A Proposal for Modelling Piggybacking on Beacons in VANETs

Klein Wolterink, W., Heijenk, G, and Karagiannis, G.

Department of Computer Science, University of Twente, The Netherlands
{w.kleinwolterink, geert.heijenk, g.karagiannis}@utwente.nl

Abstract. Piggybacking on beacons is a forwarding technique that is regularly used in vehicular ad-hoc network (VANET) research as a means to disseminate data. With this technique data is attached to and transmitted along with scheduled beacons, without changing the timing of the beacons.

The performance of piggybacking largely depends on network parameters such as the network density, the beaconing frequency, etc. It is our goal to model the performance of piggybacking as a function of such parameters. In this paper we present our methodology to achieve this goal, and show some first conclusions w.r.t. which network parameters should be taken into account in our model.

Keywords: beaconing, dissemination, piggybacking, VANET

1 Introduction

Vehicular networking can be considered as one of the most important enabling technologies for Intelligent Transportation Systems (ITS). Vehicular networking is concerned with communication between vehicles and infrastructural devices, supporting a multitude of traffic applications.

Traffic applications can typically be categorized either as safety applications or efficiency applications. A typical example of the former is the ‘Emergency electronic brake lights’ use case [1] in which a vehicle sends out a high-priority warning to all nearby vehicles. It is critical that such a warning is disseminated fast ($<$ 100 ms) to all relevant (i.e., nearby) vehicles – messages will therefore be disseminated with an increased priority. Non-delivery of a message can cause less safe situations. The distances involved are limited and can be covered by at most a few transmission hops. In contrast to this, traffic efficiency messages are typically targeted at a larger geographical region and may have a lifetime of tens of seconds. Non-delivery of a message can cause less efficient behaviour (e.g., increased travel time) but will not cause dangerous situations. A typical example is the ‘Decentralized floating car data’ use case, see [1].

The issue of disseminating safety messages has so far received a lot of attention, leading to a large number of (mainly) flooding-based solutions. When applied to disseminating efficiency messages however these solutions are far from optimal. For this reason attention has been shifting to more delay-tolerant dissemination strategies for the delivery of efficiency messages. One such strategy is disseminating messages by attaching them to network-level beacons. We refer to this technique as *dissemination by beaconing* or *piggybacking*.

With piggybacking forwarding of packets is only done by attaching them to scheduled beacons. Since the scheduling of beacons is a far from trivial problem [2], it seems preferable that the piggybacking process should not influence the beacon scheduling, thus keeping the timing of the beacons unchanged. Forwarding by piggybacking is therefore relatively slow when for instance compared to a flooding strategy. As was already noted in [3] the speed with which information is disseminated depends amongst others on the beaconing frequency, the network density, and the transmission distance.

A main *expected* advantage of piggybacking is that the impact piggybacking has on the network load should be considerably less when compared to other forwarding strategies:

- Since the packets are attached to already scheduled beacons, network- and security overhead for every transmitted packet can be saved. Together this overhead may be more than 200 bytes [4][5].
- Additionally one access to the wireless medium per packet is avoided, thus reducing contention and the risk of network collisions.

Based on our own experiences in [3] and reported results by others (see the discussion on related work in [10]) our ongoing research focuses on piggybacking.

Consider the following scenario: there exists a source node S and a destination node D, the latter which is located d meters from S. S transmits a packet at $\tau = 0$ which is forwarded by means of piggybacking to D. Our problem statement then becomes: What is the probability that D will have received the packet within τ seconds for a given set of network parameters? Examples of network parameters are the network density, the transmission power, the beaconing frequency, etc.

It is our goal to create a model that is able to predict the probability that destination node D will have received the packet within τ seconds, as a function of the set of network parameters. In this paper we present our methodology to create such a model. We also show some results of a first analysis we have performed on the impact of a number of network parameters on piggybacking in a static network.

The outline of this paper is as follows. In Section 2 we introduce our methodology. In Section 3 we present the results of a simulation study on piggybacking in a static network. We conclude the paper in Section 4.

2 Methodology

Piggybacking may be implemented in a number of ways, and it is impossible to create a single model that is able to capture the behavioural details of every possible implementation. In Section 2.1 we describe how we model the behaviour of a piggyback protocol in a generic manner, and list the assumptions that our model contains. In Section 2.2 we present our methodology.

2.1 Forwarding model

Any dissemination protocol in an intermittently connected VANET must employ two different forwarding strategies, depending on the state of the network [3] [6] [7] [8]:

1. As long as a node is able to forward the data to a node that is closer to the destination the data will actively be forwarded in that direction.
2. When an intermediate forwarder is not able to forward the data to a node that is closer to the destination, then the *store-carry-forward* mechanism is used: the data is locally stored by a node that moves in the direction of the destination. This node will carry the data until it finds a node that is closer to the destination, and will then forward the data to this node, at which point the first strategy is again applied.

