Measuring the Capacity of In-Car to In-Car Vehicular Networks

Marcelo Gonçalves Rubinstein, Universidade do Estado do Rio de Janeiro
Fehmi Ben Abdesslem and Marcelo Dias de Amorim, CNRS and UPMC Université Paris 6
Sávio Rodrigues Cavalcanti, Rafael dos Santos Alves, Luís Henrique Maciel Kosmalski Costa, and Otto Carlos Muniz Bandeira Duarte, Universidade Federal do Rio de Janeiro
Miguel Elias Mitre Campista, Universidade Federal do Rio de Janeiro and Universidade Federal Fluminense

ABSTRACT

A particular class of vehicular networks is the one that includes off-the-shelf end-user equipment (e.g., laptops and PDAs) running from the interior of vehicles: in-car nodes. They are subject to limited communication conditions when compared with nodes specifically designed to this context. Existing works either consider antennas installed on top of the vehicle roof or nodes that operate in infrastructure mode. In this article, we investigate through real experiments the characteristics of links formed by in-car nodes running off-the-shelf wireless technologies such as IEEE 802.11(a/g) in ad hoc mode. We surprisingly observe that in-car nodes do show enough performance in terms of network capacity to be used in a number of applications, such as file transfer in peer-to-peer applications. Nonetheless, we identify some key performance issues and devise a number of configuration recommendations and future work directions.

INTRODUCTION

Intervehicular networks are one of the most significant and challenging modern communication systems [1]. Both academia and industry are extremely active in such a prolific area of research, and fundamental advances are expected to happen in the next years. The main reason for the success of vehicular networks as a research area is that the related applications have a direct impact on everyday life. In particular, we can cite safety [2], entertainment [3], or driving-assistance [4] applications.

Inter-vehicular communications can take place in two basic ways, either in pure ad hoc mode (vehicular ad hoc network [VANET]) or with the support of fixed nodes along the roadside (infrastructure mode). In the ad hoc case, vehicles communicate without any external support. In the infrastructure case, some typically static nodes are deployed along the roads in order to improve both connectivity and service provisioning. For more details on applications and categories of vehicular networks, the reader is invited to refer to the survey of Luo and Hubaux [5].

Perhaps the main issue vehicular networking faces today is adoption. Indeed, the basic assumption we have to make when conceiving vehicular networks is that vehicles are somehow equipped with some communication capability, in ad hoc or infrastructure modes. In particular, existing works (as we will see in the next section) assume either that vehicles have external antennas to improve connectivity or that one of the nodes operates in infrastructure mode. At least for the next five to ten years, this assumption will not be realistic.

We can assume that, at least for some period of time, there will be a particular configuration consisting of end-user equipments running as nodes inside vehicles. Users might want to use their laptops or PDAs inside their car as a node of the network without any additional equipment or hardware modification. Furthermore, such a possibility would be an additional feature for users that do not travel with their own vehicles (e.g., in taxis or buses). Joining a vehicular network so easily will make it much more popular. From a software point of view, there would be no problem for managing such nodes. The problem we investigate here is how these nodes would interact with the others from a connectivity point of view. This means that end-user equipments would be a particular class of nodes equipped with off-the-shelf wireless interfaces and operating from the inside of a metallic mass (the vehicle). These nodes, that we call *in-car* nodes, are then subject to different propagation conditions. To verify these different propagation conditions, we have measured received signal strength indication (RSSI) at different distances between two nodes using the Madwifi driver. We have used two configurations; one is the in-car node where two laptops inside the cars communicate, while in the other configuration the lap-

0163-6804/09/$25.00 © 2009 IEEE

tops are outside the cars. When the nodes are nearby (separated by 5 m), the RSSI is 11.1 in the in-car configuration against 16.1 for the free-space configuration; at 50 m, the average values are 2.9 and 10.8, respectively, and at 100 m, the values are 2.3 and 7.6. It is then important to understand whether in-car to in-car scenarios with off-the-shelf equipments provide enough communication resources to potential users.

A number of papers present measurement results that intend to evaluate the communication capabilities in networks of vehicles. Unfortunately, most of these works only consider scenarios with prepared vehicle, i.e., vehicles with external antennas or some other specific hardware. In this article, on the other hand, we focus on the in-car to in-car scenario. We perform a number of measurement tests in a real scenario in order to get the first insights and help the community understanding the constraints of these environments. We evaluate two variations of the de facto IEEE 802.11 standard (a and g) with both UDP and TCP and investigate the behavior of the system under different speeds and variable packet sizes.

Although the natural feeling would be to consider that in-car nodes are not adapted to vehicular networks, we make interesting observations:

- We show the feasibility of VANETs composed of off-the-shelf in-car nodes. We show that the data amount transferred during a contact of a few seconds is of the order of few Mbytes. Such an amount of data is enough for a variety of applications, including peer-to-peer applications (important share of the Internet traffic), safety applications such as emergency-break alerts, as well as delay-tolerant communications, to cite a few.
- For the same network interface, using IEEE 802.11g provides much better goodput than using IEEE 802.11a.
- When TCP is used, the instant at which the connection is requested is fundamental. A bad choice might result in degraded performance.
- When UDP is used, there is a clear relationship between packet size and goodput, depending on the car speed, and this relationship is not linear.

Based on our experience and our experimental tests, we also provide recommendations to deploy and enhance an ad hoc vehicular network with in-car nodes. To our knowledge, this is the first work that evaluates inter-vehicle communications in ad hoc mode and with no support of any infrastructure.

