I. J. Computer Network and Information Security, **2013, 4, 1-7**
Published Online April 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2013.04.01

Formal Verification of Congestion Control Algorithm in VANETs

Mohamad Yusof Darus
Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia (UTM), Malaysia
yusof@tmsk.uitm.edu.my

Kamalrulnizam Abu Bakar
Faculty of Computer Science & Information Systems, Universiti Teknologi Malaysia (UTM), Malaysia
knizam@utm.my, kamarul@fsksm.utm.my

Abstract — A Vehicular Ad-Hoc Networks (VANETs) is the technology that uses moving cars as nodes in a network to create a mobile network. VANETs turn every participating car into a wireless router, allowing cars of each other to connect and create a network with a wide range. VANETs are developed for enhancing the driving safety and comfort of automotive users. The VANETs can provide wide variety of service such as Intelligent Transportation System (ITS) e.g. safety applications. Many of safety applications built in VANETs are required real-time communication with high reliability. One of the main challenges is to avoid degradation of communication channels in dense traffic network. Many of studies suggested that appropriate congestion control algorithms are essential to provide efficient operation of the network. However, most of congestion control algorithms are not really applicable to event-driven safety messages. In this paper we propose congestion control algorithm as solution to prevent congestion in VANETs environment. We propose a complete validation method and analyse the performance of our congestion control algorithms for event-driven safety messages in difference congested scenarios. The effectiveness of the proposed congestion control algorithm is evaluated through the simulation using Veins simulator.

Index Terms — VANETs, Congestion control, Event-driven safety message, Beacon messages, CCH channel

I. INTRODUCTION

With VANETs (Vehicle Ad-Hoc Networks) are composed of vehicles equipped with advanced wireless communication devices without any base stations. Each vehicle equipped with VANETs device will be a node in the ad-hoc network and can receive and relay others messages through the wireless network. This type of network can provide wide variety of services such as Intelligent Transportation System (ITS) [1,2]. The safety application is one of the most crucial application in ITS. The safety messages must deliver to each neighbouring node without any delay. Basically, we can divide safety applications in two (2) categories; periodic (beacon) messages and event-driven safety messages.

The periodic safety message exchange is preventive in nature, and the objective is to avoid the occurrence of dangerous situations. The periodic safety message may contain information regarding the position, direction, and speed of vehicles. The event-driven safety messages may be generated as a result of a dangerous situation or when an abnormal condition is detected such as road accident [1, 3]. The event-driven safety messages disseminated within a certain area with high priority. The event-driven safety messages usually have strong reliability and need to be delivered to each neighbor with almost no delays. Both of these safety messages will send through one single channel kwon as Control Channel (CCH), which is restricted to safety communications only.

In high traffic, a large number of vehicles broadcast beacon safety messages at a high frequency or event-driven safety messages are broadcast multiple times; the CCH channel will easily congested. This situation will decrease a throughput while delay is increasing significantly. On the other hand, the simplest way of broadcasting beacon safety messages to all nodes is using blind flooding. Each node that received the packets will rebroadcast this packet and will lead the broadcast storm. In VANETs, it is based on the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) scheme [3, 4, 5]. In CSMA/CA, when a node received a packet that is transmitted, it senses to make sure the channel is clear (no other node is transmitted at the time). If the channel is idle, then the packet is sent. If the channel is busy, the node must defer its access and during high utilization periods this could lead to unbound delayed, it will affect the safety applications [4]. Furthermore, blindly broadcasting the packet may lead to frequent contention and collisions in transmission among neighbouring nodes. Moreover, a node can experience very long channel access delay due to the risk of the channel being busy during its listening period messages [6]. It will affect the performance of safety applications in VANETs.

This paper aims to develop congestion control algorithm to provide reliability for disseminating event-driven safety messages. Then our algorithm is validated through formal verification simulation using UPPAAL integrated tool. Finally, the performance of proposed congestion control algorithm evaluated through Veins simulator based on warning packet delay.

