Effects of Transmission Queue Size, Buffer and Scheduling Mechanisms on the IEEE 802.11p Beaconing Performance

Luuk Hendriks
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
l.hendriks@student.utwente.nl

ABSTRACT

The IEEE 802.11p Wireless Access in Vehicular Environment (WAVE) protocol is specified to be used in a Vehicular Ad-Hoc Network, or VANET. This paper provides an analysis of the 802.11p performance when used for the exchange of beacons. Beacons are used to enable vehicles to establish a cooperative awareness from which many vehicular applications can draw their inputs. In particular, this paper focuses on the impact of transmission queue size, transmission buffer and scheduling mechanisms on the IEEE 802.11p beaconing performance. Based on these investigations is concluded that a queue with a length of 5 beacon sizes is sufficient to be used for 802.11p beaconing. Moreover, the beaconing performance is not influenced by the type of the buffering and scheduling mechanism used. A new proposed buffering mechanism denoted as oldest packet drop, ensures that up to date information is disseminated without negatively impacting the beaconing performance.

Keywords

beaconing, 802.11p, C-ACC, performance, EDCA, Oldest Packet Drop

1. INTRODUCTION

The exchange of information is everywhere around us in modern day life, and vehicle-to-vehicle communication is an uprising phenomenon. This type of communication is supported by a Vehicular Ad-Hoc Network, or VANET. In particular, a VANET is a wireless ad-hoc network that supports the communication (1) amongst vehicles and (2) between vehicles and Road Side Units (RSUs). The main goal of VANETs is to support Intelligent Transportation System (ITS) applications which aim at providing entertainment, traffic efficiency and traffic safety [21].

The latter ITS application being naturally more time-critical, and therefore inherently benefit from better performance. The IEEE 802.11p protocol is the protocol standard approach used for information dissemination in a VANET [9][21].

15th Twente Student Conference on IT June 20th, 2011, Enschede, The Netherlands.

Wireless communication between vehicles is essential to provide information exchange. A way to disseminate this information is beaconing. Beacons are short messages sent periodically and used to enable vehicles to establish a cooperative awareness from which many vehicular applications can draw their inputs. A traffic efficiency application that can be used to increase passenger comfort and reduce traffic jams is the Cooperative Adaptive Cruise Control (C-ACC) [20].

Using the beaconing feature of VANETs, vehicles (and if required RSUs) cooperate to obtain the necessary lead vehicle dynamics information (such as speed, position, and acceleration) and a general view of traffic ahead. By using this information the performance of the currently available Adaptive Cruise Control (ACC) system is enhanced. The system can control the accelerator, the engine powertrain and the vehicle brakes in order to maintain a desired time-gap or time headway to the vehicle ahead [11]. It is clear that in this situation beaconing reliability and delay are critical factors in order to achieve a trustworthy safety aspect.

The IEEE 802.11p uses a Media Access Control (MAC) protocol that is based on a Carrier Sense Multiple Access protocol with Collision Avoidance (CSMA/CA). Due to the fact that the channel capacity is limited and beacons may be sent several times per second [9, 16] and vehicular density can vary greatly (and become large during traffic jams), it is expected that the channel may become congested. This will result in a deterioration of the Cooperative Awareness and hence the performance of the ITS applications. In particular, when a node tries to access the medium and finds the channel busy, it chooses a random backoff time from the interval [0, CW] and delays the medium access for the duration of that backoff. The parameter CW represents the size of the Contention Window. If no acknowledgement is received (e.g. a collision occurs) the CW size is doubled and the process starts over. However, beaconing uses the IEEE 802.11p common broadcast channel [9, 22], which does not support beacon retransmissions nor acknowledgements on sent beacons. Therefore, when beacons are sent the CW is never increased and beacons are not retransmitted. It is therefore, needed to use a beaconing mechanism that is well performing.

Beaconing is used to enable vehicles to establish a cooperative awareness from which many vehicular applications can draw their inputs. The cooperative awareness should be up to date. Therefore the information that is disseminated using beaconing is only relevant and useful when it is not too old. This paper defines a new buffering mechanism that can be used in situations of transmission queue overflow to discard oldest packets instead of discarding the newest packets that are buffered into the queue.

In this paper we focus on the impact of the transmission queue size, transmission buffering and scheduling mechanisms on the IEEE 802.11p beaconing performance. In particular, this paper focuses on the buffering and scheduling mechanisms that are specified for IEEE 802.11p, as well as the new buffering mechanism that can be used to discard oldest packets first in situations of transmission queue overflow.

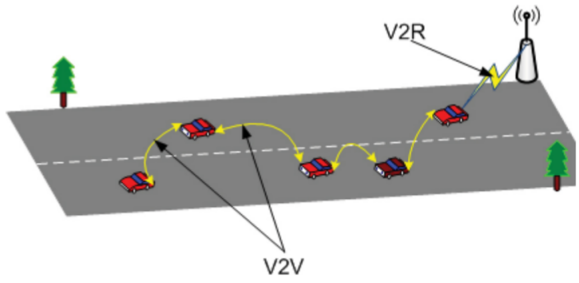

Figure 1. VANET: v2v and v2i elements, copied from [13]

This paper describes a simulation experiment of a Vehicular Ad-Hoc Network (VANET), see figure 1, where beaconing is used. Note that in this field of research the terms *v2v* and *v2i* are often used, meaning respectively vehicle-to-vehicle and vehicle-to-infrastructure. The experiment scenarios can be seen as VANETs solely based on v2v elements. By using different combinations of queue sizes and buffer mechanisms we will determine their performance, expressed in measures described in section 4.2.

This leads to the main research question to be answered in this paper: What is the impact of different transmission queue sizes and buffer mechanisms on the IEEE 802.11p beaconing performance?

