Hindawi Publishing Corporation
EURASIP Journal on Wireless Communications and Networking
Volume 2010, Article ID 712525, 13 pages
doi:10.1155/2010/712525

Research Article

A Cooperative Congestion Control Approach within VANETs: Formal Verification and Performance Evaluation

Mohamed Salah Bouassida and M. Shawky

Laboratoire HEUDIASYC, Centre de Recherche de Royallieu, Université de Technologie de Compiègne, B.P. 20529, 60205 Compiègne, France

Correspondence should be addressed to Mohamed Salah Bouassida, mbouassi@utc.fr

Received 1 June 2009; Revised 27 October 2009; Accepted 15 December 2009

Academic Editor: Christian Ibars

The main objective of congestion control is to best exploit the available network resources while preventing sustained overloads of network nodes and links. Appropriate congestion control mechanisms are essential to provide effcient operation of a network. Ensuring congestion control within vehicular ad hoc networks faces special challenges, due to the specificities of such environment (High mobility of nodes, high rate of topology changes, high variability in nodes density and neighborhood configuration, broadcast/geocast communication nature, etc.). In this context, we present in this paper a cooperative and fully distributed congestion control approach, based on dynamic scheduling and transmission of priority-based messages, to ensure reliable and safe communication architecture within VANET. Messages priorities are dynamically evaluated according to their types, the network context, and the neighboring nodes configuration. Considering the context of high reliability and real-time response required for intervehicular communications (including emergency breaking notification for example), we propose a complete validation method of our congestion control algorithms, taking into account reliability, temporal, and operational aspects.

1. Introduction

Vehicular ad hoc networks (VANET) are a type of MANETs used for communication among vehicles and between vehicles and roadside equipment (cf. Figure 1 [1]). In addition to the challenging characteristics of MANETs (such as lack of established infrastructure, wireless links, multi-hop broadcast communications), VANET brings new challenges to realize safe communication architecture within such environment. Indeed, within VANET networks, nodes are characterized by high dynamic and mobility, in addition to the high rate of topology changes and density variability. Zang et al. [2] evaluate the neighboring nodes configuration of vehicular networks within a four highway lanes context (two lanes for each direction). They carried out simulations and analysis showing that the average number of potential communication neighbors is optimally four. In addition, in 50% of all occurrences, the maximum potential communication duration is 1 second; in 90% of the occurrences, the upper boundary for the communication time is 5 s. Another important constraint in the multi-hop inter-vehicular communications is the limited bandwidth within a such environment. Indeed, the wireless channel can be occupied by competitive nodes for many reasons (collisions, interferences, insufficient signal strength, duration of the transmission sequence, etc.) [3].

To deal with this environment constraints, and in order to ensure safe and optimized communication architecture (to guarantee required services on a "best effort" network), setting up quality of service policies becomes mandatory, which inspires a congestion control approach within VANET (cf. Figure 2). We propose in this context a cooperative and fully distributed congestion control approach, dedicated to operate within vehicular networks, integrated within the 802.11p underway standard, and based on dynamically scheduling packets according to their priorities. Moreover, the available bandwidth is shared among neighbors so that vehicles sending higher priorities packets are favored. Considering the high real-time and reliability level needed by the inter-vehicular safety communications, we undertook a

Figure 1: Vehicular Ad hoc Network.

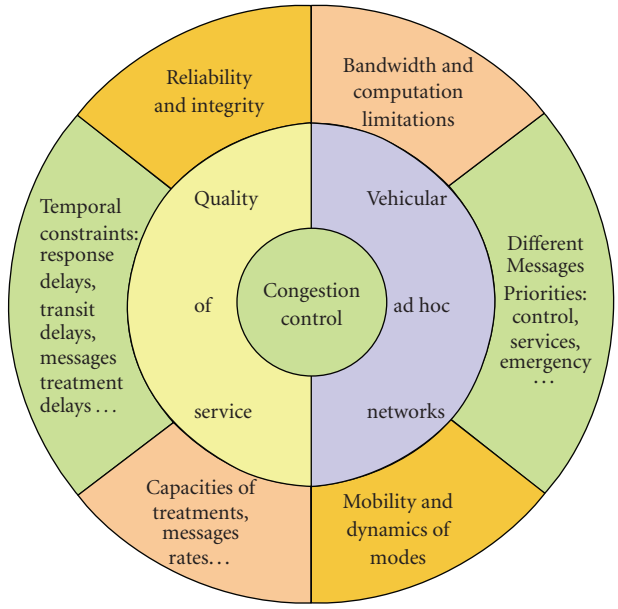

Figure 2: Congestion Control Context within VANET.

complete model based validation approach of our congestion control algorithms, taking into account reliability, temporal and operational aspects.

To present our contributions, this paper is structured as follows. In Section 2, we present related work concerning congestion control approaches within MANETs. Section 3 presents our cooperative congestion control approach. The real applicability of our approach is validated through formal verification, simulations and performance evaluations, that we present in Section 4. Finally, Section 5 concludes this paper, presenting the implementation of our congestion control scheme in the context of the European Integrated Project SAFESPOT (http://www.safespot-eu.org/), and details our future work concerning the real tests methodology we envisage to validate this implementation.

2. Congestion Control Approaches within MANETs

Congestion control is a challenging subject in mobile ad hoc networks, mainly because of the shared nature of the wireless multihop channel and the frequent changes of the network topology. Indeed, routes changes due to dynamic and mobility of nodes result in unsteady packet delivery delays and packet losses, which should not be considered as congestion control faults. In addition, the use of a shared multihop channel allows only one data transmission at a time within the interference range of a node. Thus, congestion in ad hoc networks affects a whole area and not only overloaded nodes [4]. During the last years, several congestion control approaches have been presented, dedicated to operate within ad hoc networks. In this section, we cannot claim to present an exhaustive study of these approaches. However, we distinguish two congestion control techniques for wireless networks: end-to-end and hop-by-hop families. End-to-end protocols aim to ensure flows fluidity between senders and receivers, without worrying about the internal relay nodes, whereas hop-by-hop congestion control methods take into consideration the capacities of the internal links. We present hereafter some protocols belonging to these two approaches.

2.1. End-to-End Congestion Control Approaches within MANETs. The TCP-F (TCP-Feedback) approach, proposed by Chandran et al. [5], consists of disabling the TCP congestion mechanisms in case of network-induced, non-congestion related, losses and time-out events. Each sender is notified when routing failure or route change occur, to freeze its TCP state values (timers and window sizes).

In the proposal of Rath et al. [6], an end-to-end congestion control technique is presented, carried out by TCP and physical layers. The adaptive windows based congestion control mechanism used by TCP for wired network may not be appropriate for wireless network. This is due to the time varying nature of a wireless channel and interference from other nodes causing packets loss, which is different from packets loss due to congestion. But TCP's congestion control mechanism does not distinguish between packets loss due to congestion and that due to bad channel or interferences; it rather applies the same congestion control mechanism for both. For this reason, within the proposed cross layer approach, the MAC layer changes transmission power as per the channel condition and interference received from the neighboring nodes, whereas the TCP layer controls congestion using Reno-2 windowing flow control.

The protocol presented in [7] updates the flow control model for a TCP-like method and extends it to the context of wireless network. It proposes two new congestion control schemes. The first one employs a static algorithm while the second applies a dynamic one. Both algorithms modify the number of connections that a single user has with the network and thus provide an appropriate number of connections, opened at the application layer by a sender.