Related Work

A few papers performed measurement analysis of vehicle-to-infrastructure communications. Ott and Kutscher [6] use UDP and TCP to transfer data between a car equipped with an external antenna and a fixed station connected to an IEEE 802.11b access point. The tests were performed on a German freeway. They report that using an external antenna is mandatory in order to communicate with the access point in their scenario. They varied the car speed from 80 to 180 km/h. The authors seem to have performed a single run of each configuration (speed, packet size, and transport protocol). Results for UDP and 1250-byte packets show that throughput is low at large distances (more than 250 m from the access point) and reaches about 4Mb/s when in range of the access point irrespective of the speed. Moreover, 8.8 Mbytes could be transferred in a single pass. When using TCP, the throughput presents a significant amount of variability and is lower than with UDP. Cumulative data in a single pass reaches 6 Mbytes.

Gass *et al.* [7] use a scenario that is similar to Ott and Kutscher's one, but do not use external antennas in the car and perform measurements in the Californian desert, where interference due to other access points or cars is non-existent. Car speeds vary from 8 to 120 km/h. Each test was performed only twice. A stream of data consisting of UDP packets with sizes 50, 100, 200, 400, 800, and 1500 bytes was used in order to evaluate the effect of packet size. The results are presented as the average over the different packet sizes. TCP stream only used 1500-byte packets. Results show that the maximum average throughput is obtained when closer to the access point: 5.5 Mb/s for TCP and 3.5 Mb/s for UDP. The authors argue that such an unexpected behavior is due to the different UDP packet sizes used during tests. They have also shown that 92 Mbytes can be transferred when moving at 8 km/h and 6.5 Mbytes at 120 km/h.

Bychkovsky *et al.* [8] have used a completely different scenario. Nine cars belonging to people that work at MIT were used. Normal driving patterns of these people were observed for almost one year. Each car attempts to connect to open access points in the Boston area and to transfer data to a specific host. Maximum measured throughput of a TCP connection was about 700 kb/s. The maximum number of bytes transferred during one connection was about 8 Mbytes.

Hadaller *et al.* [9] also use a car equipped with an external antenna to communicate with an access point on a rural highway. All 15 experimental runs were performed with the car moving at 80 km/h and TCP traffic being sent from the access point to the car. Results show that the maximum average throughput near the access point is about 22 Mb/s and the maximum data transferred in a run achieves 51.1 Mbytes.

Wellens *et al.* [10] tested data transfers between cars as we do in this work, but using infrastructure mode (one of the cars has an access point, and the other one is a client), with 5 dBi gain external antennas fixed on the roofs. Measurements were performed in urban scenarios as well as in a highway. Results show that the goodput is mostly independent from the speed. The major impact factors are the distance between cars, the availability of line-of-sight, and the rate adaptation algorithm.

Our work is complementary to the aforementioned ones, as we focus on a different scenario and a specific class of nodes. As aforementioned, the scenario we investigate is more likely to happen in a nearer future; unfortunately, measurement campaigns on this specific case are still lacking.

Inter-vehicular communications can take place in two basic ways, either in pure ad hoc mode (vehicular ad hoc network [VANET]) or with the support of fixed nodes along the roadside (infrastructure mode).

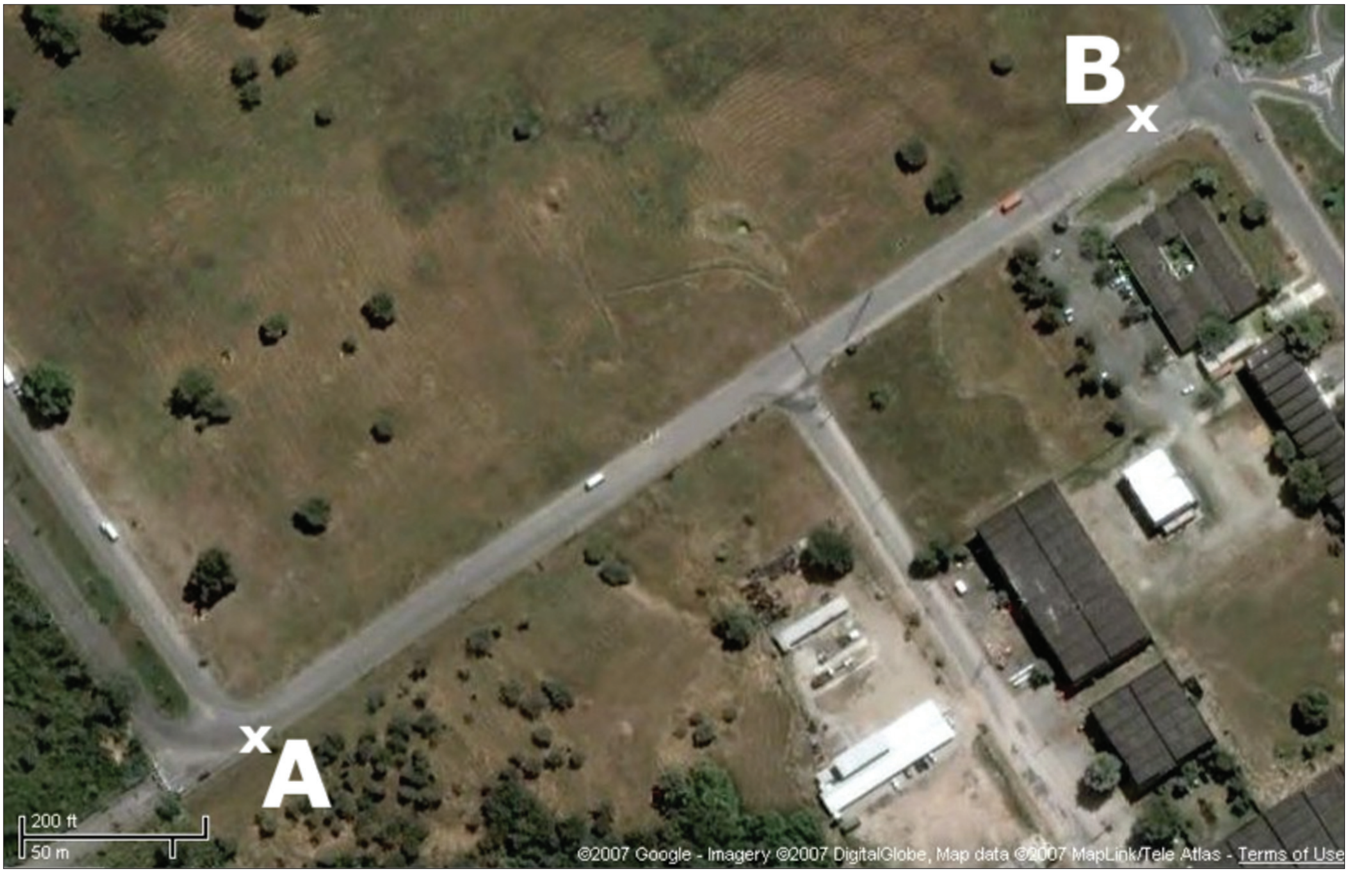
B
x
x
A
200 ft
50 m
©2007 Google - Imagery ©2007 DigitalGlobe, Map data ©2007 MapLink/Tele Atlas - Terms of Use