The rest part of the paper is organized as follows: In section II, we present related work concerning congestion control algorithms within VANETs. Section III, we discussed our congestion control algorithms. The verification and performance evaluations are presented in section IV. Finally, section V concludes the paper with outlooks on the future work.

II. Related works

There have been several works addressing the congestion problem in VANET. In research [7], they developed a congestion control approach based on the concept of dynamic priorities-based scheduling. They evaluated dynamic priority factor based on: node speed consideration, message utility consideration and message validity consideration. This approach required context exchange between neighbour nodes, which generates a communication overhead. On other hand, the congestion control algorithm for event-driven safety messages is proposed in [8]. This congestion control evaluated the performance of the Safety Electronic Brake Light with Forwarding (EEBL-F). Research in [8] set the predefined threshold in their congestion control algorithm based on channel usage level. Each device periodically senses the channel usage level, and detects the congestion whenever the measured channel usage level exceeds the predefined threshold. Measuring the channel usage level is too difficult to analyse under realistic environment due of the different traffic load. In a similar study in [9], they proposed congestion control algorithm for DSRC based on safety applications. However, they just assumed the CCH channel is successfully reserved for event-driven applications without testing the successfully rate for event-driven safety messages. In research [9], they set the channel occupancy time as threshold. If channel occupancy time measured at a node in CCH interval is longer than a given threshold, all beacon safety messages will be blocked immediately in the remainder of that CCH interval and the CCH interval followed to reduce channel load and reserve space for event-driven safety messages. Measuring the channel with the channel occupancy time is too difficult to analyse and needs the proper rate control design.

III. The Proposed Congestion Control Algorithm

In this research, we proposed congestion control algorithm that can ensure high reliability and timely delivery of disseminating event-driven safety messages. Our propose congestion control algorithm can be divided into two main parts: event-driven detection and measurement-based detection. In Figure 1 shows the flowchart steps of the proposed congestion control algorithm. Details on Measurement-Based Detection and Event-Driven Detection that has been used in the flowchart will be explained later.

A. Measurement-Based Detection

The measurement-based congestion detection will monitor CCH channel based on packets channel queue. The CCH channel is congested if the number of messages in the queue exceeded a defined threshold. Based on research in [1] is concluded that a queue with a length of five beacon messages is sufficient to be used for 802.11p beaconing. In our proposed congestion control algorithm, the congestion control will discard a beacon safety messages whenever the length of packet queue more than five beacon safety messages.

B. Event-Driven Detection

The event-driven detection method monitors the event-driven safety message and decides to start the congestion control algorithm whenever event-driven safety message is detected or generated. The congestion control will launch immediately the queue freezing method for all MAC transmission queues except for the event-driven safety message. In order to send event-driven safety message with the minimum delay, the lower priority messages such as beacon messages emission is freeze. Currently, the event-driven detection method has been used in the existing of congestion control algorithm [8].

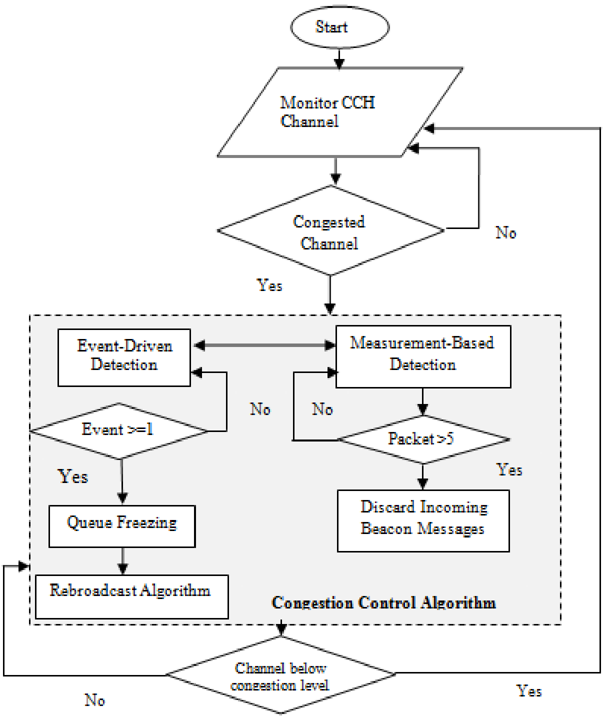

Figure. 1. Flowchart steps of the proposed congestion control algorithm

V. Simulation and experiment

A. Formal verification

In order to verify our congestion control algorithm, we propose to specify it using the UPPAAL integrated tool. This tool was developed in collaboration between the Department of Information Technology at Uppsala

University, Sweden and the Department of Computer Science at Aalborg University in Denmark. The formal verification has been extensively used to analyze congestion control algorithm in vehicular networks [2, 7, 10].