2.2. Hop-by-Hop Congestion Control Approaches within MANETs. The authors in [8] show that congestion control techniques intended to operate within wired networks are not suitable for wireless environment, and that a cross layer hop-by-hop redesign is needed to consider the nature of

wireless communications. Indeed, with TCP, a flow can only obtain congestion feedback from every directed link along its path. However, nearby links not directly on the path can interfere with this flow because the fundamental difference between wireless and wired environment is that wireless communications are of a broadcast nature.

In the same point of view, Yi et al. argue in [9] that hop-by-hop schemes are feasible over a wireless network. Such techniques provide feedback about the congestion state at a node to the hop preceding it. This preceding node then adapts its transmission rate according to this feedback. Feedback is typically provided according to the queue length at the congested node. If the queue length exceeds a threshold, congestion is indicated and the preceding node is notified in order to decrease its transmission rate. Consequently, hop-by-hop schemes require to have per-flow state management for intermediate nodes, which induces scalability problems. However, in a wireless network, the number of flows per node is of much smaller order than in the Internet. Further, wireless networks usually have per-flow queuing because of packet scheduling, and the fact that different users are at different locations, thus requiring different physical layer strategies (such as the channel coding and modulation of the power level). The congestion control proposed in this paper is thus based on a hop-by-hop approach, which is shown to converge in the absence of delay and allocates bandwidth to various users in a proportionally-fair manner. The proposed hop-by-hop algorithm is established according to the queue length and the feedback delays. However, the utility and validity of the sent data are not considered within this technique, hence it is not adapted to the specific context of vehicular ad hoc highest transmission priority.

Another congestion avoidance protocol, named C3TCP [10], is also founded on link capacity measurements and adjustment of the outgoing data rate according to these measurements, at each sender node side. This solution requires the introduction of an additional module within the IEEE 802.11 protocol stack in order to carry out the capacity measurements. The same principle was used by Rangwala et al. in [11] to establish a congestion control scheme for static wireless mesh networks: a distributed rate controller estimates the available capacity within each neighborhood and divides it to contending flows.

Zhang et al. [12] investigate congestion control problems in multihop wireless networks, with time varying link capacities. By modeling time variations of capacities as perturbations of a constant link, the authors propose a primal-dual congestion control algorithm and prove its trajectory stability without feedback delays consideration. However, in the presence of delay, they define theoretical conditions for the technique to be locally stable. Both proposals presented in [9] and [12] are derived from the studies carried out by Kelly et al. [13]. In the context of vehicular ad hoc networks, the same authors (Zang et al.) present in [14] a congestion control approach for safety applications. The basic idea of this scheme is to identify congestions using event-driven detection and measurement, and to manipulate MAC transmission queues for IEEE 802.11p, in order to ensure the safety messages sent on the control channel. The event-driven congestion detection is triggered reactively whenever a high priority safety message is recognized to guarantee the QoS of the safety applications, while the measurement based congestion detection consists of measuring the channel usage and comparing it with a defined threshold. However, the effective transmission of the safety messages is not guaranteed because the neighborhood context and the effective bandwidth sharing are not considered.

The congestion control approach proposed in [15], dedicated to operate within vehicular ad hoc networks, consists of adapting transmissions to the available bandwidth in a hop-by-hop manner. Thus, nodes transmitting information with a high utility for VANET will be allowed to consume a larger share of the available bandwidth. A priority is evaluated for each packet, depending on its utility and size. Then, an instantaneous data rate is determined, according to the computed priority. The utility of messages is evaluated at the application layer and do not consider the neighborhood context in terms of density and dynamics of nodes. In addition, this approach generates a communication overhead, due to the context exchange between neighbor nodes in order to share the available bandwidth between them, without considering the capacity and the congestion state of the forwarder nodes.

Torrent-Moreno et al. present in [16] a fair bandwidth sharing approach for VANETs. This approach consists of limiting the wireless load resulting from the periodic messages, by requiring a strict fairness among the vehicles. The authors assume a constant packet generation rate, and propose a centralized power control algorithm that provides the optimum transmission range of every node. This proposal was formally validated. However, simulations have been carried out under idealistic conditions, assuming that interferences between nodes transmissions follow a deterministic model. Moreover, the proposed algorithm requires synchronization between vehicles, which generates communication overhead.

2.3. Summary. In order to define a congestion control approach within vehicular ad hoc networks, several constraints should be considered, related to the characteristics of the environment and also to high quality of service (QoS) required for the safety-oriented data. On one hand, as argued in [9], end-to-end congestion control approaches are not suitable for wireless ad hoc networks. Indeed, within these approaches, relay nodes context are not considered, and thus, interferences, collisions and transmission problems are neglected. However, the required quality of service of a transmission can be defined by the sender.

On the other hand, hop-by-hop approaches suffer from lack of scalability, when the number of transmitted flows increases within the network. However, it is known that the size of the transmitted data within VANET is small, due to the dynamic nature of this environment, and to the nodes limitations in terms of storage and computation capacities. It is thus admitted that hop-by-hop congestion control approaches are more suitable for VANET, while considering the required quality of service of the transmitted data, as for the end-to-end approaches. Hop-by-hop congestion control approaches, described above, present some

disadvantages (communication and computation overheads, reactive congestion control techniques, idealistic verification frameworks, etc.). In addition, several parameters are not completely considered within these protocols, such as the velocity of the sender node, the appropriate choice of the next forwarders, the validity and the utility of the sent messages according to the neighborhood context and the required QoS.

We propose in the next section a congestion control approach for vehicular ad hoc networks, considering these drawbacks, whose design should ensure the following objectives.

(i) Cooperative and adaptive congestion control approach: approach dynamically adaptable to the neighborhood and the context of VANET, while taking into consideration the required QoS metrics (in terms of reliability and delays) especially for emergency and safety messages.

(ii) Hop-by-hop approach: to deal with the capacity of relay/forwarders nodes and in order to efficiently share the effective bandwidth between neighbors nodes, while considering the required quality of service as for the end-to-end congestion control approaches.

(iii) Cross-layered approach in order to ensure a dynamic and reliable messages scheduling and transmissions processes, while remaining adapted to the IEEE 802.11p underlying standard.

(iv) Applicative layer congestion control approach, in order to define packets priorities according to their application, their utilities and validities in the network and the neighborhood context.

3. Cooperative Congestion Control Approach within VANET

3.1. Overview. The basic idea of our applicative-layer congestion control approach is to define policies, in order to dynamically and cooperatively schedule messages transmission in the network. Messages scheduling is carried out according to priorities, evaluated as a function of the utility of the concerned messages, the sender application and the neighborhood context. The messages transmission in the vehicular network is carried out in an efficient and cooperative manner, by favoring vehicles holding the highest-priority messages to send. Therefore, our approach is divided into three steps that we present hereafter: dynamic priority assignment, message scheduling and cooperative message transmission.

3.2. Priority Assignment. Messages will be assigned a priority by application initiating. The relative time of transmission of each priority level will however vary as network density increases: medium and low priority packets being delayed to allow high priority packets to be sent without delays. The priority of a packet is composed of 2 fields: the first is static, deduced from the application type and the second is dynamic, obtained from the specific context of the VANET (neighborhood density) and determined by the congestion control module. The size of the message, the dynamic and static fields are combined to obtain the overall priority indicator ($\text{Pri}_{\text{message}}$ = Dynamic_factor × Static_Message_Priority/Message_size).