Figure 1. *Satellite view of the 400-m-long street used for the experiment. Vehicles have two different start points (noted A and B); these points are far away enough so that nodes are not within the communication range of each other.*

Measuring the Capacity of In-Car to In-Car Links using Off-the-Shelf Wireless Cards

As stated before, our vehicular network does not rely on any infrastructure, i.e., its nodes operate in ad hoc mode. Without loss of generality, we consider an application of file transfer; the quality of the network will then be evaluated as the amount of data that can be transferred within the opportunistic link that is formed when two moving vehicles encounter. We underline again that we only use off-the-shelf equipments and available device drivers and software. This is essential as we consider near-term in-car to in-car vehicular networks.

Measurement Setup

Our testbed is composed of IBM T42 Laptops, using Linksys WPC55AG (IEEE 802.11 a/b/g) CardBus interfaces, based on the Atheros chipset, and u-blox EVK-5H GPS devices attached via USB. Laptops are held on the lap of the passenger and no external antenna is used. We use the Linux operating system with kernel version 2.6.22-2-686 and Madwifi driver version 0.9.3.3. We tuned a few parameters of the default bitrate selection algorithm used in Madwifi, called SampleRate, following [9]. In order to send back-to-back data that is enough to saturate the network (bulk data), we use Iperf version 2.0.2. Sending rate is set to 30 Mb/s for UDP.

We set beforehand some simple parameters to avoid any extra delays due to configuration. We fixed the IP addresses of the laptops and set the MAC addresses in the configuration file of ARP to avoid requests. We also fixed both the ESSID and channel at which the network operated. Note that these parameters can be configured by the application running on the user's node (and thus with no intervention from the user herself). Also note that by previously setting a number of parameters, we intend to obtain results that are closer to the optimum achievable conditions under the off-the-shelf scenario.

As mentioned before, an in-car to in-car VANET is well suited for P2P applications. Moreover, taking a P2P application as an example, our measurement setup with preconfigured parameters can be easily implemented. P2P applications are characterized by the user's interest in a specific content (rather than a specific server). Therefore, there is no need for the client peer to know the IP address of the content-provider peer. The content request may be broadcast. As for the client's IP, a preconfigured address based on the MAC address or on the license plate number can be used to avoid DHCP. Also, in that case the identification of the wireless network is less important; therefore, using a preconfigured BSSID would be fine.[1] Once the two vehicles enter in contact, a single packet may be sufficient for the client to request a file from the peer. Given those assumptions, the data transfer phase of a contact can be maximized.

[1] *Such a strategy was adopted by the IEEE 802.11p working group, which decided to strongly simplify the setup procedures in the upcoming standard, allowing the use of a wildcard BSSID to avoid lengthy association and authentication procedures.*

The testbed was deployed over a straight and 400m-long street of the Universidade Federal do Rio de Janeiro (UFRJ) campus, under light traffic. Figure 1 shows a satellite view of the street as well as two points *A* and *B* that indicate the points at which the cars start moving towards each other. This represents a sparse topology, which is expected to happen quite frequently in reality [11]. In order to be sure that our experiments were performed with little external influence, we conducted a mapping of existing access points that could interfere with our results. No other IEEE 802.11 networks operate in this area, except some with weak signals on other channels that were detected near point B (next to the building). We underline however that this segment is out of range of the crossing point where data were transferred. At their starting points, the cars are out of range. We varied both car speeds from 20 km/h to 60 km/h, ranging the relative speed from 40 km/h to 120 km/h. One of the cars serves as an Iperf server, whereas the other car is the client. Iperf client sends bulk data (in TCP and UDP) to the Iperf server, and the amount of data received is obtained by running tcpdump on the server car. GPS devices record the position of both cars four times per second, which is the highest frequency of measurements supported by our GPS devices.

Laptops were synchronized using Coordinated Universal Time (UTC) obtained from GPS and cars start moving at the same time. At start, a script launches Iperf at the server car. Iperf is also run at the client, but its instantiation time depends on the transport protocol. When using UDP, Iperf is immediately run at the client when cars begin to move. We tried to use the same approach with TCP, but obtained no successful reception at the server, because of TCP connection timeouts. To avoid this issue, we modified the client script to send ping packets and to start Iperf when first ping response is received. Both cars move on the right side of the road at the same speed and cross approximately in the middle of the street.