The answer to this question will be formed based on six sub-questions:

1. How is beaconing performed in IEEE 802.11p based vehicular networks?
   Answered in section 2

2. Which transmission queue sizes, transmission buffering and scheduling mechanisms are applied in 802.11p based vehicular networks?
   Answered in section 3

3. Which transmission buffering and scheduling mechanisms can be applied to ensure that cooperative awareness in a vehicular network is up to date?
   Answered in section 3

4. Which measures can be used to quantify the IEEE 802.11p beaconing performance?
   Answered in section 4.2

5. Which simulation experiments can be performed in order to analyse the performance of IEEE 802.11p beaconing?
   Answered in section 4

6. Have the transmission queue size and buffer mechanism a significant impact on the performance of IEEE 802.11p beaconing?
   Answered in sections 5.1, 6

In order to answer the research questions the following method has been applied. For the first two and the fourth sub questions a small literary study was accomplished. The third research question has been answered by developing a new buffering mechanism that is denoted in this paper as Oldest Packet Drop (OPD). The answers to the first four research questions provided the information needed to answer the fifth research question. The sixth sub question was answered by running multiple simulation experiments in OMNET++. These simulation experiments had different parameters and were run several times to get statistically significant results. The result of each simulation experiment was logged in a database. These databases were then queried using scripts and analysed.

The remainder of the paper is divided as follows: Section 2 briefly provides the related work in the area of IEEE 802.11p buffering and scheduling and introduces beaconing and discusses how its performance can be improved. In section 3 different IEEE 802.11p buffering and scheduling mechanisms are described, including the new OPD buffering mechanism. In section 4 the simulation experiments that were run are discussed and in section 5.1 the results of these simulation experiments are analysed. Section 6 concludes and provides recommendations for future activities.

2. BEACONING IN 802.11P

The IEEE 802.11p protocol is a variant of the IEEE 802.11 protocol [9], enabling Wireless Access in Vehicular Environments (WAVE). Much research has been done in the field of 802.11p IEEE 802.11e performance. It is important to note that the IEEE 802.11p protocol specification [9] is recommending to use IEEE 802.11e [8] for QoS differentiation at the MAC layer. The IEEE 802.11e MAC standard amendment [8] used to support QoS differentiation at the MAC layer of IEEE 802.11. IEEE 802.11e can be used to solve three main challenges: (1) the handling time-varying network conditions, (2) adapting to varying application profiles and (3) managing link layer resources. In IEEE 802.11e two modes of operation are defined: (1) the HCCA (Hybrid Coordination Control Access) and (2) the Enhanced Distributed Channel Access (EDCA). These methods are used to enhance the MAC functionalities specified in IEEE 802.11a/b/g/p.

The Enhanced Distributed Channel Access (EDCA) can classify traffic through the use of access categories (ACs). In [8], 4 ACs are defined. Differentiation in access priority between each AC is accomplished by setting different values for the channel access parameters, see figure 2.

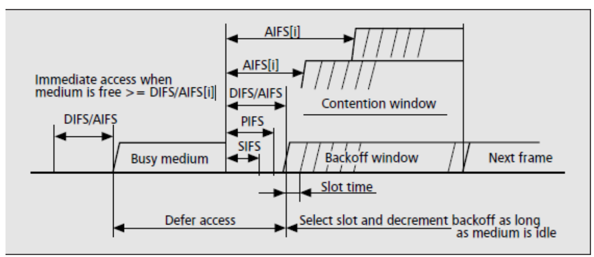

Figure 2. EDCA channel access prioritisation, copied from [15]

The channel access parameters used for this differentiation are: (1) Arbitrary Interface Space Number (AIFSN), (2) Contention Window (CWmin and CWmax) and (3)

Transmission OPportunity (TXOP) limit. A description of how this is accomplished is given in [8, 15].

Eichler evaluated the performance of 802.11p using the standard EDCA parameters in a quantitative way [5]. The optimal configuration for a 802.11e enhanced EDCA is researched in [1]. On the subject of queue lengths, [19] observed and proved that a reduction of the MAC buffer size has a positive effect in terms of throughput, using a standard DCF scenario. In [21] a analytical model is presented and used to compute the probability of successful reception and mean transmission delay in a 802.11p beaconing environment. Using Markov chains, [6] models the EDCA mechanism and analyses delay and throughput for varying traffic densities. A model for performance analysis of 802.11e EDCA is proposed in [7]. A comparison between 802.11p EDCA and the commercially applied STDMA (Self-Organizing Time Division Multiple Access) is given in [3], based on simulations of a highway scenario. In [22] an adaptive mechanism to perform broadcasting in 802.11 is proposed, supported by simulation studies claiming to be performing better in terms of reception rate and channel utilisation. None of the above mentioned papers has investigated the impact of the combination of transmission queue size, transmission buffering and scheduling mechanisms on the beaconing performance. This investigation is the main research goal of this paper.

In beaconing, the 802.11p is used to send short messages with a regular frequency. For a C-ACC application a frequency of 10Hz is sufficient [20]. This results in a maximum latency of 100ms, to prevent the sending node from filling the air with information that could be outdated before the receiving node is able to use it. The size of beacon messages can be small, for example 20 bytes as used in [17] in a simulation of a one-dimensional single-lane road with moving vehicles. However, when beaconing is applied in a C-ACC scenario, a message size of 400 bytes is approximated in [20]. This includes necessary security requirements (certificate and signature) as well as the actual information about the sending vehicle. The 802.11p standard specifies a maximum frame body size of 2304 bytes, which obviously is sufficient for a C-ACC scenario.

Propagation of beaconing messages can be accomplished in two ways: one-hop or multi-hop. In a one-hop scenario a node does not forward anything it receives. When using multi-hop, a receiving node will broadcast the received information again (making it actually to propagate through the network as opposed to the single-hop method). Note that this introduces a trade-off: the retransmission (in a broadcasting fashion) enables the information to reach vehicles further away from the initially sending node. But due to that same retransmission, the IEEE 802.11p common broadcasting channel will be used by more frames than using single-hop. As the saturation of the IEEE 802.11p common broadcasting channel affects the performance of it, using multi-hop is not a better option per se. The propagation solution space of beaconing in VANETs has been investigated extensively, see e.g., [20].