3.2.1. Static Factor from Application Class. The static priority factor is defined according to the sender application, and the content of the message. Five priority levels are adopted by the C2C Communication Consortium, defined hereafter.

(i) $\text{PRI}_{\text{Emergency}}$ is the priority affected to single hop emergency messages to notify an important event without delay. The safety of vehicles depends on this kind of messages. Regarding Multihop and Geocast communication it is assumed that only the first hop can have the priority $\text{PRI}_{\text{Emergency}}$.

(ii) $\text{PRI}_{\text{VANET}}$ is the priority affected to the network layer beacons.

(iii) PRI_{HIGH} is the priority affected to high priority safety applications.

(iv) PRI_{MID} is the priority affected to normal safety applications.

(v) PRI_{LOW} is the priority affected to low priority applications.

3.2.2. Dynamic Factor from Network Context. We present in the following how our congestion control approach evaluates the dynamic priority factor of sent messages (a part of this work has been carried out jointly with the VANET research group in the HEUDIASYC laboratory).

Node Speed Consideration. The Dynamic factor takes into account the node speed, according to the covered zone at each dt, as illustrated in Figure 3. The priority of a message increases when the speed of the sender increases. Thus, at each dt, the dynamic factor is re-evaluated as follows:

$$\text{Speed\_factor} = \frac{\pi R^2 + 2RV \cdot dt}{\pi \cdot R^2}, \tag{1}$$

where R is communication range and, V: mean speed.

Message Utility Consideration. The dynamic factor considers also the utility of the sent message, according to the number of its retransmissions by the neighborhood, in case of periodic or geocast messages. Thus, when a node A has to send a periodic or geocast message M, and receives the same message M, sent by another node B (cf. Figure 4), it should calibrate the dynamic factor of the message M, in order to take into account the zone covered by the node B, compared to its communication range zone ($= \pi \cdot R^2$ with R is the

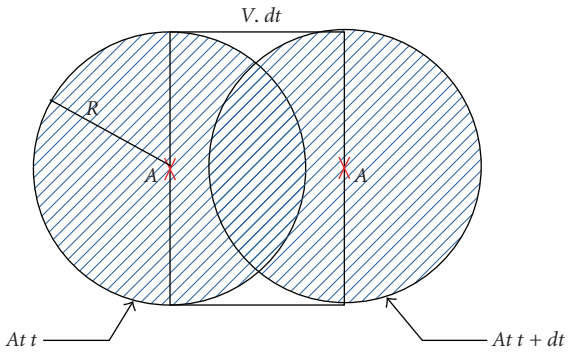

Figure 3: Node Speed Consideration.

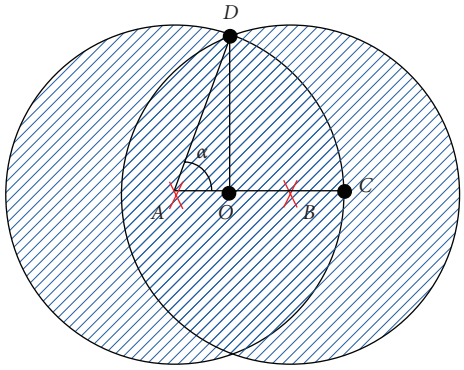

Figure 4: Message Utility Consideration.

communication range). The smaller is the covered zone, the higher is the priority to send the message. The dynamic factor is thus equal to the ratio between the total zone covered by the receiver and the already covered zone (CZ):

$$\text{Utility}_{\text{factor}} = \frac{\text{Total}_{\text{Zone}}}{\text{Covered}_{\text{Zone}}} = \frac{\Pi R^2}{\text{CZ}}. \quad (2)$$

We demonstrated that the covered zone CZ by the node B (doubly hatched in Figure 4) at the side of the node A is evaluated as follows (we note d the distance between A and B, and R the range of A and B):

$$\text{CZ} = 4 \times \left(\arccos\left(\frac{d}{2R}\right) \times \frac{R^2}{2} - \frac{d}{2} \times \sqrt{R^2 - \left(\frac{d}{2}\right)^2} \right). \quad (3)$$

Message Validity Consideration. The dynamic factor considers the message validity (maximum duration of the message). As for the EDF scheduling approach (Earliest Deadline First), the message whose deadline is earliest, holds the highest priority. The final dynamic factor is thus computed, according to the speed, utility and validity factors, as follows:

if remaining_time_to_deadline $\neq 0$

$$\text{Dynamic\_factor} = \frac{\text{Speed\_factor} \times \text{Utility\_factor}}{\text{remaining\_time\_to\_deadline}} \quad (4)$$

else

$$\text{Dynamic\_factor} = \text{Speed\_factor} \times \text{Utility\_factor} \quad (5)$$

end if.

3.3. Messages Scheduling. Each node schedules its messages according to their priorities, in the appropriate channel. The Car to car Communication Consortium (C2C CC) considers two VANET wireless channels (control and service channels), each used for different traffic [17].

Control Channel (CCH). The control channel is primarily used to transmit beacons and high/first hop priority traffic. All messages that are necessary to maintain the VANET are transmitted on this channel, especially the network layer beacons. Furthermore, high priority messages (emergency notifications) are sent on this channel. Normally, such messages occur on an event basis. With multihop communications, only the first hop will require high priority.

Service Channel (SCH1). This channel is available for safety applications with lower priority. Here periodic messages could be sent. This channel should also be used by forwarders of multihop and geocast messages. A second service channel (SCH2) is intended to short distance peer to peer VANET communications, with reduced power level. However, this service channel is currently unused.

Hence, we split the scheduling process into two phases: static and dynamic, presented hereafter.

3.3.1. Static Scheduling. The static scheduling process consists of dispatching messages according to their priorities, into the suitable communication channel queues. Thus, $\text{PRI}_{\text{Emergency}}$, $\text{PRI}_{\text{VANET}}$ and PRI_{HIGH} priority messages are affected to the control communication channel queue, whereas PRI_{MID} and PRI_{LOW} priority messages are affected to the service queue.

3.3.2. Dynamic Scheduling. Periodically, each node triggers a rescheduling process, which scans the messages queues, and computes the overall priority indicator for each message (considering the dynamic factor of each priority, presented above). The rescheduling process then reorders the messages according to their new computed priorities.

Considering that the number of messages sent within the control channel is smaller than the number of messages sent within the service one, we adopt the following policy: when the service channel is overloaded and the control channel is free, messages within the service queue are switched to the control one, and considered as high priority messages. We estimate that the service channel is overloaded if the number of messages in the queue exceeds a defined threshold, called "Service Channel Congestion Threshold".

3.4. Cooperative-Based Messages Transmission. Messages transmission process sends the highest priority message within the corresponding channel, whenever it is free. However, sending high priority packets via the control channel is preemptive, compared to packets sent via service channel. Indeed, in order to send high priority packets with the minimum delay, lower priority packets emission is freezed, even if their corresponding channel is free. We divide our cooperative transmission technique into two main mechanisms that we present hereafter: the available

bandwidth sharing and the next forwarder selection for the multihop communication case. These procedures require the modification of the periodic beacon structure, that we present later.

