rates when collision errors occur. In contrast, because MTRA only utilizes the context information to predict the packet error probability, it is unaffected by the collision errors. Also note that the performance of MTRA is very close to the performance of the SNR method, which assumes perfect channel state information at the transmitter. To further study the time efficiency of the algorithms, in Fig. 6, we plot the average per-packet transmission time in Scenario 1(c) as an example. We can see that the average per-packet transmission time of MTRA is much lower than that of AARF, while it stays close to that of SNR. This is because MTRA largely reduces the number of retransmissions as it acts correctly even with collision errors. Falling back to lower rates in the case of collision errors not only reduces the throughput, but also leads to higher collision probability with extended per-packet transmission time.

C. *Hot-Spot Scenario*

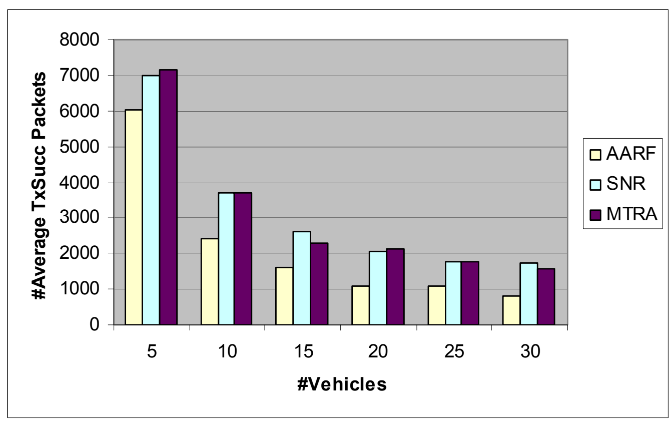

Fig. 5. Scenario 1(c): Average number of successfully transmitted packets when the base station is located at the end of the highway.

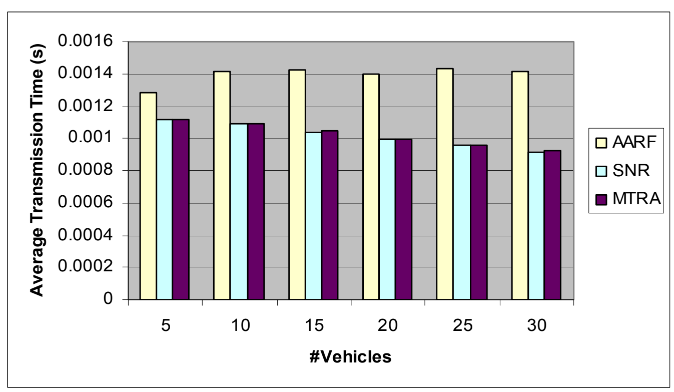

Fig. 6. Scenario 1(c): Average per-packet transmission time when the base station is located at the end of the highway.

In this simulation, we use Scenario 2 where the base station is placed at the center of the bi-directional, 4-lane 1 km highway. The vehicles may upload/download application data (CBR traffic) to/from the base station. The ratio of the upload/download connections is set to 0.3. The speeds of all vehicles are uniformly distributed over the range [10, 20] m/s. The vehicles start at either end of the highway and move to the other end, crossing the base station in the center. We plot the aggregate successfully-transmitted packets (as there are both uplink and downlink connections) varying the number of vehicles in Fig. 7. We observe that in all cases, MTRA transmits more successful packets than AARF. This can be better understood if we plot the rate distribution over all transmitted packets, as shown in Fig. 8. Clearly, MTRA and SNR methods manage to send most packets at the highest data rate 54 Mbps, while AARF sends most packets at lower data rates such as 24 and 12 Mbps.

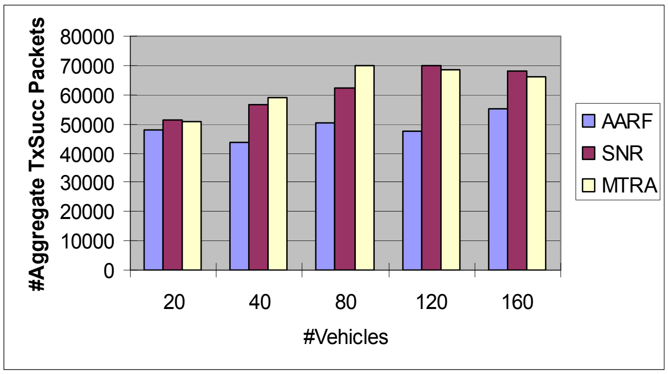

Fig. 7. Scenario 2: Aggregate number of successfully transmitted packets when the base station is located in the center of the highway.