We divided our congestion control algorithm into four independent sub-systems: message manager, message queue, message dequeue and engine.

a. *The Message Manager*

This sub-system is responsible for generating messages and transferring messages to the congestion control module. The congestion control module will process messages according to their priorities such as safety messages or comfort messages. In this research, we consider two level priorities of safety messages which are event-driven messages and beacon messages. The event-driven safety message has highest priority compared to beacon safety messages. The both of safety messages transmit are through CCH channel and comfort messages are transmit throughout SCHs channel. This sub-system illustrated in Fig 2.

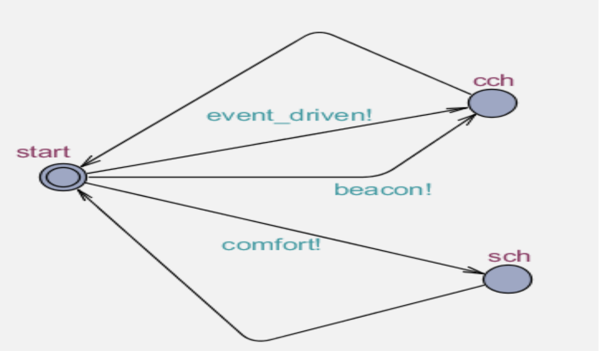

Figure 2. Message Manager Sub-System

b. *The Message Queue*

The message queue sub-system is responsible for the reception of messages from the message manager sub-system. The event-driven safety messages and beacon safety messages are required to transit from the *Idle* state to the *New_CCH_Msg* state. At the *Idle* state, the congestion control will identify the priority message whether it's event-driven, beacon or comfort messages. The automaton switches the both of safety messages to the *New_CCH_Msg* state whenever the comfort messages switch to *New_SCH_Msg* state.

In this research, the proposed congestion control algorithm defined the threshold based on packet queue. If the beacon messages in packet queue exceed the predefined threshold, the congestion control will discard extra incoming beacon safety messages. The CCH channel is congested whenever threshold detected beacon messages in packet queue more than five beacon messages. Fig. 3 illustrated process in sub-system message queue.

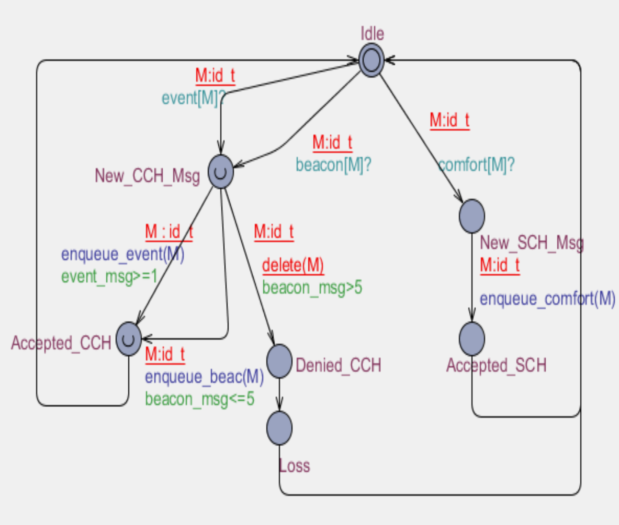

Figure 3. Message Queue Sub-System

c. *Message Dequeue*

The congestion control message dequeue sub-system are illustrated in Fig. 4. The message dequeue sub-system are responsible for withdrawing messages from accessing the CCH channel queue and service channels queues. In the propose congestion control, the beacon messages are discarding if the beacon messages more than five packets in packet queue. The message dequeue sub-system also can transmit messages to the engine sub-system. This process needs synchronization over channels, between the congestion control message dequeue sub-system and the engine sub-system are called as *send_ctrl* and *send_serv*. The synchronize processes is done by annotating edges in the model with synchronisation labels. Synchronisation labels are syntactically very simple. They are of the form M? or M!, where e is a side effect free expression evaluating to a channel.