3.4.1. Bandwidth Sharing. Concerning the dynamic use of the bandwidth within VANET, the IEEE 802.11p underway standard supports three mandatory user data rates 3 Mbit/s, 6 Mbit/s and 12 Mbit/s within a 10 Mhz channel, and some optional data rates up to 27 Mbit/s. The most robust data rate is the 3 Mbit/s one. This rate must be shared among all applications and vehicles inside the interference range. In order not to saturate the provided bandwidth and to allow a reliable transmission of the emergency messages, the bandwidth offered to VANET application per 10 Mhz is equal to the half of the total bandwidth.

The simplest way to share the available bandwidth to the neighbors is to divide it equitably between them, as follows. Let n denotes the number of neighbors of a node. The effective bandwidth that a vehicle can use within the vehicular network is thus computed as

$$\text{Effective\_Bandwidth} = \frac{\text{Selected\_Bandwidth}}{2(n+1)}. \quad (6)$$

However, such a solution treating all the neighbors equitably does not favour higher-priority messages holders to use the available bandwidth, nor avoids eventual collisions and interferences. The suitable solution to eliminate these drawbacks is to offer the available bandwidth to the transmitter node whose message holds the highest priority compared to the messages of its neighbors.

In order to notify its neighbors about the priority of the first message it has to send, each vehicle includes this information in its beacon structure, as presented in Section 3.4.3. Therefore, as for the token ring communication protocol, a node can use the available bandwidth only if it holds the highest-priority message (it does not receive any beacon notifying the occurrence of a higher-priority message holder within its neighborhood). In the same way, when a node receives beacons notifying the presence of higher-priority messages than messages that it will send (the first messages in its queues), it freezes its transmission.

When two vehicles have to send two messages with the same priority, as for the FIFO scheduling model, the available bandwidth will be devoted to the first vehicle who notifies the priority of its message. The time of a first notification corresponding to a message priority should thus be included in the beacon structure. Note that generally, messages corresponding to a vehicular application have the same priority; consequently, the priority notification sent within beacons is not frequently modified.

3.4.2. Next Forwarder Selection. In the case of multi-hop inter-vehicular communications, the choice of the next forwarder is essential in order to enhance the performances of the communication architecture. Therefore, the next forwarder should be chosen as the less congested node within the neighborhood. In this context, we propose to define the congestion level principle as follows.

Identity	Geographical localization	Message priority	First notification time	Congestion control level

Figure 5: Structure of a Beacon.

Definition 1. The congestion level of a node (expressed in seconds) is evaluated as the ratio between the total size of messages to send by the available theoretical bandwidth (cf. Equation (6)). This parameter evaluates the required time (in seconds) in order to send all the waiting messages stored in the queues, using the effective bandwidth (according to the number of neighbors).

Therefore, a vehicle chooses the neighbor with the smallest congestion control, as a next forwarder. In order to notify its neighbors about its congestion level, each vehicle includes it within its beacon structure, as presented in Section 3.4.3. The congestion control level can also be used by each node to control its internal congestion: for example, if this level exceeds a defined threshold, the lowest-priority messages will be deleted.

3.4.3. Periodic Beacon Structure. Beacons messages are sent periodically by each node of the network, to enable neighbors discovery and the network maintenance. Within vehicular architectures, the beacon contains mainly the identity of the node and its geographical localization. In order to take into account the bandwidth sharing and the next forwarder selection procedures, the beacon structure should include, in a piggybacking manner, the priority of the first message it has to send, the time of the first notification of this priority and the last evaluated congestion level, as shown in Figure 5. Note that the increase of the beacon structure could generate a communication overhead. This overhead should remain negligible compared to the messages sending load.

4. Analysis and Validation

4.1. Objectives. In order to validate the real applicability of our congestion control approach and its operability within vehicular networks, while considering the criticality and hostility of such an environment, we followed a complete QoS validation approach, according to two main steps.

(i) *Formal modeling and verification*. This validation step deals with the reliability and operational aspects of our congestion control approach, to verify reachability, safety, liveness and no deadlock properties.

(ii) *Simulations and performance analysis*. This second step verifies temporal and operational constraints by measuring the delays of messages before their sending on the appropriate queue, and the service packets-loss rate.

4.2. Formal Verification. Formal verification of a hardware or software system consists of proving or disproving the correctness of intended algorithms/approaches underlying a system with respect to a certain formal specification or property, using formal methods of mathematics.

In our context, the quality of service of a system is composed of a constraints set (delays, response time, data rate, etc.). The system operation ensures the required QoS if it satisfies the needed constraints. Automata allow to model dynamic control aspects of a system operation. These models, in addition to temporal mechanisms (called temporized automata), are suitable to specify temporal QoS constraints [18, 19].

In this context, and in order to verify and validate formally our congestion control technique, we propose to specify it using temporized automata, through the UPPAAL (http://www.uppaal.com/) integrated tool environment for modeling, validation and verification of real-time systems. UPPAAL is developed in collaboration between the Department of Information Technology at Uppsala University, Sweden and the Department of Computer Science at Aalborg University in Denmark. To present our validation process, we first start by presenting an overview of temporized automata and the UPPAAL tool. Then, we present our objectives, simulations and results.

4.2.1. Temporized Automata. In a temporized automaton, transitions between states are conditioned by temporal constraints on clock variables. An elementary temporal constraint is a boolean property in which a clock variable is compared to a constant integer. Extended temporized automata enhance temporized automata by providing the possibility of manipulation of non-temporal variables. In the following description, we do not distinguish between "temporized automata" and "extended temporized automata".

A temporized automaton is a sextuplet $(\sum, S, S_0, C, V, E)$, where

(i) $\sum$ is a finite set of actions,

(ii) S is a finite set of states,

(iii) $S_0 \in S$ is the initial state,

(iv) C is a finite set of clocks,

(v) V is a finite set of variables,

(vi) E is the set of transitions. A transition is a tuple $(s, \mu, \gamma, \lambda, \lambda', s')$ indicating that, starting by the state s, the automaton executes the action μ, if the constraint γ is satisfied; clocks of λ are reset, variables of λ' are updated and the new state is s'.

4.2.2. The UPPAAL Modeling, Validation and Verification Tool of Real-Time Systems. UPPAAL is a toolbox for validation (via graphical simulation) and verification (via automatic model-checking) of real-time systems. It consists of two main parts: a graphical user interface and a model-checker engine. The idea is to model a system using temporized automata, simulate it and then verify properties on it. A real-time system in UPPAAL is composed of concurrent processes, each of them modeled as an automaton. The automaton has a set of locations. Transitions are used to change location. To control when to fire a transition, it is possible to have a guard and a synchronization. A guard is a condition on the variables and the clocks precizing when the transition is enabled. The synchronization mechanism in UPPAAL is a hand-shaking synchronization: two processes take a transition at the same time in the synchronization channel **a**, one will have a **!a**, and the other **a?**. The verification tool provided by UPPAAL, checks for the following properties.

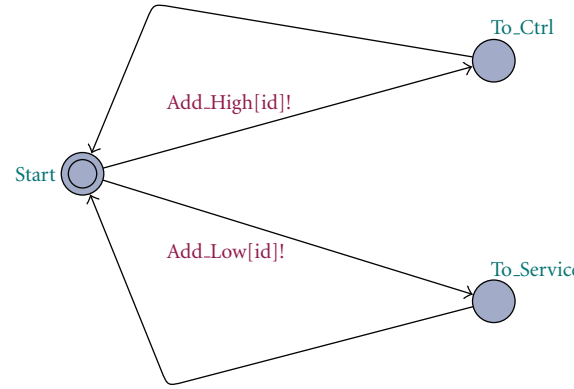

Figure 6: Message Manager.