We performed our experiments using IEEE 802.11 technology, in both a and g modes. We will present most of the results with IEEE 802.11g because, as we will see later in this section, for our setup, IEEE 802.11a leads to poor performance — the contact time was shorter and the amount of data successfully transferred much smaller than the one obtained with IEEE 802.11g. For a similar reason, we perform more analyses using UDP as the transport protocol. Indeed, UDP is more appropriate for lossy links, and we believe it will play the main role in future vehicular networks. TCP reduces the bit rate because of its window-based packet loss recovery mechanism, and is often blamed for applying network congestion algorithms when the link is lossy. We argue that most applications for vehicular networks can handle packet loss without TCP. For instance, exchanging a file can be done using UFTP [12], an FTP application running over UDP.

We also varied other parameters when performing the measures, namely the speed and the packet size. When the speed of a car increases, the contact time decreases and then the total amount of data transferred also decreases. Regarding packet size, small packets are less exposed to data errors and have more chances to arrive with a correct checksum than larger packets.

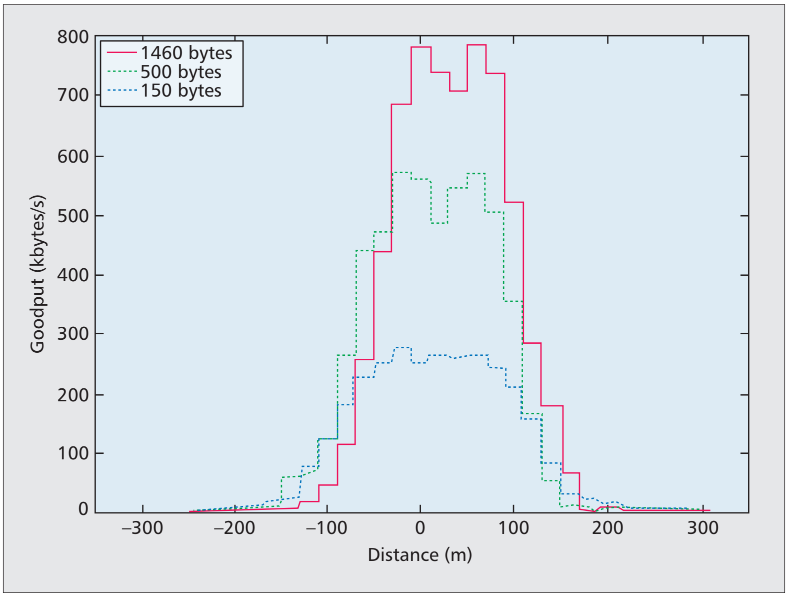

Figure 2. *Average goodput over UDP using IEEE 802.11g between cars moving at 20 km/h.*

Parameter	Value
IP address	Fixed
ARP	Manual
ESSID	Fixed
Channel	Fixed
Wireless technology	IEEE 802.11a/g
Transport protocol	UDP/TCP
Speed	20/40/60 km/h
Packet size	150/500/1460 bytes

Table 1. *Main measurement setup parameters.*

We performed a total of 150 runs during a couple of days. Each configuration (IEEE 802.11 standard, speed, and packet size) was tested 10 times, to get more accurate results, and for each experiment we show the average values of those 10 runs.

Main measurement setup parameters are summarized in Table 1.

Measurement Results

We now present the measurement results we obtained during the experimentation. For the purposes of this article, we decided to focus on the data transferred during a contact between the two vehicles. We define the contact time between two vehicles as the time interval

Speed	Packet size (in bytes)	Transferred data (in Mbytes)	Contact time (in seconds)	Goodput (in Mb/s)	Error rate
20 km/h	150	6.2 (σ = 0.87)	45.17 (σ = 6.73)	1.11 (σ = 0.14)	0.06 (σ = 0.01)
	500	10.7 (σ = 2.35)	41.30 (σ = 5.09)	2.08 (σ = 0.38)	0.05 (σ = 0.02)
	1460	13.0 (σ = 2.58)	38.48 (σ = 6.67)	2.75 (σ = 0.55)	0.06 (σ = 0.01)
40 km/h	150	1.7 (σ = 0.35)	16.46 (σ = 2.43)	0.84 (σ = 0.13)	0.15 (σ = 0.03)
	500	3.3 (σ = 1.12)	15.57 (σ = 3.33)	1.72 (σ = 0.47)	0.10 (σ = 0.05)
	1460	3.6 (σ = 2.16)	14.90 (σ = 3.66)	1.90 (σ = 0.87)	0.15 (σ = 0.09)
60 km/h	150	1.5 (σ = 0.32)	11.72 (σ = 2.40)	1.08 (σ = 0.22)	0.13 (σ = 0.03)
	500	2.7 (σ = 1.23)	11.51 (σ = 2.30)	1.81 (σ = 0.69)	0.10 (σ = 0.05)
	1460	1.6 (σ = 1.39)	10.83 (σ = 2.78)	1.15 (σ = 0.90)	0.25 (σ = 0.18)

Table 2. *Average amount of data transferred, contact time, and goodput for car-to-car transfers over UDP and IEEE 802.11g.*

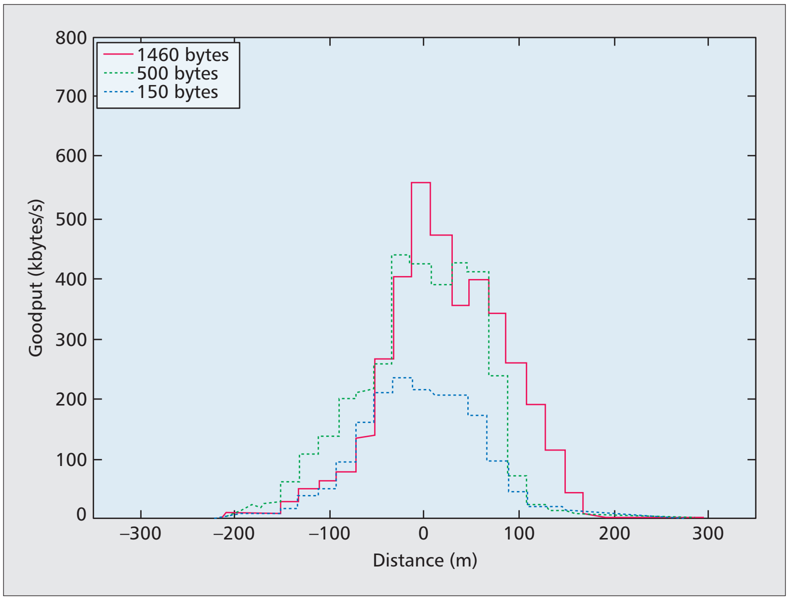

Figure 3. *Average goodput over UDP using IEEE 802.11g between cars moving at 40 km/h.*