3. TRANSMISSION QUEUE SIZE, TRANSMISSION BUFFERING AND SCHEDULING

As already mentioned in the previous sections, the beaconing performance can be influenced by the length of the transmission queue and the buffering and scheduling mechanisms used at the transmission side of the IEEE 802.11p protocol. This section describes these parameters in more detail.

3.1 Queue length

The queue length or buffer length is a number describing how many elements/packets can be queued before dropping occurs. A large queue length is able to hold more elements/packets, but this has its reflection in the physical implementation as a larger memory module is required. Note that the length of the queue can affect performance in multiple ways. A larger queue can store more elements/packets and therefore, in average, less elements/packets will be dropped. However, if a an element/packet is stored in a queue for a too long time then the information that is included in such an element/packet can become outdated. Based on this rationale, research has been done to evaluate the performance of beaconing using very small queue lengths [19]. Similarly, in this paper we mainly investigated small queue sizes, starting from a queue with a length of 1 element/packet up to a length of 30 elements/packets. Furthermore, this paper focused only on the impact of the queue that is used at the transmission side of the IEEE 802.11p protocol.

3.2 Buffering or Queuing mechanisms

A buffering or queuing mechanism is the procedure of how an arriving element/packet, e.g. a beacon, is handled (queued) depending on the state of the queue. Dropping a packet that arrives at a full queue is a simple example of a buffering mechanism, called Tail-Drop or Drop-Tail [4].

Random Early Detection (RED) is a well-used buffering mechanism in the Internet. RED drops packets with a probability based on how full the queue is when a new packet arrives. An empty queue will accept the packet with a high probability. The more the queue is filled, the higher the dropping probability of a packet becomes[4]. The current IEEE 802.11p OMNET++ MiXiM simulation models use the Drop-Tail buffering mechanisms, see section 3.5. In order to minimise the situation that a packet is stored in a queue for a too long time, a novel buffering mechanism, denoted as the Oldest Packet Drop (ODP), is introduced in section 3.5. Both buffering mechanisms will be investigated in this paper.

3.3 Scheduling disciplines

A scheduling discipline describes how a queued element leaves the queue. Two well-known scheduling disciplines are: First-in, First-out (FiFo) and Last-in, First-out (LiFo). A FiFo scheduler will send out the elements in the order in which they arrived. Opposed to that, a LiFo scheduler will send out the most recent arrived element. Another often applied scheduling discipline is Priority Based Scheduling [18], which sends out elements based their classification. The current IEEE 802.11p OMNET++ MiXiM simulation models use the FiFo scheduling discipline. In order to investigate how the FiFo and LiFo scheduling disciplines interact with the Drop-Tail and OPD buffering, the LiFo scheduling mechanism is also considered in the accomplished simulation experiments.

3.4 EDCA

As described in the previous sections, the IEEE 802.11e Enhanced Distributed Channel Access (EDCA) [8] can classify traffic through the use of 4 access categories (ACs) by setting different values for the channel access parameters. The IEEE 802.11p and EDCA uses a Media Access Control (MAC) protocol that is based on a Carrier Sense Multiple Access protocol with Collision Avoidance

(CSMA/CA). The four classes are numbered AC0 through AC3, the former having the lowest priority. For each class one queue is used, see figure 3, and three parameters are used to determine the waiting time. These parameters are CWmin and CWmax, describing the boundaries of the Contention Window, and the Arbitration Inter-Frame Space (AIFS), the time the channel should be sensed free before the node starts sending data. Average waiting times computed using the parameters range from 264 μs for AC0 to 56 μs for the AC3 classified messages. In the case of the channel being busy while sensing (or becomes busy during the AIFS duration), the *backoff* mechanism starts: based on the contention window a random number is drawn from the interval [0, CW] (distributed uniformly) which is decremented every *slot time* (depends on the PHY) the channel is free. The message is send as soon as the value reaches 0.

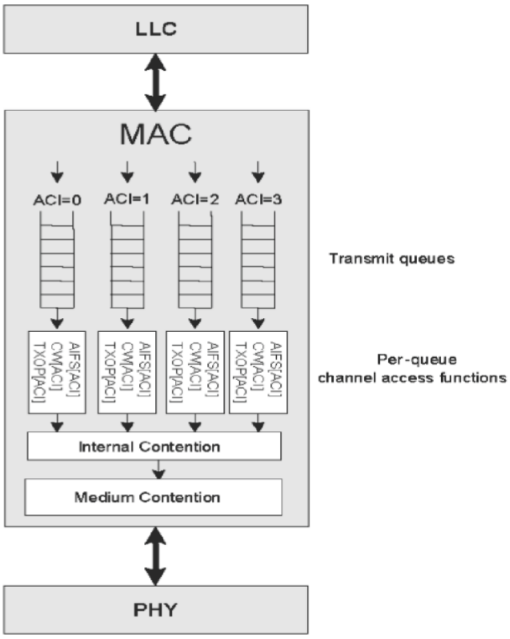

Figure 3. EDCA buffers, copied from [2]

3.5 Newest Packet Drop (NPD), Oldest Packet Drop (OPD)

The current EDCA implementation in OMNET++ and MiXiM [14, 12] uses the FiFo scheduling discipline and the Drop-Tail buffering mechanism. One of the research goals of this paper is to investigate combinations of buffering and scheduling mechanisms that can be used to ensure that queued packets are not queued for a too long time and become outdated.