(i) Reachability properties. These properties ask whether for a given state formula φ, there exists a path starting at the initial state, such that φ is eventually satisfied along that path. Reachability properties do not by themselves guarantee the correctness of the system, but they validate the basic behavior of the model.

(ii) Safety properties. These properties ask whether a bad result will never happen, or an awaited result is invariantly true.

(iii) Liveness properties. These properties are of the form "an awaited result will eventually happen". With this type of properties, response conditions can be verified as follows "whenever φ is satisfied, eventually ψ will be satisfied".

(iv) No Deadlock property. This property verifies if there is no any state where there are no outgoing action transitions, neither from the state itself or any of its delay successors.

4.2.3. UPPAAL Simulations and Results. To simulate via UPPAAL our priority-based scheduling mechanism within VANET, we divide our system into four independent subsystems, represented each by an automaton that we describe hereafter.

(i) The message manager automaton (cf. Figure 6): this subsystem is responsible for generating messages and sending them to the congestion control module, which will process them according to their priorities. To simplify our simulation platform, and without loss of generality, we consider hereafter two levels of priorities (high priority packets sent through the control channel and low priority packets transmitted in the service channel).

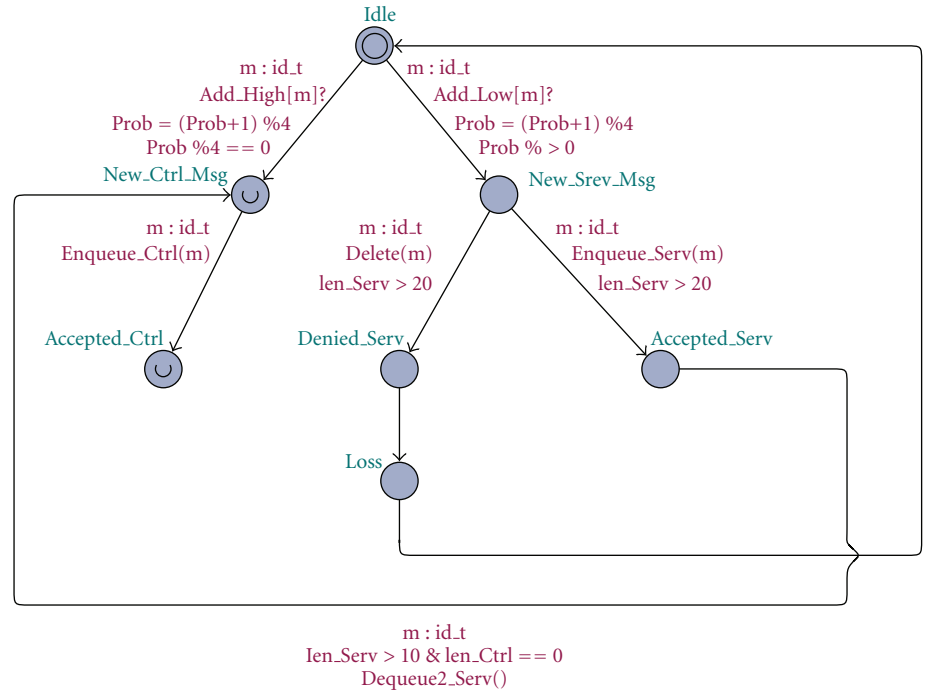

Figure 7: Congestion Control Message Enqueueing.

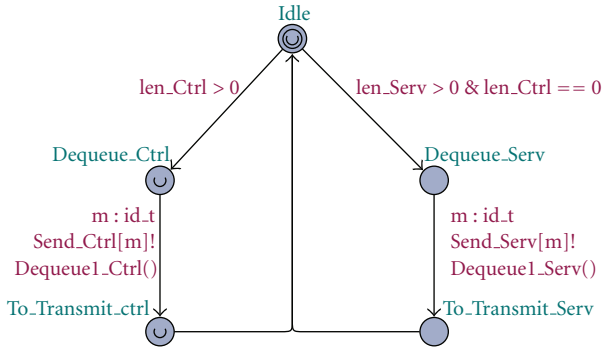

Figure 8: Congestion Control Message Dequeueing.

(ii) The congestion control message enqueueing automaton (cf. Figure 7): this sub-system is responsible for the reception of messages from the message manager and for their addition to the appropriate queues. A Add_High synchronization message is required to transit from the Idle state to the New_Ctrl_Msg one. The automaton switch to the Accepted_Ctrl state after adding the high priority message to the appropriate queue. The addition of a low priority message follows the same process, with the difference that the congestion control module can deny sending a message if the service channel is overloaded (number of low priority messages > 20). Note that when the service channel is overloaded and there is no high priority packets to send, a low priority message can be sent via the control channel. This message is thus dequeued from the service queue list and enqueued in the control channel one.

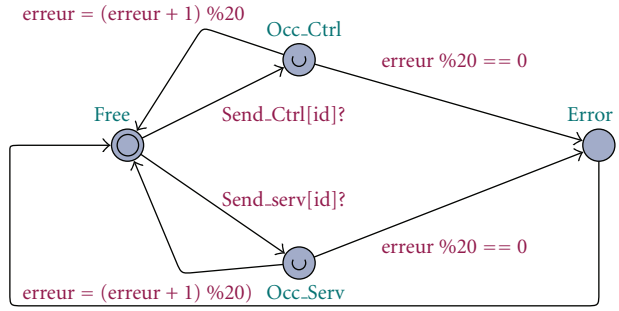

Figure 9: Transmission Engine.

(iii) The congestion control message dequeueing automaton (cf. Figure 8): this subsystem is responsible for withdrawing messages from the control and service queues and transmit them to the transmission engine. The synchronization messages between the congestion control message dequeueing automaton and the transmission engine one are Send_Ctrl and Send_Serv.

(iv) The transmission engine automaton (cf. Figure 9): this sub-system is responsible for the messages effective transmission on the appropriate channels. A sending error rate of 5% (The sending error rate is defined in the context of the SAFESPOT European Integrated project.) is chosen. When an error occurs during message sending, the automaton switches to the Error state.

The first step of validation is the simulation. We thus randomly simulate all the possible transitions of the four