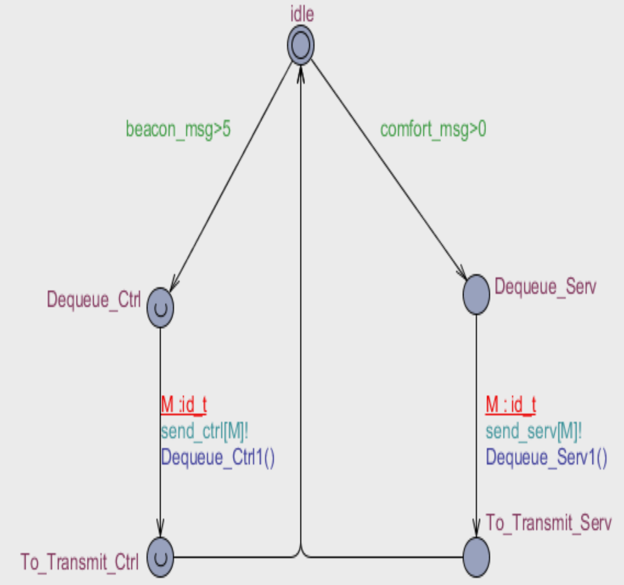

Figure 4. Message Dequeue Sub-System

d. *Engine*

The engine sub-system is illustrated in Fig. 5. The main function of engine sub-system is responsible for the successfully messages transmission on CCH and SCHs communication channels. If an error detects during message sending, the automaton switches to the *Error* state. This sub-system is very important to verify efficiency of our proposed congestion control. In this research, we assumed the sending error rate is 5%. The percentage sending rate is defined in SAFESPOT European Integrated project.

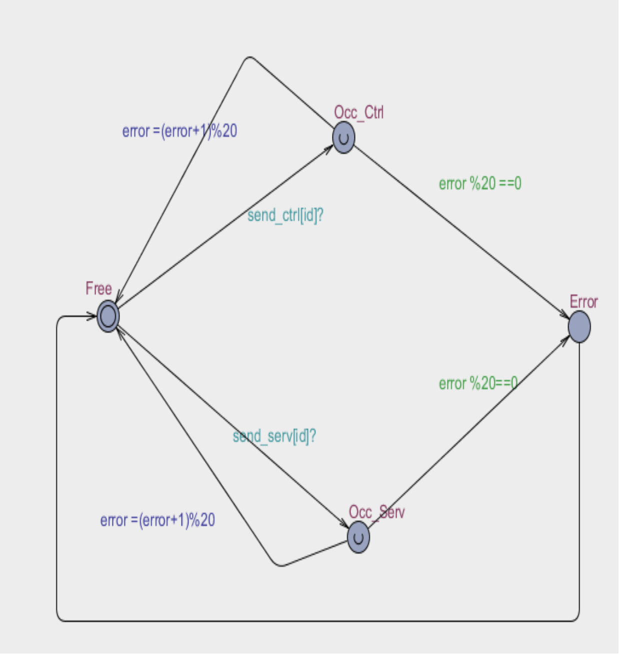

Figure 5. Engine Sub-System

To verify and validate our proposed congestion control, we randomly simulates all the possible transitions of the four sub-system; message manager, message queue, message dequeue and engine. In the configuration, the following aspects of networks are considered:

i. No deadlock. The state property is deadlock state if there are no outgoing action transitions neither from the state itself or any of its delay successors. All states of the model have successors.

ii. Reachability Properties. The reachability properties are the simplest form of properties. The properties ask whether for a given state formula φ, there exists a path starting at the initial state, such that is φ eventually satisfied along that path. In UPPAAL, when creating a model of a communication protocol involving a sender and a receiver, it makes sense to ask whether it is possible for the sender to send a message at all or whether a message can possibly be received.

iii. Differentiate three types of messages; a) the event-driven safety message b) beacon safety messages and c) comfort messages. The event-driven safety message has the highest priority.

iv. The event-driven and beacon safety messages are effectively sent on the CCH communication channel and comfort messages sent through the SCH communication channels.

v. The dissemination of event-driven safety message are considered as critical message, therefore all lower priority messages will be freeze.