Therefore, combinations of two buffering mechanisms and two scheduling mechanisms are used. The two scheduling mechanisms are the FiFo and the LiFo scheduling disciplines. The two buffering mechanisms are: *Newest Packet Drop* (NPD) and *Oldest Packet Drop* (OPD). When a packet arrives at a full queue, NPD will ensure that the packet containing the newest information is dropped. Note that this is the well known Tail-Drop buffering mechanism, but due to the fact that these buffering mechanisms will be used in combination with either FiFo or LiFo, the terminology *head* or *tail* are avoided in order to prevent confusion. The combination of FiFo and NPD can be considered to be the 'default EDCA'.

When applied in e.g. C-ACC, NPD seems to be unsuitable. Newer information is dropped in favour of old information, in a scenario where old information might already be useless, and the newer, discarded information be critical.

OPD addresses this problem by acting in the exact opposite way. A packet arriving at a full queue will not be discarded, as it is the newest information available. OPD will drop the queued packet that contains the oldest information, creating space for the arriving packet. The new packet will be inserted in the queue and be processed using the scheduling discipline in place. This is illustrated in figure 4 for both FiFo and LiFo scenarios.

It is important to note that when a contention occurs and the transmission queue is not empty, then a packet that is selected for transmission will be stored in a contention queue. However, the IEEE 802.11p OMNET++ and MiXiM simulation models do not use a separate queue for contented packets, but they are queued in the transmission queue. This means that when a queue with a length of 1 is used and a new packet arrives at the transmission queue, then (depending on the scheduling discipline and buffer mechanism), the packet ready to be sent after contention can be the dropping candidate. To prevent this, a packet in contention is protected from being dropped out of the queue. In that case, the second packet up for transmission will be dropped to create space for the arriving packet.

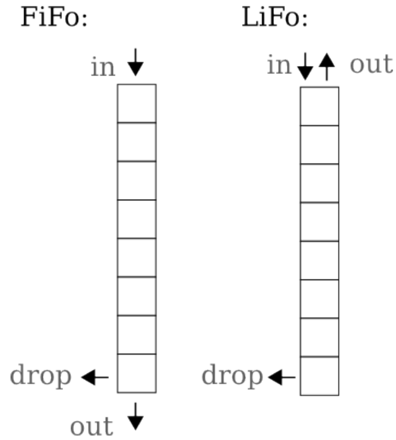

Figure 4. OPD applied with FiFo and LiFo

4. SIMULATION EXPERIMENTS

4.1 Simulation environment

The simulation experiments are performed using OMNeT++ [14], a discrete time event-based simulator primarily used for the simulation of networking scenarios. MiXiM [12] is a framework based on OMNeT++ and provides functionality for simulation of wireless networks.

In our simulation configuration, all nodes are within each others reach, and there is no chance of loss due to propagation. Thus, we have a scenario without hidden terminals and the only possibility of loss after transmission is collision loss. Hidden terminals identify a situation where a communication between two vehicles say vehicle A and vehicle B is disturbed by one or more vehicles located between these two communicating vehicles. We are mainly interested in a form of beacon loss that occurs before transmission: loss due to queue drops, which is also used as a performance measure. Beacons are generated at a rate of 25Hz, see [22]. The data rate, for all nodes, is 3Mbit/s. Furthermore all 802.11p properties are used as specified in [9]. The number of nodes varies per simulation, resulting in scenarios with varying traffic density and thus varying network utilisations. The vehicle densities range from 20 nodes per 100m, to 160 nodes per 100m. Every scenario is simulated multiple times (using the method of independent replications, [10]).

When performing simulations of a system one needs to take in account that a certain number of samples gathered from the beginning of a simulation run are not to be used. In that start-up period, the system is in the so-called transient state [10] which is most-likely not stable and therefore not representative for the rest of the simulation run. There are several methods to handle these unwanted samples: [10] lists six. For this experiment, we determined the duration of the transient phase, and chose a simulation time that will be long enough to make the unwanted samples insignificant for the results. This resulted in a simulation time of 400 seconds. The number of beacons to be sent is chosen to be very high, so the simulation time becomes the limiting factor of a simulation run.

For every set of parameters, four scenarios are tested. These four scenarios are combinations of two scheduling disciplines, FiFo and LiFo, and the two buffer mechanisms, Oldest Packet Drop and Newest Packet Drop. Note that entering the queue is not a guarantee for being transmitted.

To simulate the behaviour on different link utilisations, the runs are performed using a varying vehicle density. Note that all vehicles are generating beacons with a frequency of 25Hz. In these simulation experiments also the channel utilisation is measured and mapped to the vehicle densities. The channel utilisation is measured by computing the ratio of total transmit and receive time on the transmission channel, divided by the total simulation time. This division will not reach 100%. This is due to the fact that the use of inter-frame gaps in 802.11p, see figure 2, cause that the maximum calculated channel utilisation reaches only 87%. Therefore, all channel utilisation values listed in the simulation results and analysis of the experiments, see Section 5, are normalised such that the maximum utilisation value can reach 100%. This means that a measured and calculated channel utilisation value of 87% will map to a normalised value of 100% channel utilisation.

4.2 Performance measures

There are multiple options to measure the performance of beaconing. This sections provides a list measures we will use while analysing the simulation results, briefly describing each of them.

4.2.1 Queuing delay

The time a beacon is stored and remains in the queue and is not in contention state. A contention state starts at the moment that the beacon is selected for transmission but due to the fact that the transmission channel is busy it has to wait until the transmission channel becomes idle. A contention state ends at the moment the beacon is broadcasted. This time is measured from the moment a beacon enters the queue until the moment this beacon is in contention state.

4.2.2 Contention delay

The time that a packet changes from a queuing state to a contention state until the actual broadcasting of the beacon.

Table 1. Simulation parameters

Beacon generation rate λ_g	25Hz
Beacon size	400 bytes
CW_{min}	15
Data rate	3Mbit/s
Queuing mechanism	EDCA
Simulated time limit	400s

4.2.3 Dropping probability

The number of beacons dropped out of the queue (thus not broadcasted) at $node_i$, divided by the number of beacons generated by $node_i$.