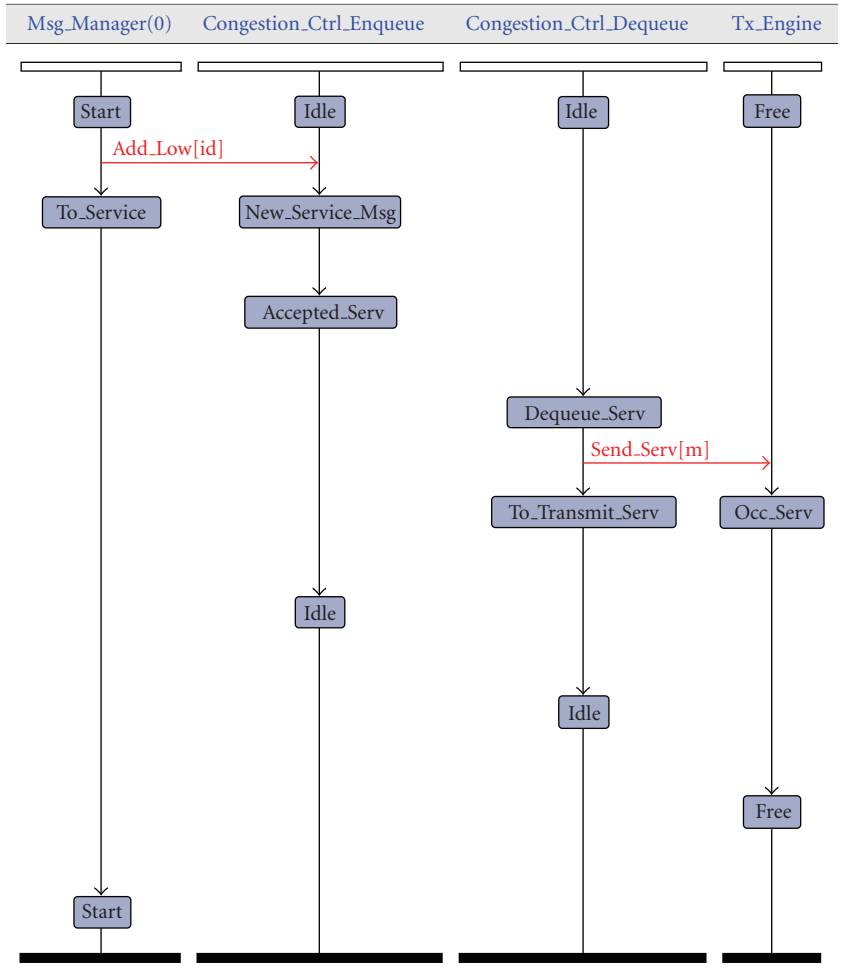

Figure 10: Service Packet Sending Diagram.

automata. Figure 10 illustrates an example of the execution of a low priority packet sending (activity diagram generated by UPPAAL).

The verification step validates the following results:

(i) No deadlock in the operation of our messages priorities-based scheduling approach. All states of the modeled automata have successors.

(ii) All states of the modeled automata are eventually reachable.

(iii) All the high priority messages are effectively sent on the control channel. However, some low priority packets can be deleted, due to service channel congestion (bounded service messages queue).

(iv) The transmission of high priority messages is preemptive comparing to the emission of low priority messages. Service messages are sent only when there is no any control message in the queue of the control channel.

(v) In addition, the emission of a high priority message is carried out without delay. All the high priority messages are considered as emergency, requiring thus to freeze the emission of lower priority messages.

We tried then to carry out the second validation step provided by UPPAAL: the model checking verification, after specifying our verification objectives (queries) via a description language. However, this verification step did not succeed due to memory limitations. Indeed, the number of states and variables in UPPAAL can make the model-checking verification process very constraining and complex [20]. Although the success of the model checking verification step may validate explicitly the correctness of the verification objectives (by exploring all the possible paths in the graph), we can affirm nevertheless that our priority-based scheduling approach is correct, considering the results of the first verification step (consisting of validating with success the use-cases corresponding to the congestion control scheme operation: sending high-priority and low-priority messages on the control and service channels resp.).

4.3. Performance Evaluation and Analysis. We developed a simulator, in C language, in order to validate the performances of our congestion control approach, according to the metric of service packets-loss rate, and the delays of the messages before their effective transmission on the appropriate queues. Note that service messages are considered lost when the number of waiting service messages exceeds a defined

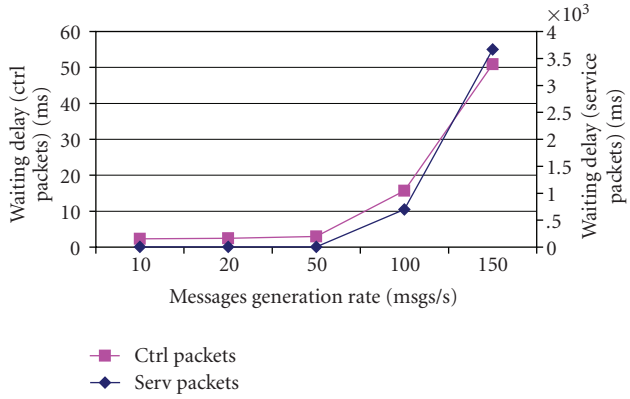

Figure 11: Delay versus Messages Generation Rate (mean message size = 500 bytes and mean number of neighbors = 50).

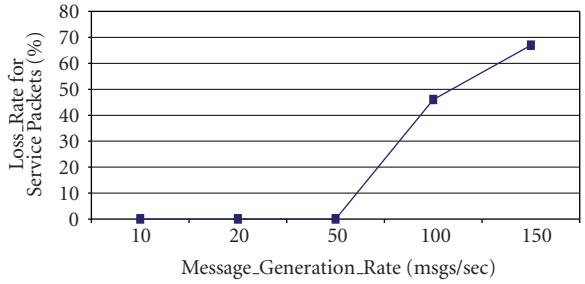

Figure 12: Service Packets-Loss Rate versus Messages Generation Rate (mean message size = 500 bytes and mean number of neighbors = 50).

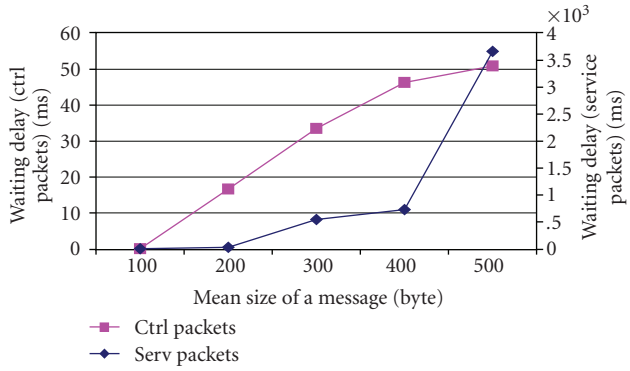

Figure 13: Delay versus Mean Message Size (message generation rate = 150 msgs/s and mean number of neighbors = 50).

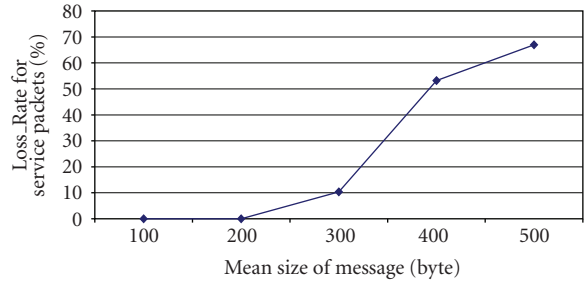

Figure 14: Service Packets-Loss Rate versus Mean Message Size (message generation rate = 150 msgs/s and mean number of neighbors = 50).

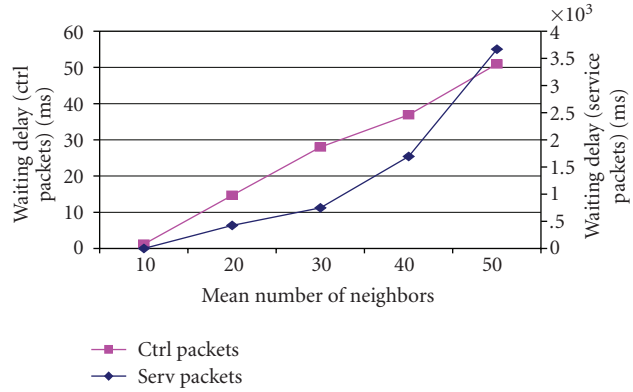

Figure 15: Delay versus Mean Number of Neighbors (message generation rate = 150 msgs/s and mean message size = 500 bytes).