The above two-step approach also holds for piggybacking. Thus, the time it takes to piggyback data from the source to the destination depends on:

1. $d_{forwarding}$ – The forwarding delay: the time it takes to actively forward data a certain distance.
2. $d_{carrying}$ – The carrying delay: the time it takes for a node to carry the data to a node that is closer to the destination.

In our model we treat these two delays independently. Work on calculating $d_{carrying}$ can be found in [9]. Our work initially focuses on $d_{forwarding}$; later on we will combine the two delays into a single model.

We now state two assumptions that are meant to ease the modelling of $d_{forwarding}$ for piggybacking, while we argue that it will not diminish the applicability of our model. The first assumption is that both source and destination are situated on the same stretch of road. The second assumption is that this stretch of road is straight.

The first assumption is rather unrealistic of course, but since the node topology in a VANET is by definition limited to the road network every possible route between a source and a destination can be broken down into a limited set of stretches of road. For each of these stretches our assumption holds.

Although the second assumption is incorrect as well, we do not expect it to have a significant impact on the model. In non-urban situations curves in a road are rather gradual, while inside an urban area sharp curves often imply a new stretch of road, or can be modelled as such.

2.2 Methodology

As has already been stated, our work focuses on the forwarding delay. Our methodology to calculate $d_{forwarding}$ consists of the following steps:

1. First we simulate the performance of a piggybacking protocol for a range of network parameters.

2. We then analyse the impact each network parameter has on the performance of the piggybacking protocol.
3. Based on this analysis we model the performance of the piggybacking protocol based on those network parameters that were found to have a significant impact.

To ease the modelling of the impact of the network parameters we initially ignore mobility. Later on we will repeat the three steps with mobility taken into account. We expect the effect of mobility on the forwarding delay to be limited, since the speeds with which vehicles move are typically not significant w.r.t. the transmission ranges and beacon frequencies involved.

Currently we have performed the first two steps for a static network. We discuss the results of these steps in Section 3.

3 Simulation Study of a Static Network

In this section we describe the set-up and discuss the results of a simulation study that investigated the performance of piggybacking in a static network. This study was part of the first two steps of our methodology, see Section 2.2. The goal of this study was to answer the following research questions:

1. Which network parameters should be taken into account to express the probability that a packet has been piggybacked a certain distance within a certain time interval?
2. How significant is the impact of each network parameter on the performance of the piggybacking (specified below), and in what way do the parameters affect performance?
3. Can the effect of some network parameters be combined into a single parameter?

In Section 3.1 we describe the set-up of the experiment, the parameters that have been varied during simulation, and the performance metrics that have been measured. In Section 3.2 we discuss the results.

Due to space limitations the description of the experiment and the discussion of the results are limited and incomplete – for a complete discussion of our simulation study see [10].

3.1 Experimental Set-up

The piggyback protocol that we have used in our experiments is relatively simple, such that we are better able to judge the impact of the network parameters. It is similar to the protocol we have used earlier [3] which despite its simplicity proved to be quite effective. Once we have fully modelled the performance of this protocol as a function of network parameters, we expect that it should take considerably less effort to model more involved protocols.

It is assumed that nodes know their own geographical position. The forwarding rules for every node are as follows. At $\tau = 0$ the source node piggybacks a service data unit (SDU) that has a geographical destination region attached to it. When a node receives a beacon that contains an SDU, it will encapsulate this SDU in its next scheduled beacon if by that time all of the following (still) hold:

1. The node is not the source of the SDU.
2. The node has not received the SDU from another node that is located closer to the destination region.
3. The node has not included the SDU in a previous beacon.

Once the SDU reaches a node that is inside the destination region the forwarding stops.

Different scenarios have been created by varying the following parameters: the distance over which a packet must be piggybacked (the *dissemination distance*), the average distance between nodes (the *inter-node distance*), the transmission power, the transmission bit rate, the beacon frequency, the size of the beacon window (not explained here as it was found to have no significant impact on performance), the size of a beacon, and the size of the SDU.

For each experiment the following performance metrics have been measured:

1. $d_{forwarding}$ – The forwarding delay: the time it takes to forward the SDU to the destination region
2. P_{pr} – The probability that the SDU reaches the destination region. It may be that the SDU is lost during piggybacking because of transmission failures.
3. $P_{pr}(\tau)$ – The measured probability whether the SDU has reached the destination region within τ seconds. I.e., if we have measured that 80% of all SDUs reached the destination within 10 s, we state that $P_{pr}(10) = 0.8$.

3.2 Results Analysis

The three network parameters that have the biggest impact on performance are the dissemination distance, the inter-node distance, the transmission power, the transmission bit rate, and the beacon frequency.