4.2.4 Update delay / Inter-arrival time

The time, as experienced by $node_i$, between receiving two consecutive updates from $node_j$. This measure is important in C-ACC situations, and ideally should be equal to $\frac{1}{\lambda_g}$ [16].

4.3 Simulation setup

For all simulation runs the parameters that are not changed during the simulations are listed in table 1.

These static parameters will be used in conjunction with the following parameters:

Number of nodes per 100m:
[20,40,80,120,160]

MAC queue length:
[1,2,3,4,5,10,15,20,30]

Scheduling
[FiFo, LiFo]

Buffer mechanism
[Newest Packet Drop, Oldest Packet Drop]

In order to guarantee a high statistical accuracy of the obtained results, multiple runs have been performed and 95% confidence intervals have been calculated for most of the simulation experiments. For all NPD related experiments, the largest calculated confidence interval is 2% of the shown calculated mean values. Due to the fact that these calculated confidence intervals were significantly low, only one run per experiment has been performed for all the ODP related experiments and no confidence intervals were calculated for these experiments.

5. SIMULATION RESULTS & ANALYSIS

This section contains the results from the simulations ran as described in section 4. All diagrams, visualising a performance measure, contain the graphs associated with the four simulated scenario's. They are divided per number of nodes for which the corresponding utilisation can be looked up in table 2.

Table 2. Measured utilisation per number of nodes

Nodes per 100m	Network utilisation
20	17%
40	33%
80	62%
120	87%
160	97%

Vehicle density: 20 vehicles/100m

- Queuing delay results are depicted in figure 5
- Contention delay results are depicted in figure 6
- Dropping probability results are depicted in figure 7
- Update delay / Inter-arrival time in figure 8

Vehicle density: 40 vehicles/100m

- Queuing delay results are depicted in figure 9
- Contention delay results are depicted in figure 10
- Dropping probability results are depicted in figure 11
- Update delay / Inter-arrival time in figure 12

Vehicle density: 80 vehicles/100m

- Queuing delay results are depicted in figure 13
- Contention delay results are depicted in figure 14
- Dropping probability results are depicted in figure 15
- Update delay / Inter-arrival time in figure 16

Vehicle density: 120 vehicles/100m

- Queuing delay results are depicted in figure 17
- Contention delay results are depicted in figure 18
- Dropping probability results are depicted in figure 19
- Update delay / Inter-arrival time in figure 20

Vehicle density: 160 vehicles/100m

- Queuing delay results are depicted in figure 21
- Contention delay results are depicted in figure 22
- Dropping probability results are depicted in figure 23
- Update delay / Inter-arrival time in figure 24