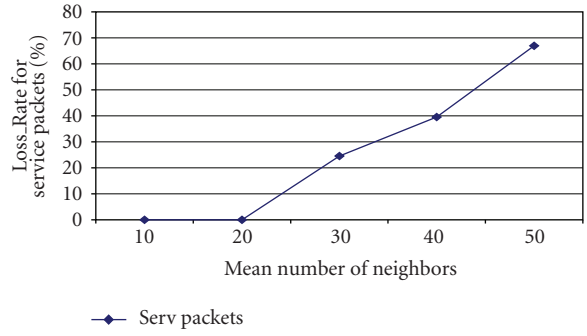

Figure 16: Service Packets-Loss Rate versus Mean Number of Neighbors (message generation rate = 150 msgs/s and mean message size = 500 bytes).

threshold; the service messages with the lowest priorities are thus dropped. The parameters of our simulations are presented hereafter:

(i) messages generation rate: number of messages generated by the messages manager per second,

(ii) mean message size, in order to consider the necessary delay to send each message, according to the effective bandwidth,

(iii) mean number of neighbors, in order to evaluate the effective theoretical bandwidth available for each node.

The results of our simulations are presented in Figures 11, 12, 13, 14, 15 and 16. Figures 11 and 12 show the delay for the service and control packets, and the service packets-loss rate, by the messages generation rate chosen by the message manager. Figures 13 and 14 present the impact of the mean messages size on the delays and the service packets loss rate. And finally, Figures 15 and 16 show the impact of the mean

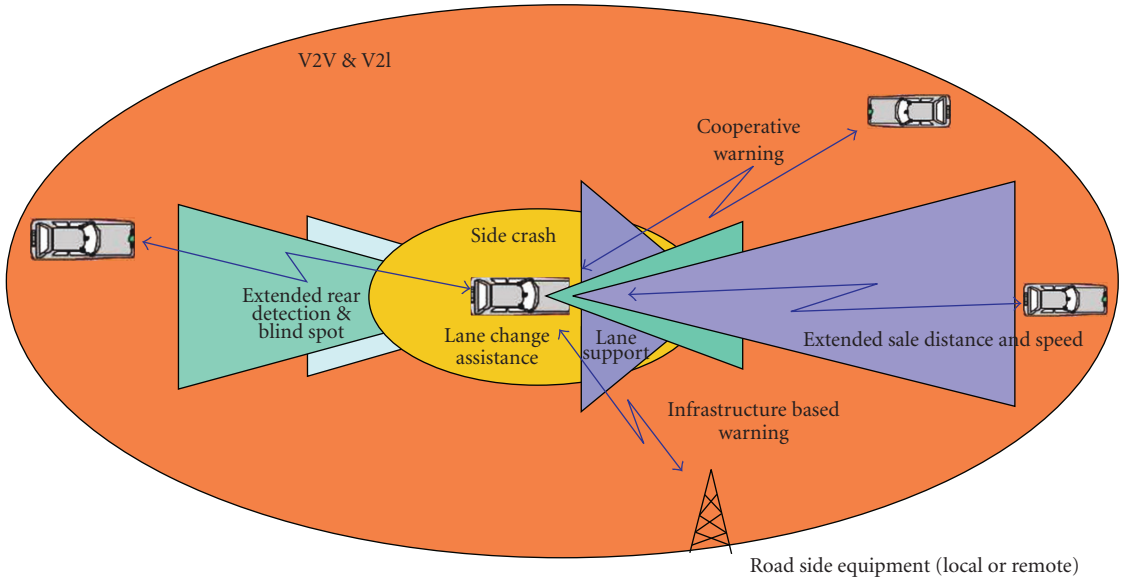

Figure 17: SAFESPOT European Integrated Project Context.

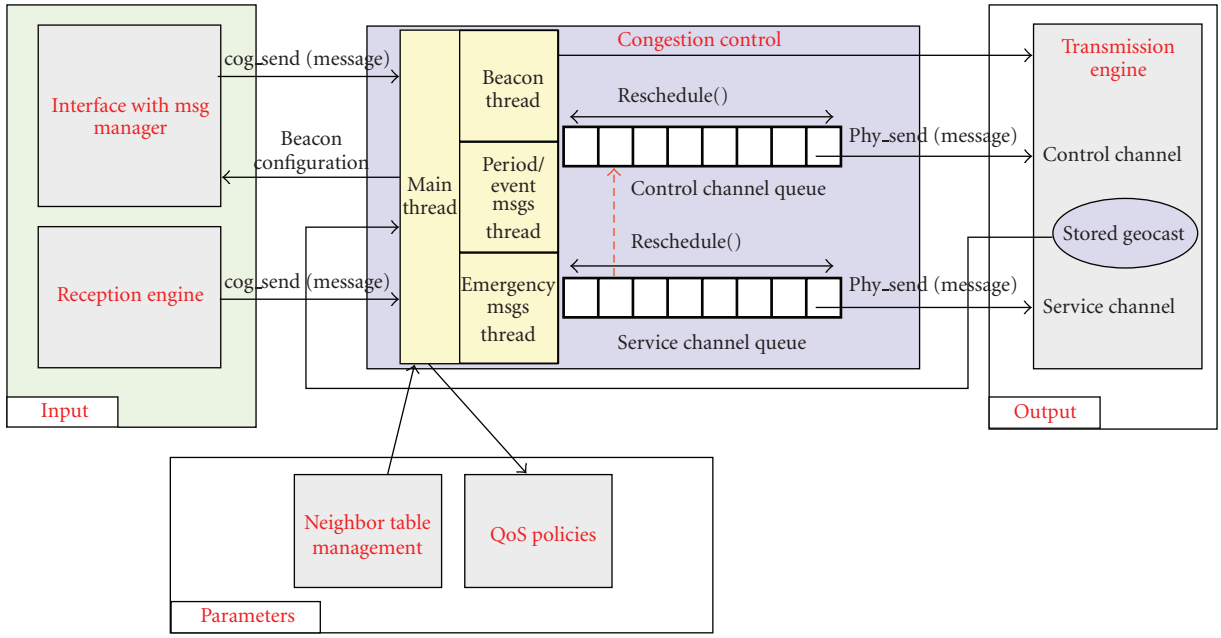

Figure 18: VANET Router Architecture.

number of neighbors on the delays and the service packets-loss rate.

From these simulations, we show that the service packets-loss rate is affected by the three parameters of our analysis. Indeed, it increases with the increase of the messages generation rate, the mean size of messages and the mean number of neighbors. However, the service packets-loss rate remains almost negligible in most cases of the network context and the ego node charge. Note that messages sent on the control channel cannot be lost or dropped. Their transmission is preemptive comparing to messages sent on the service channel.

The delays of the control and service packets before their effective emission are also affected by the three parameters of our simulations. These delays increase with the increase of messages generation rate, the mean size of messages and the mean number of neighbors. We note that the delays concerning the control messages are low, and do not exceed in the worst cases 60 ms. QoS requirements for the high priorities messages are thus satisfied, which is a challenging issue for the emergency alerts dissemination within VANETs. However, the delays of the service messages is almost 1 s for the "normal" context of network and charge, and can reach 3 s in the worst cases.

In addition, our priority-based congestion control technique is dynamically adaptable to the context of the neighborhood, taking into consideration the local density and messages priorities in order to evaluate the optimal bandwidth sharing process between neighbor mobiles. Indeed, Figure 15 and Figure 16 illustrate the adaptability of our congestion control method to the variation of the mean number of neighbors: its performances in terms of delays and service packets loss-rate are very promising in the "normal" context of vehicular environment (e.g., the average number of potential communication neighbors within a four highway lanes context is appreciatively four [2]).