An increase in the dissemination distance has the obvious effect of increasing the network delay. An increase of the dissemination distance gives a linear increase in $d_{forwarding}$ and a linear decrease in P_{pr}. If the dissemination distance and the inter-node distance are changed but their reciprocal ratio remains the same, $P_{pr}(\tau)$ (and thus the other two performance metrics as well) will remain largely unchanged. E.g., if the dissemination distance is doubled and the inter-node distance is halved, you will get the same results.

The reception probability for a single hop is mainly determined by the transmission power and the transmission bit rate used. If different combinations of these two network parameters result in the same single hop packet reception probability, $d_{forwarding}$, P_{pr}, and $P_{pr}(\tau)$ for the multi-hop case are also the same for these combinations. We can therefore combine the network parameters transmission power and transmission bit rate into a single network parameter: the single hop reception probability.

The main effect of increasing the beacon frequency is an exponential decay in $d_{forwarding}$. The main effect of increasing the mean inter-node distance is a linear increase of the network delay and a decrease of the packet reception probability. The main effect of increasing the beacon size is a linear increase in the network delay.

Increasing the transmission power, transmission bit rate, the beacon frequency, and the beacon size will lead to an increased network load, as will a decrease of the inter-beacon distance. As the network load is increased a point will be reached where increasing the load further leads to an increase in the amount of unsuccessful transmissions. An increase of the network load beyond this point will lead to an increase in $d_{forwarding}$ and a decrease of P_{pr}. E.g., as the beacon frequency is increased beyond this point, P_{pr} decreases linearly.

The effect of the size of the SDU on performance is significant but negligible compared to the effect of the network parameters mentioned previously. The size of the beacon window has no impact on performance.

4 Conclusions & Future Work

Piggybacking is a method to disseminate data by attaching it to already scheduled beacons. It is our goal to model the performance of piggybacking as a function of relevant network parameters. Our main performance metric, and the outcome of our model, is the probability that a destination node has received the data within τ seconds.

In this paper we have presented our methodology to create such a model and the specific steps involved. We have also shown how the delay of piggybacking data can be broken down into two parts: the forwarding delay when actively forwarding data from node to node, and the carrying delay when data is carried by a node to nodes that are closer to the destination. Our research focuses on the forwarding delay.

In Section 3 we have discussed some results of a simulation study on piggybacking inside a static network. The following network parameters should at least be taken into account when modelling piggybacking: the dissemination distance, the average distance between nodes, the transmission power, the transmission bit rate, and the beacon frequency. Although the beacon size and the size of the SDU also impact performance significantly, their effect is an order of magnitude less.

Our next steps will be to model the piggyback performance for the static case. We will then simulate, analyse and model the impact of mobility on performance. Finally we will combine our model to calculate $d_{forwarding}$ with (an) existing model(s) to calculate $d_{carrying}$, into a model that is able to calculate the complete delay to piggyback information from a source node to a destination region.

References

1. "Intelligent transport systems (its) – vehicular communications – basic set of applications – definitions," European Telecommunications Standards Institute, Technical Report 102 638, 2009.
2. M. van Eenennaam, W. Klein Wolterink, G. Karagiannis, and G. Heijenk, "Exploring the solution space of beaconing in vanets," in *Vehicular Networking Conference (VNC), 2009 IEEE*. IEEE, pp. 1–8.
3. W. Klein Wolterink, G. Heijenk, and G. Karagiannis, "Dissemination protocols to support cooperative adaptive cruise control (cacc) merging," in *International Conference on ITS Telecommunications (to appear)*, 2011.
4. M. Raya and J. Hubaux, "Securing vehicular ad hoc networks," *Journal of Computer Security*, vol. 15, no. 1, pp. 39–68, 2007.
5. C. Project, "D31 european its communication architecture – overall framework – proof of concept implementation," Information Society Technologies, Tech. Rep., 2008.
6. C. Sommer, R. German, and F. Dressler, "Adaptive beaconing for delay-sensitive and congestion-aware traffic information systems," *University of Erlangen, Dept. of Computer Science, Technical Report CS-2010-01*, 2010.
7. L. Wischhof, A. Ebner, and H. Rohling, "Information dissemination in self-organizing intervehicle networks," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 6, no. 1, pp. 90–101, 2005.
8. O. Tonguz, N. Wisitpongphan, and F. Bai, "Dv-cast: a distributed vehicular broadcast protocol for vehicular ad hoc networks," *Wireless Communications, IEEE*, vol. 17, no. 2, pp. 47–57, 2010.
9. S. Yousefi, E. Altman, R. El-Azouzi, and M. Fathy, "Analytical model for connectivity in vehicular ad hoc networks," *Vehicular Technology, IEEE Transactions on*, vol. 57, no. 6, pp. 3341–3356, 2008.
10. W. Klein Wolterink, G. Heijenk, and G. Karagiannis, "Information dissemination in vanets by piggybacking on beacons an analysis of the impact of network parameters," in *IEEE Vehicular Networking Conference (VNC) 2011 – To appear*. IEEE, 2011.