between the first packet correctly received and the last one. This parameter alone is significant for a large plethora of applications. The average goodput is plotted versus distance between cars. Negative values refer to cars approaching and positive values indicate cars moving away from each other. We define goodput as the application available throughput; that is, the number of bits per unit of time excluding protocol overhead (headers) and retransmission packets.

Figure 2 shows the average goodput of data received by the car running the Iperf server, when both cars are moving at 20 km/h. As shown in Table 2, the average contact time is about 42 seconds. Within the contact time, as we can observe in Table 2 and in the graphs, larger packets allow transferring greater amounts of data because of the smaller protocol overhead. The peak goodput is of 6.4 Mb/s, obtained with 1460-byte packets when the cars are side by side.

When we increase the speed of the cars to 40 km/h, setting the packet size to 1460 bytes still allows to transfer more data during the crossing. Nevertheless, as shown in Fig. 3, the peak goodput is around 4.8 Mb/s. The difference on the amount of data transferred for 500-and 1460-byte packets is less important than with 20 km/h. Besides contact time, which is about half the one for 20 km/h, we observe a trade-off between car speed and packet size. This trade-off becomes clearer in our tests for 60 km/h.

We also performed the same experience with cars moving at 60 km/h. The results are shown in Fig. 4. As we can see, our observations are confirmed, with some variations. Table 2 shows the total amount of data transferred between the two vehicles to have further insight into the capacity of the links formed in the three scenarios. For each value, the table also indicates the standard deviation (denoted by σ). On the one hand, as expected, higher speeds produce shorter contact times and smaller amounts of data transferred. On the other hand, we observe that the higher the speed, the lighter the impact of the packet size on the amount of data transferred. In the case cars move at 60 km/h, we can even observe a slight decrease in this value: 2.7 Mbytes were transferred using 500-byte packets, whereas only 1.6 Mbytes of data were transferred using 1460-byte packets. These measurements indicate a trade-off between packet size and speed. To better understand this trade-off, we computed the average goodput and average packet error rate for each speed and packet size. Goodput is calculated by dividing the amount of data transferred by the contact time. Packet error rate is computed by comparing tcpdump traces obtained at the server and at the client.

As shown in Table 2, for 150- and 500-byte packets, goodput and packet error rates are similar for the same packet size and different speeds. Nevertheless, for larger 1460-byte packets, the goodput decreases whereas the packet error rate increases. This effect can be explained by variations of channel conditions due to fading, which are faster for higher vehicle speeds. Channel estimation is done only at the beginning of the packet. The higher the packet size, the higher the variations of channel conditions within the transmission of a single packet, as pointed out in [13]. Thus, the higher the speed, the higher the impact of fading over the amount of data transferred. This is the main reason why we observe a drop in the goodput. Nevertheless, the reduction of the amount of data transferred when speed increases is mainly due to a shorter contact time.

We now compare the performance of TCP against UDP. Using TCP instead of UDP reduces the average total amount of data transferred (Fig. 5a). At 40 km/h, we were able to transfer only 1.5 Mbytes using TCP and 500-byte packets, whereas UDP was able to transfer 3.3 Mbytes with 500-byte packets at the same speed. Table 3 shows that TCP performs worse than UDP. For example, at 60 km/h, for 4 runs out of 10, no data were received at all. As a result, the average amount of data received per run is very small, around 80 kbytes.

We performed some tests using the same dual-mode wireless cards operating on IEEE 802.11a. Because of its higher frequency (over 5.15 GHz), the overall transmission range of IEEE 802.11a is shorter than IEEE 802.11g. In addition, transmissions using higher frequencies are more prone to propagation problems, such as diffraction, reflection, and absorption. In our experiments, the result of this property is a shorter contact time between the cars, as shown in Fig. 5b. We can then see that TCP leads to poor performance when compared with UDP, and shows to be a bad choice of transport protocols in such a scenario. Even for UDP, the performance over IEEE 802.11a is quite limited: 3.1 Mbytes transferred at 20 km/h, 1.6 Mbytes at 40 km/h, and only 0.8 Mbytes transferred at 60 km/h.

According to our experiments, the capacity of in-car to in-car links is sufficient to transfer amounts of data of a few Mbytes. Such a result is complementary to existing works that focus on more specific conditions, once no special equipment was used and the scenario is pure ad hoc. We confirm that two cars crossing can run typical peer-to-peer file sharing applications and have enough bandwidth to exchange documents, short videos or MP3 files; and this only with off-the-shelf hardware, and despite the small contact time.