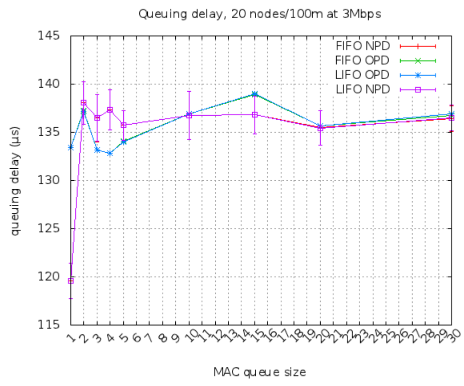

Figure 5. Queuing delay, 20 nodes/100m

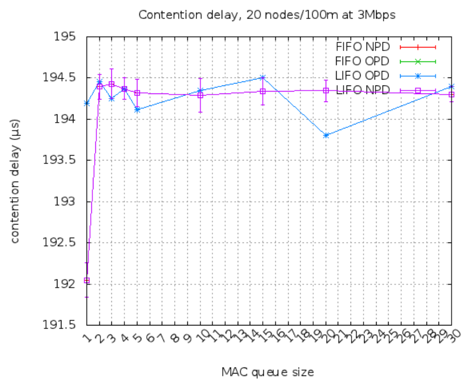

Figure 6. Contention delay, 20 nodes/100m

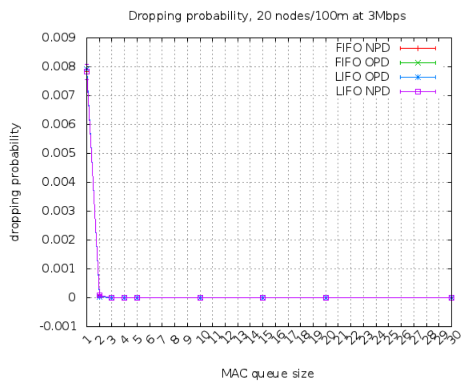

Figure 7. Dropping probability, 20 nodes/100m

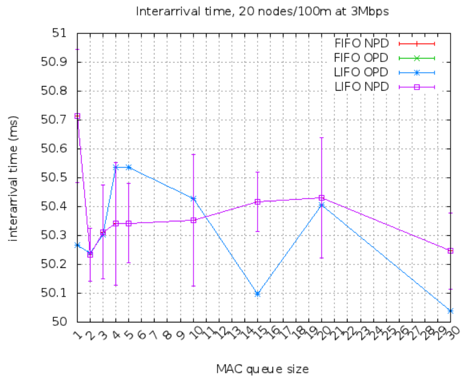

Figure 8. Interarrival time, 20 nodes/100m

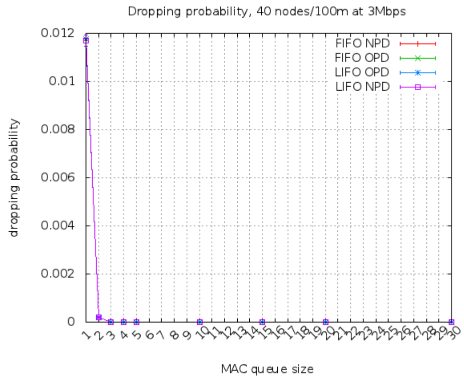

Figure 11. Dropping probability, 40 nodes/100m

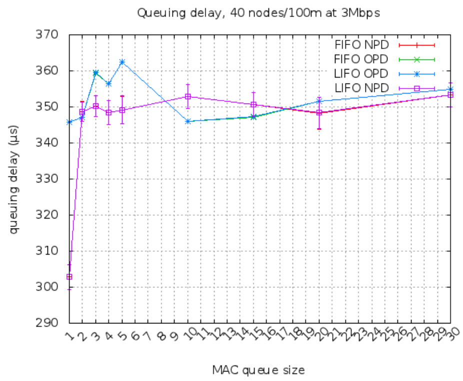

Figure 9. Queuing delay, 40 nodes/100m

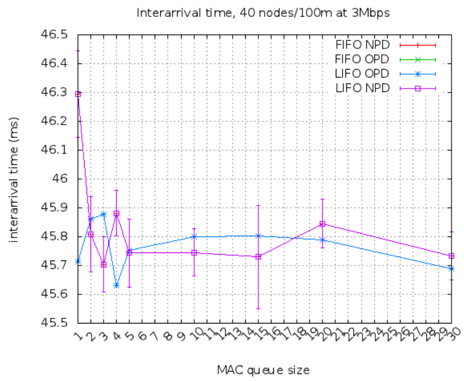

Figure 12. Interarrival time, 40 nodes/100m

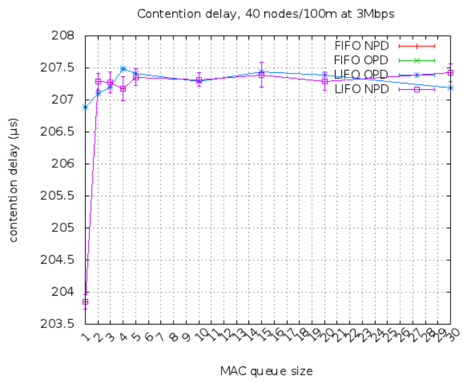

Figure 10. Contention delay, 40 nodes/100m

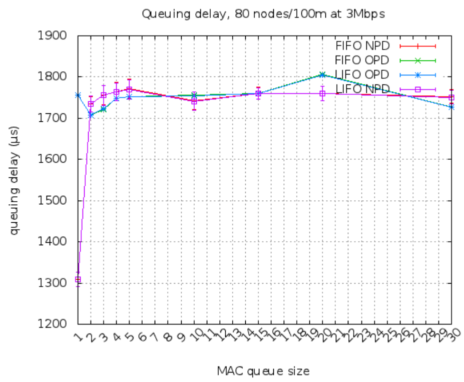

Figure 13. Queuing delay, 80 nodes/100m

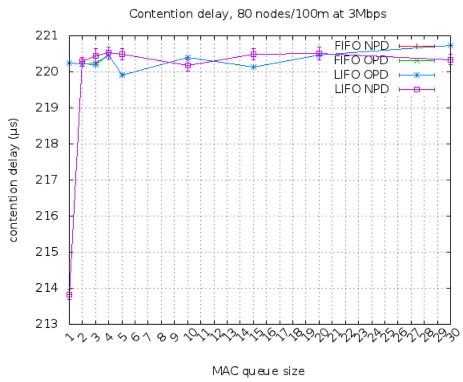

Figure 14. Contention delay, 80 nodes/100m

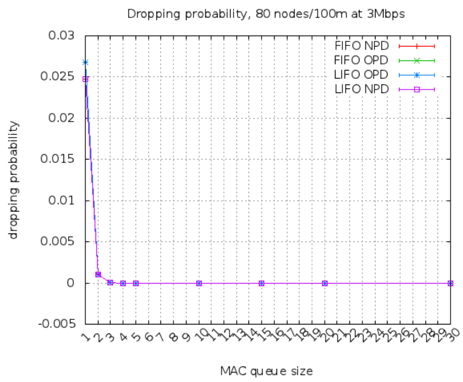

Figure 15. Dropping probability, 80 nodes/100m

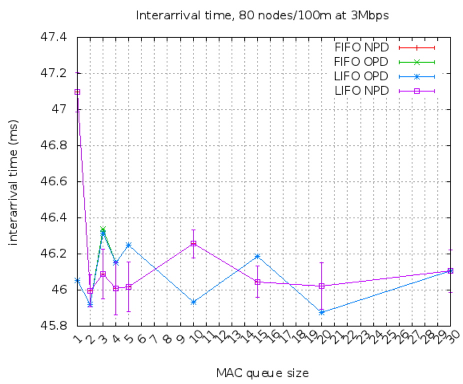

Figure 16. Interarrival time, 80 nodes/100m

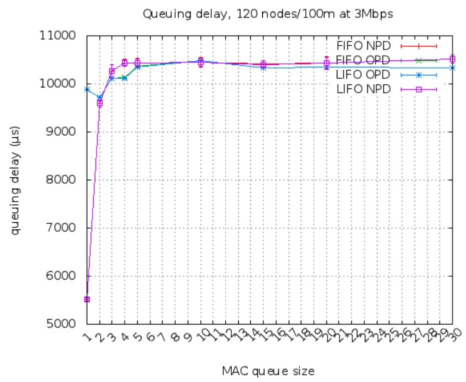

Figure 17. Queuing delay, 120 nodes/100m

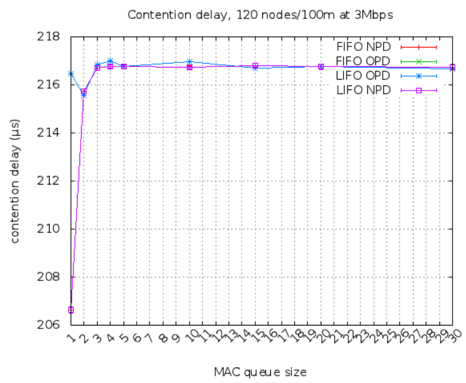

Figure 18. Contention delay, 120 nodes/100m

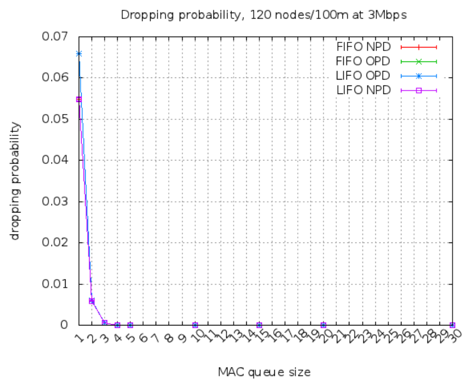

Figure 19. Dropping probability, 120 nodes/100m

5.1 Analysis

Figures 5 to 24 visualise the results obtained from the simulation experiments. For every traffic density/utilisation, the four performance measures are plotted (for all the scheduling discipline/ buffer mechanism combinations).

5.1.1 Queuing delay analysis

The graphs visualising queuing delay (figures 5, 9, 13, 17, 21) show, as expected, that there is no significant difference in queue delays when the FiFo or LiFo scheduling disciplines are used. The NPD and OPD mechanisms perform equally on queue delay, disregarding whether FiFo or LiFo is applied. For a queue size of only 1 position, the NPD out-performs OPD when the vehicle density is being increased. This is an unexpected behaviour and it needs further investigation.