As a conclusion of the performance evaluation step, we believe that our congestion control approach within VANETs satisfies the objectives identified in Section 2.3. Mainly, fast and reliable communication scheme is guaranteed for the control channel, and an acceptable communication rate is ensured through the service one, according to the priorities of the messages and the QoS policies established by the applicative layer.

5. Conclusions and Future Work

We considered in this paper the congestion control issue within vehicular ad hoc networks. We summarized in a first step existing research work on this topic, then we presented our cooperative congestion control approach, based on dynamic messages scheduling and transmission, considering the network load and the neighborhood context. Then, we validate the efficiency and the real operability of our congestion control technique through two principal steps: formal verification and validation, and performance evaluation and analysis. The formal verification step, carried out via the UPPAAL tool, proved the reliability of our congestion control technique in terms of reachability, safety, liveness, and no-deadlock properties, whereas the performance evaluation step considers the delays of messages before their effective sending in the appropriate queue, and the service packets-loss rate.

We elaborated our congestion control approach in the context of the SAFESPOST European project. This project aims to establish a reliable communication architecture within vehicular ad hoc networks by conceiving an intelligent cooperative system able to ensure safety services to vehicles drivers, such as line change assistance, safe distance and speed evaluation, ... (cf. Figure 17).

In this context, we have developed a congestion control module, setting up the priority-based scheduling and transmission techniques within VANET. Figure 18 shows our congestion control module, and its interaction with the other modules of a VANET SAFESPOT router. We present hereafter the principles of our module and its interactions.

Congestion Control Module. Within the congestion control module, two queues are implemented, one for the control channel messages and one for the service channel messages. The dotted edge in Figure 18 represents the possibility of switching messages from the service channel queue to the control channel one, when needed. In addition, four threads are implemented, a main thread receiving messages from the input modules, and the others are intended to schedule and send messages, and control the charge of the node.

Input Modules. Input modules redirect messages to the congestion control module. The message manager module generates new messages, to be sent within the control or service channel, and the reception engine sends received messages to be forwarded by the congestion control module.

Output Module. The output module that receives messages from the congestion control module is the transmission module. Each message received from the congestion control module is affected to the corresponding channel to be effectively aired.

Parameters Modules. The congestion control module interacts with the parameters modules. From the neighbor table module, the congestion control module evaluates the network and the neighborhood context, to recompute dynamically the messages priorities. The congestion control module interacts also with QoS policies module in view of the general quality of service policies, according to the priority of the transmitted messages.

As future work, we plan to carry out real tests and measurements, in order to validate the implementation of our approach, and its interactions within the other modules of the VANET SAFESPOT router, according to the following steps.

(i) Subsystem tests. This first step consists of testing the congestion control component individually (tests of each method according to its expected values and real results).

(ii) Integrated system tests. This step consists of testing the VANET router architecture, and the interactions between the different sub-systems.

(iii) Vehicles tests. This final step consists of testing several VANET routers to validate the operational distributed behaviour of all the VANET network. A special attention will be given to the evaluation of the bandwidth consumption and the charge of the nodes, in order to validate the reliable transmission of emergency messages within VANET.

References

[1] Y. Zang, "Study on message dissemination algorithms for cooperative danger warning applications based on inter-vehicle communications," in *Proceedings of the 3rd International Workshop on Wireless Community Networks (COMNETS '08)*, Hangzhou, China, 2008.

[2] Y. Zang, L. Stibor, and H. J. Reumerman, "Neighborhood evaluation of vehicular ad-hoc network using IEEE 802.11p," in *Proceedings of the 8th European Wireless Conference*, p. 5, Paris, France, 2007.

[3] E. Minack, *Evaluation of the influence of channel conditions on Car2X communications*, Ph.D. thesis, Chemnitz University, November 2005.

[4] C. Lochert, B. Scheuermann, and M. Mauve, "A survey on congestion control for mobile ad hoc networks," *Wireless Communications and Mobile Computing*, vol. 7, no. 5, pp. 655–676, 2007.

[5] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "Feedback based scheme for improving TCP performance in ad-hoc wireless networks," in *Proceedings of the 18th International Conference on Distributed Computing Systems*, pp. 472–479, Amsterdam, The Netherlands, May 1998.

[6] H. K. Rath, A. Sahoo, and A. Karandikar, "Cross layer based congestion control in wireless networks for TCP Reno-2," in *Proceedings of the 12th National Conference on Communications (NCC '06)*, Delhi, India, April 2006.

[7] M. Chen, A. Abate, and S. Sastry, "New congestion control schemes over wireless networks: delay sensitivity analysis and simulations," in *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, January 2005.

[8] V. Raghunathan and P. R. Kumar, "A counterexample in congestion control of wireless networks," in *Proceedings of the 8th ACM Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '05)*, pp. 290–297, Montreal, Canada, October 2005.

[9] Y. Yi and S. Shakkottai, "Hop-by-hop congestion control over a wireless multi-hop network," *IEEE/ACM Transactions on Networking*, vol. 15, no. 1, pp. 133–144, 2007.

[10] D. Kliazovich and F. Granelli, "Cross-layer congestion control in ad hoc wireless networks," *Ad Hoc Networks*, vol. 4, no. 6, pp. 687–708, 2006.

[11] S. Rangwala, A. Jindal, K.-Y. Jang, K. Psounis, and R. Govindan, "Understanding congestion control in multi-hop wireless mesh networks," in *Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MOBICOM '08)*, pp. 291–302, San Francisco, Calif, USA, September 2008.

[12] G. Zhang, Y. Wu, and Y. Liu, "Stability and sensitivity for congestion control in wireless networks with time varying link capacities," in *Proceedings of the 13th IEEE International Conference on Network Protocols (ICNP '05)*, pp. 401–411, Boston, Mass, USA, November 2005.

[13] F. Kelly, "Charging and rate control for elastic traffic," *European Transactions on Telecommunications*, vol. 8, no. 1, pp. 33–37, 1997.

[14] Y. Zang, L. Stibor, X. Cheng, H.-J. Reumerman, A. Paruzel, and A. Barroso, "Congestion control in wireless networks for vehicular safety applications," in *Proceedings of the 8th European Wireless Conference*, p. 7, Paris, France, 2007.

[15] L. Wischhof and H. Rohling, "Congestion control in vehicular ad hoc networks," in *Proceedings of IEEE International Conference on Vehicular Electronics and Safety Proceedings*, pp. 58–63, Xian, China, October 2005.

[16] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Fair sharing of bandwidth in VANETs," in *Proceedings of the Second ACM International Workshop on Vehicular Ad Hoc Networks (VANET '05)*, pp. 49–58, Cologne, Germany, September 2005.

[17] IEEE P1609.4, "Wireless access in vehicular environments (wave) multichannel operation," in *Draft Standard*, November 2005.

[18] R. Alur and D. L. Dill, "A theory of timed automata," *Theoretical Computer Science*, vol. 126, no. 2, pp. 183–235, 1994.

[19] A. Cavalli, *Ingénierie des protocoles et qualité de service, Réseaux et Télécoms*, Lavoisier Hermes, Paris, France, 2001.

[20] A. David, "Uppaal2k: small tutorial," Report Version 3.2.11, October 2002.