Figure 6 shows the Event-Driven Safety Packet Sending Diagram and Figure 7 shows the Beacon Safety Packet Sending Diagram. Both figure are the examples of result from the execution packets sending. These diagrams are generated from simulation in UPPAAL tool. Figure 6 illustrates of a high priority packet sending which is event-driven safety message. When message manager sub-system detected event-driven packet (*event [0]*, the message manager sub-system transmit to *idle* state in message queue sub-system. After that, the *idle state* in message queue sub-system transmits to *New_CCH_Msg* state, then the *Accepted_CCH* state transmit to message dequeue sub-system. In fig 6, the event-driven safety message (*send_ctrl[M]*) was successfully disseminated.

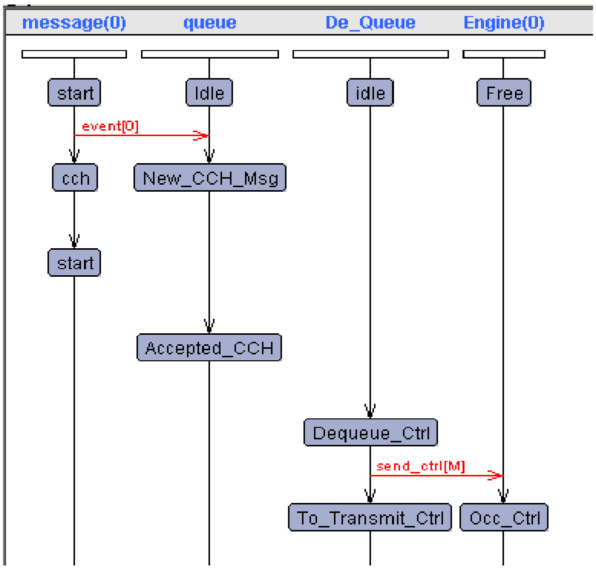

Figure 6. Event-Driven Safety Packet Sending Diagram

The Fig. 7 illustrated the sending of beacon messages. If message manager sub-system detected beacon packets (*beacon [0]*, the message manager sub-system transmit to *idle state* in message queue sub-system. After that, the *idle state* in message queue sub-system transmits to *New_CCH_Msg* state, then the *Accepted_CCH* state transmit to message dequeue sub-system. In this research, we assumed the CCH channel is congested if the beacon packet queue exceeds the predefined threshold. The incoming beacon packets will discard from accessing the CCH channel. The result shown the beacon safety messages (*send_ctrl[M]*) was successfully disseminated through CCH channnel.