5.1.2 Contention delay analysis

The contention delay, in figures 10, 14, 18 and 22 shows again that, as expected, there is no significant difference in contention delays when the FiFo or LiFo scheduling disciplines are used. The NPD buffering mechanisms used on a queue size of 1 position shows a notably lower delay in contention than the OPD buffering mechanism. This is an unexpected behaviour and it needs further investigation. When the queue size becomes larger than 1 position then

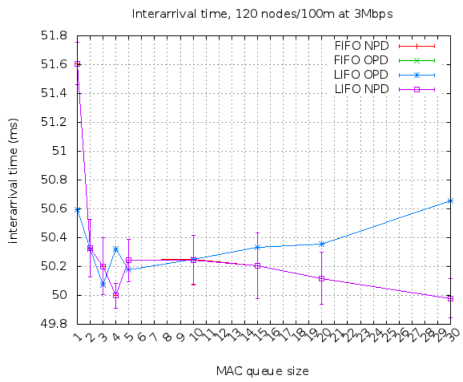

Figure 20. Interarrival time, 120 nodes/100m

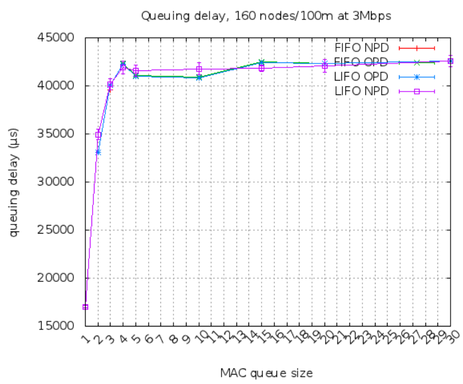

Figure 21. Queuing delay, 160 nodes/100m

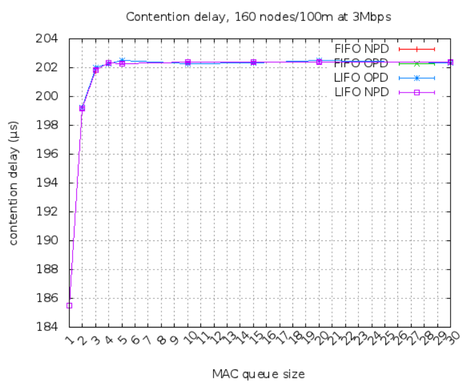

Figure 22. Contention delay, 160 nodes/100m

the NPD and ODP performance, as expected, seem to perform equally independent of the utilisation. This makes sense, as the contention period of a node is only affected by the utilisation. The contention delay rises when the utilisation of the network becomes higher.

5.1.3 Dropping probability analysis

The observed dropping probability is, as expected, almost the same for every investigated scenario. Naturally, more queue drops occur when smaller queue sizes are used, as can be seen in all of the figures (11, 15, 19 and 23). Furthermore we see that with the increase of utilisation (and therefore a busier network) more drops occur. From these experiments it can be concluded that a queue with a length of 5 beacon sizes is sufficient to be used for 802.11p beaconing.

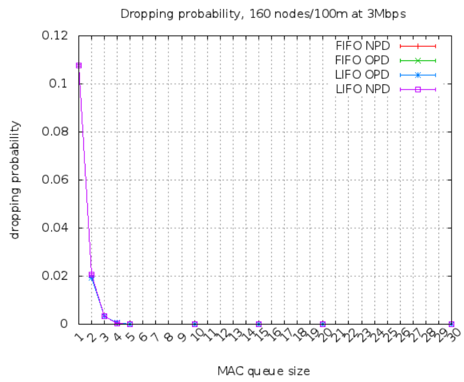

Figure 23. Dropping probability, 160 nodes/100m

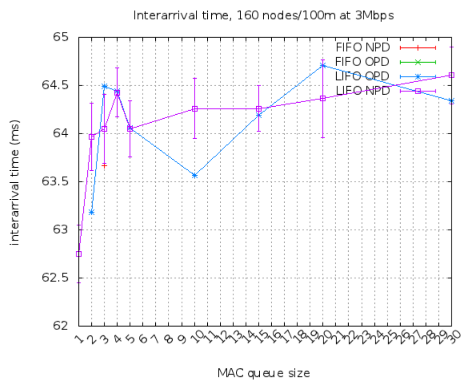

Figure 24. Interarrival time, 160 nodes/100m

5.1.4 Interarrival time analysis

The observed interarrival time is almost the same for every investigated scenario. OPD is slightly performing better in terms of interarrival time, when a queue length of only 1 position is used. This is an unexpected behaviour and it needs further investigation.