We show that in-car to in-car networking is possible even though stated differently by Ott and Kutscher in [6]. Ott and Kutscher fixed ESSID and IP address in their experiments, similar to our setup. Nevertheless, their different scenario and other parameters that were not optimized, such as the fixed wireless channel and modified bit rate selection algorithm, lead to poor performance without external antennas. Using our setup on the other hand, we could

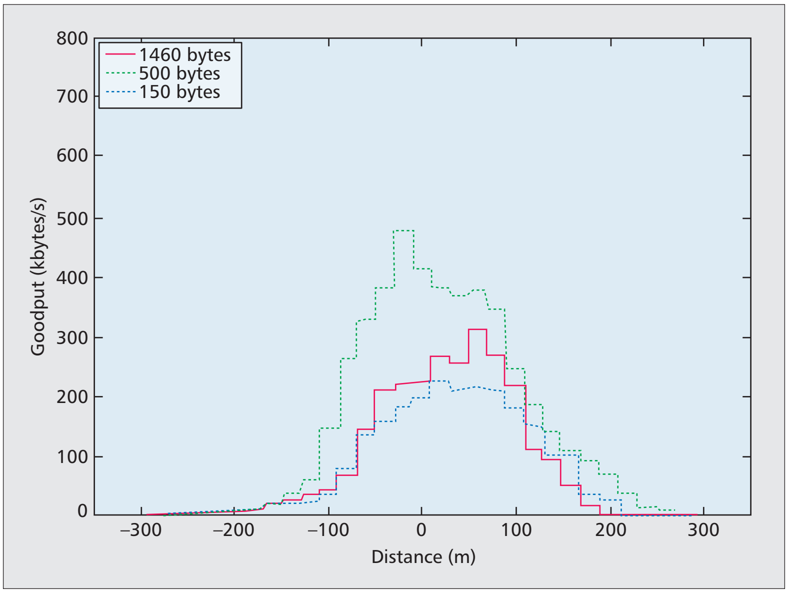

Figure 4. *Average goodput over UDP using IEEE 802.11g between cars moving at 60 km/h.*

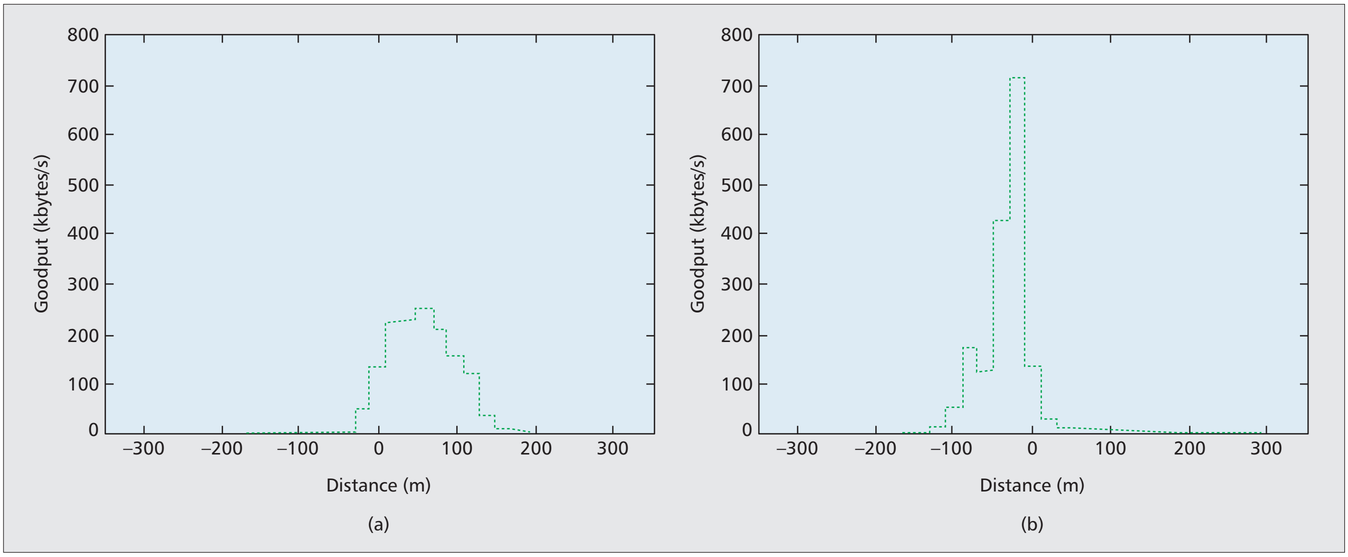

Figure 5. *Average goodput for cars moving at 40 km/h and using 500-byte packets: a) TCP over IEEE 802.11g; b) UDP over IEEE 802.11a.*

We confirmed that the car speed is indeed directly related to the transfer capacity. We also found that decreasing the packet size reduces the capacity loss due to speed, and can even increase this capacity.

Speed	Transferred data
20 km/h	3.9 Mbytes, σ = 2.53 Mbytes
40 km/h	1.5 Mbytes, σ = 0.62 Mbytes
60 km/h	0.08 Mbytes, σ = 0.10 Mbytes

Table 3. *Average total amount of data transferred from car to car over TCP and with 500-byte packets.*

achieve reasonable goodput with no external antenna.

Our results show low TCP performance in in-car to in-car environment. This is also reported by Bychkovsky *et al*. [8]. In their tests, they do not fix ESSID, IP address, and wireless channel. Nevertheless, similar to our setup, they also use in-car nodes, but in their case, to communicate with access points. Hadaller *et al*. [9] observed better TCP goodput compared with Bychkovsky *et al*.'s work. In Hadaller *et al*., the authors use external antennas and they fix the IP address. Besides some improvements on the default bit rate selection algorithm, the main difference to Bychkovsky *et al*. and our setup is the use of external antennas.

Ott and Kutscher, Bychkovsky *et al*., and Hadaller *et al*. have improved TCP performance by checking if the network is available before trying the connection. As we confirmed in our work, the time TCP connections are established should be carefully chosen to avoid underutilization of network resources. Moreover, the goodput of TCP is low because packet losses are higher and the congestion window remains small, especially at the beginning and at the end of the contact.