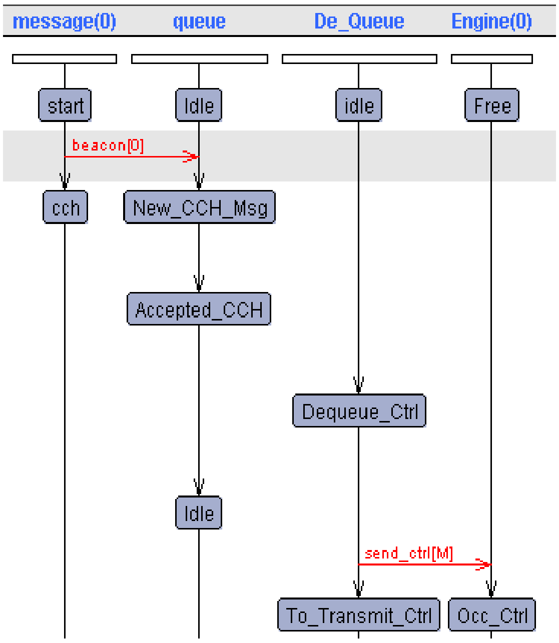

Figure 7. Beacon Safety Packet Sending Diagram

The Fig. 8 illustrated the successfully sending of comfort packets through SCHs channels. The message manager sub-system is responsible for processing messages according to their priorities. If the message manager sub-system detected comfort messages, the message manager sub-system transmit to *idle* state in message queue sub-system. After that, the *idle* state in message queue sub-system transmits comfort messages to *New_SCH_Msg* state, then the *Accepted_SCH* state transmit to *Dequeue_Serv* state in message dequeue sub-system. However, this research focus on disseminating event-driven safety messages rather than comfort messages.

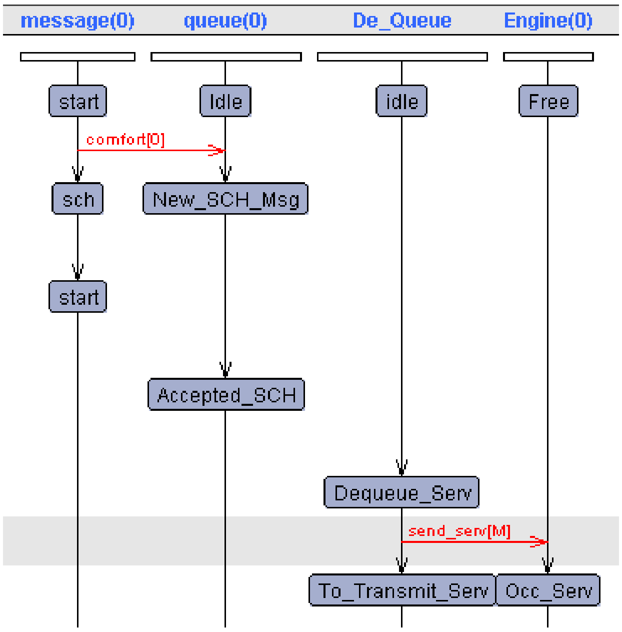

Figure 8. Comfort Packet Sending Diagram

The Fig. 6, Fig. 7 and Fig. 8 are shown free errors during messages transmission (event-driven, beacon and comfort) in CCH and SCHs communication channels. Based on the results, the proposed congestion control algorithm is competent to handle congestion especially in dense vehicular networks. However, although this suggestion has been made, verification apparently did not succeed due to memory machine limitations.

B. *Simulation results*

This section evaluates the performance of our proposed congestion control algorithm through simulation experiments using Veins simulator. The Veins simulator make up of two distinct simulators, OMNeT++ for network simulation and SUMO for road traffic simulation. The simulation scenario is a 1500m x 1500m area of Los Angeles, extracted from the TIGER/Line database of the US Census Bureau. The simulation key parameters are summarized in Table 1.

Table 1: Simulation Parameters

Parameter	Value
Simulation area	1500 m x 1500 m
Number of vehicles	150-250-350
Packet type	UDP
Node Speed	40-88 km per hour
Transmission range	400 m
Simulation time	300 s
Data packet size	512 bytes
MAC protocol	IEEE 802.11p

The vehicles are embedded with wireless radio communication in order to facilitate multi-hop communications. Similar to existing works on VANET [11, 12], it is assumed that vehicles are equipped with a GPS receiver that provides the vehicles current position to all neighbors. All vehicles are embedded with digital road maps which provide accurate locations of roads. It is also assumed that the location discovery as provided by GPS is highly precise. Since GPS error resolution is typically within 15 meters, this assumption is fair. The vehicle can observe its neighbors position and direction information by periodically broadcasting beacon messages to vehicles within its communication range.