6. CONCLUSIONS AND FUTURE WORK

This paper provided an analysis of the effects of different transmission scheduling disciplines, buffer mechanisms and transmission queue sizes on the performance of 802.11p beaconing.

In particular, this paper discussed different IEEE 802.11p buffering and scheduling mechanisms, including a new OPD buffering mechanism. This buffering mechanism is used to ensure that beacons stored in the transmission queue are not becoming outdated. Furthermore, several simulation experiments have been performed in order to investigate the impact of different transmission scheduling disciplines, buffer mechanisms and transmission queue sizes on several beaconing performance measures. These measures are the queuing delay, the contention delay, the dropping probability and the interarrival time. Based on these investigations it can be concluded that a queue with a length of 5 beacon sizes is sufficient to be used for 802.11p beaconing. Moreover, the beaconing performance, as expected, is for almost all queue lengths not significantly influenced by the type of the buffering and scheduling mechanism used.

This is however, different for the situation that the queue has a length of 1 beacon size. In this situation, the NDP buffering mechanism outperforms the ODP mechanism in terms of queuing delay and contention delay.

This difference in ODP and NDP performance for a queue with a length of 1 beacon size, is unexpected. Therefore, more simulation experiments are needed to further investigate the behaviour of buffering mechanisms using a queue with a length of 1 beacon size. In addition to this more simulation experiments need to be accomplished in order to investigate other beaconing performance measures such as total end to end delay and the beaconing reception probability.

7. REFERENCES

[1] A. Banchs and L. Vollero. Throughput analysis and optimal configuration of 802.11 e edca. *Computer Networks*, 50(11):1749–1768, 2006.

[2] S. Biddlestone and K. Redmill. A gnu radio based testbed implementation with ieee 1609 wave functionality. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–7. IEEE.

[3] K. Bilstrup, E. Uhlemann, E. Strom, and U. Bilstrup. Evaluation of the ieee 802.11 p mac method for vehicle-to-vehicle communication. In *Vehicular Technology Conference, 2008. VTC 2008-Fall. IEEE 68th*, pages 1–5. IEEE.

[4] C. Brandauer, G. Iannaccone, C. Diot, T. Ziegler, S. Fdida, and M. May. Comparison of tail drop and active queue management performance for bulk-data and web-like internet traffic. *Computers and Communications, IEEE Symposium on*, 0:0122, 2001.

[5] S. Eichler. Performance evaluation of the ieee 802.11 p wave communication standard. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 2199–2203. IEEE, 2007.

[6] J. Gallardo, S. Cruz, D. Makrakis, and A. Shami. Analysis of the edca access mechanism for an ieee 802.11 e-compatible wireless lan. In *Computers and Communications, 2008. ISCC 2008. IEEE Symposium on*, pages 891–898. IEEE.

[7] J. Hui and M. Devetsikiotis. Performance analysis of ieee 802.11 e edca by a unified model. In *Global Telecommunications Conference, 2004. GLOBECOM'04. IEEE*, volume 2, pages 754–759. IEEE, 2004.

[8] IEEE. *IEEE Std 802.11e, Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Enhancements for Quality of Service (QoS)*. IEEE, 2005.

[9] IEEE. *IEEE Standard for Local and Metropolitan Area Networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments*. IEEE, 2010.

[10] R. Jain. *The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling*. John Wiley, New York, April 1991.

[11] G. Marsden, M. McDonald, and M. Brackstone. Towards an understanding of adaptive cruise control. *Transportation Research Part C: Emerging Technologies*, 9(1):33–51, 2001.

[12] MiXiM. Simulation framework for wireless and mobile networks. http://mixim.sourceforge.net/, Feb. 2011.

[13] B. Mughal, A. Wagan, and H. Hasbullah. Efficient congestion control in VANET for safety messaging. In *Information Technology (ITSim), 2010 International Symposium in*, volume 2, pages 654–659. IEEE, 2010.

[14] OMNeT++. Omnet++ network simulation framework. http://www.omnetpp.org/, Feb. 2011.

[15] N. Ramos, D. Panigrahi, and S. Dey. Quality of service provisioning in 802.11 e networks: challenges, approaches, and future directions. *Network, IEEE*, 19(4):14–20, 2005.

[16] R. Reinders, E. van Eenennaam, G. Karagiannis, and G. Heijenk. Contention window analysis for beaconing in vanets. In *Seventh IEEE International Wireless Communications and Mobile Computing conference, IWCMC 2011*, 2011.

[17] D. Rossi, R. Fracchia, and M. Meo. VANETs: Why Use Beaconing at All? In *Communications, 2008. ICC'08. IEEE International Conference on*, pages 2745–2751. IEEE, 2008.

[18] C. Semeria. Supporting differentiated service classes: queue scheduling disciplines. *Juniper networks*, 2001.

[19] K. Suleiman, T. Javidi, M. Liu, and S. Kittipiyakul. The Impact of MAC Buffer Size on the Throughput Performance of IEEE 802.11. *submitted to IEEE Transactions on Mobile Computing for publication on January*, 14, 2008.

[20] M. van Eenennaam, W. Wolterink, G. Karagiannis, and G. Heijenk. Exploring the solution space of beaconing in VANETs. In *Vehicular Networking Conference (VNC), 2009 IEEE*, pages 1–8. IEEE, 2010.

[21] A. Vinel, V. Vishnevsky, and Y. Koucheryavy. A simple analytical model for the periodic broadcasting in vehicular ad-hoc networks. In *GLOBECOM Workshops, 2008 IEEE*, pages 1–5. IEEE, 2009.

[22] L. Yang, J. Guo, and Y. Wu. Channel adaptive one hop broadcasting for VANETs. In *Intelligent Transportation Systems, 2008. ITSC 2008. 11th International IEEE Conference on*, pages 369–374. IEEE, 2008.