Discussion and Recommendations

Network capacity depends on a number of parameters, as measured in our experiments. We confirmed that the car speed is indeed directly related to the transfer capacity. We also found that decreasing the packet size reduces the capacity loss due to speed, and can even increase this capacity. Considering the same dual-mode off-the-shelf hardware, IEEE 802.11g is more suitable for in-car to in-car networking than IEEE 802.11a, because of its greater range. UDP is also more appropriate than TCP for our context because it reaches higher goodput in presence of lossy links. Another reason is that TCP spends time during connection establishment. In practice, waiting for the destination car to be in range requires regularly checking its presence running a script sending pings, for instance.

Based on our experience and on the related work [6, 9], we recommend applying beforehand simple parameters to avoid extra delays:

- Fixing the IP addresses to avoid DHCP delays (that would have been done through a multi-hop path or an AP infrastructure)
- Fixing the ESSID and the channel (frequency) to avoid scanning delays
- For networks with limited and known users, setting the MAC addresses of all other users on each device to avoid ARP requests

Nevertheless, fixing these parameters beforehand is simple if the network is small, or if some default values are defined a priori (e.g., by a central authority). In a larger scope, some parameters are hard to be set beforehand. In this case, we give more general recommendations that have the same goal of the aforementioned ones, but can be used in most cases:

- To maximize the contact time, for example, by increasing the transmission power.
- To avoid management issues. Handling management issues today may require infrastructure access points, at least for bootstrapping. However, in the next years, we expect some parameters to be set dynamically without central management (e.g., ESSID and channel settings, rate-control algorithms).
- To avoid addressing issues, for example, by using identifiers such as the car's license plate, associating MAC and IP addresses, broadcasting addresses, or using here again an infrastructure in a first step.
- To avoid routing, for example, by determining the path in advance, if the trajectories of the vehicles are known. We can also limit communications to nodes in range. In this case, an important open issue would be to evaluate the frequency at which trajectories are known in advance (for example when drivers type their destination in their GPS devices).

Software designers must also keep in mind these recommendations before creating new applications, or adapting legacy ones for in-car to in-car networking scenarios. Concerning current well-known applications, Table 4 summarizes basic characteristics and the suitability of each one for in-car to in-car networking. Distributed applications are adapted for this scenario, because they are in accordance with the architecture of the network itself. Note that Delay Tolerant Network (DTN)-like applications can handle high delays and frequent disconnections [14]. These applications highly suit in-car to in-car environments, since mobile networks are prone to frequent link breakages.

According to our experiments, we also recommend implementing or using applications that handle packet losses by themselves, as done by UFTP, instead of relying on TCP mechanisms. This allows using UDP as a transport protocol. Applications such as P2P, security, and assistance already suit these requirements. With our hardware, we also recommend using IEEE 802.11g because of its higher range for in-car to in-car networking. On the other hand, IEEE 802.11a frequency band is closer to IEEE 802.11p one — to be released in 2010 for vehicular networks. Nevertheless, IEEE 802.11p is specifically designed for vehicular networks, using narrower channels to compensate for the increased delay spread and improved transmission masks [15]. Finally, an adaptation algorithm should be used to optimize the in-car to in-car goodput. This algorithm should reduce or increase the packet size dynamically according to the car speed. This could be done for instance

Applications	Characteristics				Suitability
	Required contact time	Transport protocol	Amount of traffic	Type of architecture	
P2P	Long	TCP or UDP	Large	Distributed	Suitable
Security	Short	UDP	Small	Distributed	Suitable
Assistance	Short	UDP	Small	Distributed	Suitable
DTN-like	Variable	TCP or UDP	Variable	Distributed	Suitable
FTP	Long	TCP or UDP	Large	Client-server	Suitable
HTTP	Variable	TCP	Variable	Client-server	Not yet suitable
Email	Variable	TCP	Variable	Client-server	Not yet suitable
Voice	Variable	UDP	Variable	Client-server	Not yet suitable
DNS	Short	UDP	Small	Client-server	Not yet suitable

Table 4. *Application characteristics and suitability for in-car to in-car scenarios.*

by a sublayer between the physical layer and higher levels to adapt the packet size. Different layers could communicate to adapt this size, from the application to the physical layer.

Following these simple recommendations, with off-the-shelf hardware and available drivers and software, should help increasing the capacity of in-car to in-car vehicular networks.

CONCLUSION

In this article, we presented a number of experimental evaluations of the capacity of in-car to in-car communications. We performed a number of tests in the context of a simple scenario composed of two cars. Our analysis allowed us to derive a number of recommendations that can help users improving the performance of their applications and setting suitable parameters. It is important to note that the results we show here serve as a benchmark for future analysis of richer topologies running equipments similar to ours.

While our results allow understanding the basic properties of communications over in-car to in-car links, there are more questions ahead before vehicular networking can be fully characterized. One of them is the analysis of denser scenarios subject to interferences. It would also be interesting to run real applications on the nodes to derive realistic expectations users would have from such networks. We also plan to perform more tests with multi-hop communications in order to generalize our results to other applications of vehicular networks (both in terms of combined contact time and end-to-end goodput). Additionally, we also plan to experiment with IEEE 802.11p devices as soon as they become available.

ACKNOWLEDGMENT

The authors would like to thank CNPq, CAPES, COFECUB, Faperj, and FUJB for partially funding this research.

REFERENCES

[1] R. Morris *et al.*, “CarNet: A Scalable Ad Hoc Wireless Network System,” *ACM SIGOPS Euro. Wksp.*, Kolding, Denmark, Sept. 2000.

[2] W. Chen and S. Cai, “Ad Hoc Peer-To-Peer Network Architecture for Vehicle Safety Communications,” *IEEE Commun. Mag.*, vol. 43, no. 4, Apr. 2005, pp. 100–7.

[3] A. Nandan *et al.*, “Co-Operative Downloading in Vehicular Ad-Hoc Wireless Networks,” *IEEE WONS*, St. Moritz, Switzerland, Jan. 2005.