By varying the simulation parameters, different experiments are conducted in different number of vehicle nodes which are 350 nodes for dense networks, 250 nodes for medium density networks and 150 nodes sparse networks. The purpose of testing in different number of vehicle nodes is to check the impacts of our congestion control in different traffic densities. The comparison studies are performed between the proposed congestion control algorithm and without congestion control based on warning packet delay. The warning packet delay is defined the delay between the time the first event-driven safety message being sent out by the accident vehicle and the time the message being received for the first time at each concerned vehicle

In Fig. 9 presents the warning packets delay for event-driven safety messages in sparse network. The result simulation shows that when the number of vehicles is increased, the packet delay consistently increased. The simulation result shows that event-driven safety packet starting delay whenever the number vehicles more than 60. In sparse network, the maximum packet delay for event-driven safety applications is 21 ms without congestion control and 8 ms for the proposed congestion control. Based on this result, the performance of event-driven safety messages are very satisfies although without congestion control algorithm.

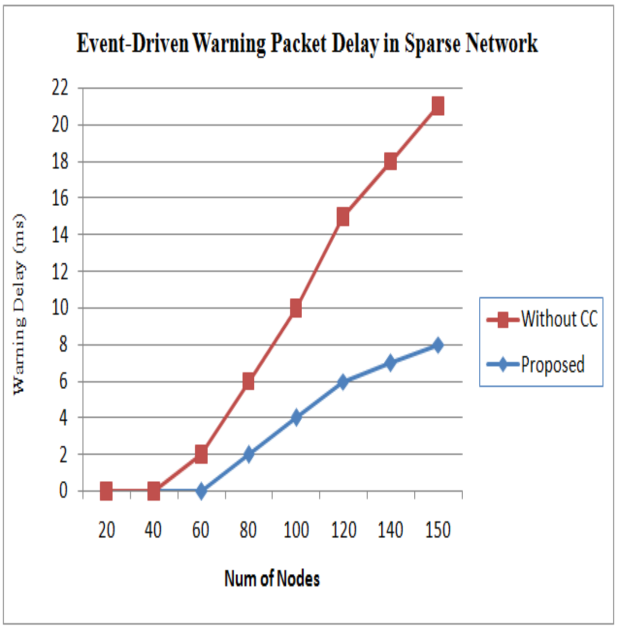

Figure. 9. The warning packets delay for event-driven safety messages in sparse network

The Fig. 10 illustrated the warning packets delay for event-driven safety versus the number of nodes in medium traffic density. The result obviously shows the delay of event-driven safety messages are related to the numbers of vehicles. Based on this simulation testing, the proposed congestion control shows the great peformance of QoS in medium network. In the proposed congestion control the warning packets delay is 21 ms and the 48 ms for the without congestion control. However, in medium network density, the warning packets delay in congestion control and without congestion control still satisfies and not exceed in worst cases which is 60 ms.

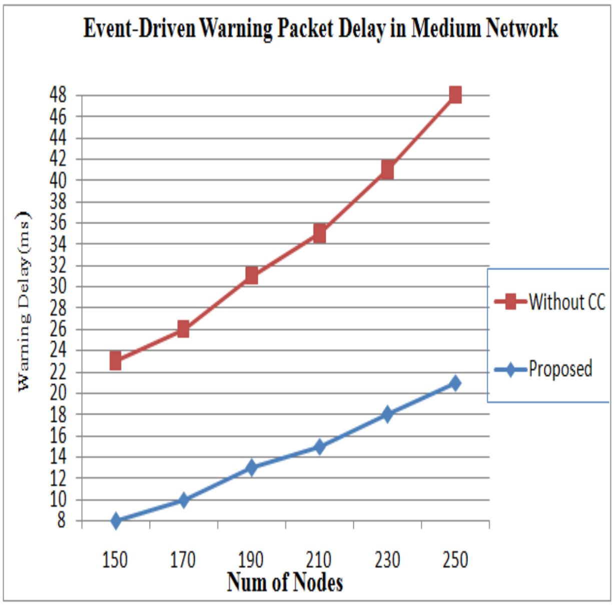

Figure. 10. The warning packets delay for event-driven safety messages in medium network

Fig. 11 illustrated the warning packets delay for event-driven safety messages in high traffic density. The result simulation shows the proposed congestion control is very efficient for event-driven safety applications in dense network, the maximum warning packet delay only 37 ms compared 78 ms for without congestion control and exceed in worst cases which is 60 ms. The results have been proved that an efficient congestion control algorithm is efficient method for disseminating event-driven safety applications in dense network. The implementation of efficient congestion control is significantly increases the efficiency of disseminating of event-driven safety messages.