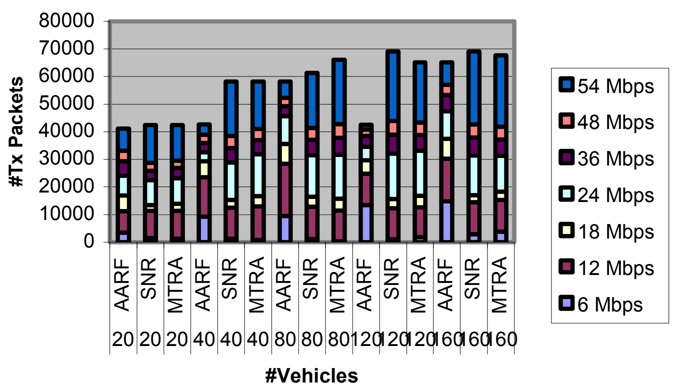

Fig. 8. Scenario 2: Rate distribution over all transmitted packets when base station is located in center of the highway.

D. *Ad Hoc Scenario*

Next we simulate an ad hoc scenario using a bi-directional, two-lane highway (Scenario 3). There are active vehicles of the same number in both directions: vehicles in one direction are set to be the servers and vehicles in the other direction are the clients. All vehicles select their speed uniformly over [10, 20] m/s. We show the aggregate number of successfully transmitted packets in Fig. 9. MTRA can better tune to the vehicular environment and sends more packets than AARF.

IV. CONCLUSION

In this paper, we first identify the challenges of rate adaptation for the vehicular networks which are more demanding than the traditional 802.11 wireless LANs. We then propose to use an efficient self-adaptive model-tree-based machine-learning method MTRA that can predict the packet error rate and adapt the data rate in real time. We also present a detailed methodology for the PER model tree building and update. The ns-2 simulation study under various scenarios shows that MTRA can achieve much better performance compared to AARF, and this performance is as close as the quasi-optimal SNR method. For future work, we will investigate other complex situations (e.g., changing weather conditions, high vehicle density, etc.). We also plan to test this scheme in a variety of real vehicular environments.

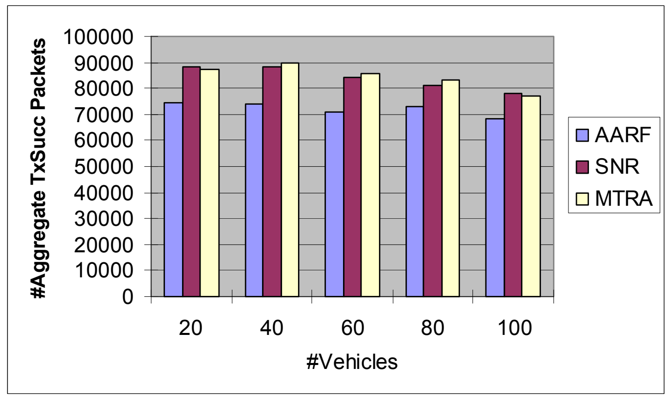

Fig. 9. Scenario 3: The number of successfully transmitted packets when vehicles communicate with each other.

REFERENCES

[1] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems (MSWiM)*, Venice, Italy, 2004.

[2] G. Holland, N. Vaidya, and P. Bahl, "A rate-adaptive MAC protocol for multi-Hop wireless networks," in *Proceedings of the 7th annual international conference on Mobile computing and networking (MobiCom)*, Rome, Italy, 2001.

[3] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *Proceedings of the 12th annual international conference on Mobile computing and networking (MobiCom)*, Los Angeles, CA, USA, 2006.

[4] D. Hadaller, S. Keshav, T. Brecht, and S. Agarwal, "Vehicular opportunistic communication under the microscope," in *Proceedings of the 5th ACM international conference on mobile systems, applications and services (MobiSys'07)*, San Juan, Puerto Rico, 2007.

[5] "Draft Standard Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 7: Wireless Access in Vehicular Environments (WAVE)," IEEE P802.11p/D3.0, 2007.

[6] J. R. Quinlan, "Learning with Continuous Classes," in *Proceedings of the 5th Australian Joint Conference on Artificial Intelligence*, Hobart, Tasmania, 1992, pp. 343-348.

[7] Y. Wang and I. H. Witten, "Inducing model trees for continuous classes," in *Proceedings of the 9th European Conference on Machine Learning (ECML'97)*, Prague, Czech, 1997.

[8] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications High-speed Physical Layer in the 5 GHz Band," IEEE Std 802.11a-1999(R2003), 1999.

[9] "NS-2: the network simulator," http://www.isi.edu/nsnam/ns.

[10] W. R. Braun and U. Dersch, "A physical mobile radio channel model," *IEEE Transactions on Vehicular Technology,* vol. 40, pp. 472-482, 1991.