[4] L. Wischhof *et al.*, “SOTIS: A Self-Organizing Traffic Information System,” *IEEE VTC*, Jeju, Korea, Apr. 2003, pp. 2442–46.

[5] J. Luo and J.-P. Hubaux, “A Survey of Inter-Vehicle Communication,” *Tech. Rep. no. IC/2004/24*, School of Comp. Commun. Sci., EPFL, Switzerland, 2004.

[6] J. Ott and D. Kutscher, “Drive-thru Internet: IEEE 802.11b for “Automobile” users,” *IEEE INFOCOM*, Hong Kong, Mar. 2004.

[7] R. Gass, J. Scott, and C. Diot, “Measurements of In-Motion 802.11 Networking,” *IEEE WMCSA*, Apr. 2006, pp. 69–74.

[8] V. Bychkovsky *et al.*, “A Measurement Study of Vehicular Internet Access using in Situ Wi-Fi Networks,” *ACM MobiCom*, Los Angeles, CA, 2006, pp. 50–61.

[9] D. Hadaller *et al.*, “Vehicular Opportunistic Communication under the Microscope,” *ACM MobiSys*, San Juan, Puerto Rico, June 2007, pp. 206–19.

[10] M. Wellens, B. Westphal, and P. Mähönen, “Performance Evaluation of IEEE 802.11-based WLANs in Vehicular Scenarios,” *IEEE VTC*, Dublin, Ireland, Apr. 2007, pp. 1167–71.

[11] M. Zhang and R. S. Wolff, “Routing Protocols for Vehicular Ad Hoc Networks in Rural Areas,” *IEEE Commun. Mag.*, vol. 46, no. 11, Nov. 2008, pp. 126–31.

[12] D. Bush, “UFTP-UDP based FTP with Multicast”; http://www.tcnj.edu/~bush/uftp.html.

[13] J. Yin *et al.*, “Performance Evaluation of Safety Applications over DSRC Vehicular Ad Hoc Networks,” *ACM VANET*, 2004.

[14] K. Fall, “A Delay-Tolerant Network Architecture for Challenged Internets,” *ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003, pp. 27–34.

[15] D. Jiang and L. Delgrossi, “IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments,” *IEEE VTC-Spring*, May 2008, pp. 2036–40.

BIOGRAPHIES

MARCELO G. RUBINSTEIN [S’95, M’03] received his B.Sc. degree in electronics engineering, and M.Sc. and D.Sc. degrees in electrical engineering from Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1994, 1996, and 2001, respectively. From January to September 2000 he was at the PRiSM Laboratory, University of Versailles, France. He is now an associate professor with Universidade do Estado do Rio de Janeiro (UERJ). His major interests are in wireless networks, home networking, medium access control, and quality of service.

FEHMI BEN ABDESSLEM received his M.Sc. and Ph.D. degrees in computer science from the University of Paris 6, France, in 2005 and 2008. He is now a postdoctoral fellow at the School of Computer Science of the University of St Andrews, United Kingdom. His current research interests include designing wireless protocols and algorithms with an engineering approach, and building experimental testbeds for vehicular ad hoc networks, sensor networks, and wireless mesh networks.

SÁVIO R. CAVALCANTI received his B.Sc. degree in statistics from Escola Nacional de Ciências Estatísticas, Rio de Janeiro, Brazil, in 1994 and his M. Sc. degree in electrical engineering from UFRJ in 2008. He is currently with the Escola de Guerra Naval (EGN), part of the Brazilian Navy. His major research interests are vehicular networks, routing protocols, and network security.

MIGUEL ELIAS M. CAMPISTA [S'05] received his telecommunications engineering degree from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2003, and M.Sc. and D.Sc degrees in electrical engineering from UFRJ in 2005 and 2008, respectively. Since 2009 he has been an associate professor with the Telecommunications Engineering Department of UFF. His major research interests are in multihop wireless networks, quality of service, wireless routing, and home networking.

RAFAEL DOS S. ALVES is an undergraduate student of computer and information engineering at UFRJ, Brazil. Since 2006, he is a member of a research mentorship program in Electrical Engineering Program of COPPE, also in UFRJ. His academic interests include vehicular networks, wireless mesh networks, and future Internet.

LUÍS HENRIQUE M. K. COSTA [M'99] (luish@gta.ufrj.br) received his electronic engineer and M.Sc. degrees in electrical engineering from UFRJ in 1997 and 1998, respectively, and a D.Sc. degree from the Université Pierre et Marie Curie (Paris 6), Paris, France, in 2001. Since August 2004 he has been an associate professor with COPPE/UFRJ. His major research interests are in the areas of routing, wireless networks, and group communications. He has been a member of the Association for Computing Machinery since 2001.

MARCELO D. DE AMORIM [S'00, M'01] is a full-time research scientist at the French National Center for Scientific Research (CNRS) and a member of the LIP6 laboratory of the Université Pierre et Marie Curie-Paris 6. He holds a B.Sc. in electronics engineering and a M.Sc. in electrical engineering, both from Universidade Federal do Rio de Janeiro (UFRJ), and a Ph.D. from the University of Versailles, France. His research interests focus on self-organizing networks.

Otto Carlos M. B. Duarte received electronic engineer and M.Sc. degrees from UFRJ in 1976 and 1981, respectively, and a Dr.Ing. degree from ENST/Paris, France, in 1985. Since 1978 he has been a professor with UFRJ. From January 1992 to June 1993 he was with MASI Laboratory, Université Paris 6. In 1995 he spent three months with the International Computer Science Institute (ICSI), University of California, Berkeley. In 1999, 2001, and 2006 he was an invited professor at Université Paris 6. His major research interests are in multicast, QoS guarantees, security, and mobile communications.