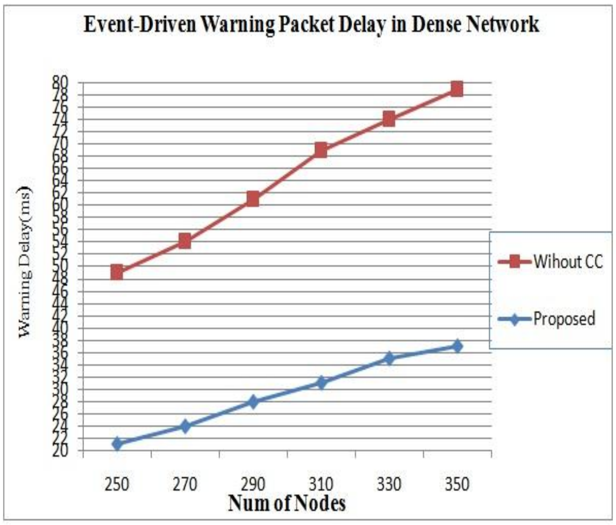

Figure. 11. The warning packets delay for event-driven safety messages in dense network

VI. Conclusions

In this paper, we studied the congestion problem in favour of event-driven safety messages in VANETs environment. The proposed congestion control algorithm has been proved as an efficient method for disseminating event-driven safety messages in dense network scenario. With extensive and fair simulation results show that our proposed congestion control algorithm performs the best in terms of packets delay ratio. The event-driven safety messages must deliver to each neighbouring node without any delays. A single delayed of event-driven safety messages could result in loss of life.

As the next step of this work, we would like to integrate our proposal with the efficient transmit power control to maximize energy consumption and connectivity for point-to-point communications. Furthermore, we also plan to explore uni-priority of event-driven safety messages in VANETs. The uni-priority packet is caused by the traffic of the same high priority, typically the event-driven safety messages of safety applications from different transmitters

REFERENCES

[1] Hendriks, L., Effects of Transmission Queue Size, Buffer and Scheduling Mechanisms on the IEEE 802.11p Beaconing Performance, 15th Twente Student Conference, University of Twente, Faculty of Electrical Engineering, Mathematics and Computer Science. (2011).

[2] Konur, S. and Fisher, M., Formal Analysis of a VANET Congestion Control Protocol through Probabilistic Verification, *In proceeding IEEE 73rd Vehicular Technology Conference (VTC Spring)*, Budapest, (2011), pp. 1 – 5

[3] Li, M., Lou, W., & Zeng, K., OppCast: Opportunistic Broadcast of Warning Messages in VANETs with Unreliable Links. In Proc. IEEE 6th International Conference (MASS '09). (2009), 534 – 543

[4] Campolo, C., Cortese, A., & Molinaro, A., CRaSCH A Cooperative Scheme for Service Channel Reservation in 802.11p/WAVE Vehicular Ad Hoc Networks. In Proc. International Conference on Ultra Modern Telecommunications & Workshops. (2009), 1-8

[5] Wang, Y., Ahmed, A., Krishnamachari, B., & Psounis, K., IEEE 802.11p Performance Evaluation and Protocol Enhancement. In Proc IEEE International Conference on Vehicular Electronics and Safety Columbus. (2008), 317-322.

[6] Bilstrup, K. , Uhlemann, E., Strom, E.G. & Bilstrup, U., Evaluation of the IEEE 802.11p MAC method for Vehicle-to-Vehicle Communication. In Proc. IEEE 68th Vehicular Technology Conference.(2008), 1-5

[7] Bouassida, M.S., and Shawky, M., On The Congestion Control Within Vanet. 1st IFIP Wireless Days.(2008), 1-5.

[8] Zang, Y.P., Stibor, L., Cheng, X., Reumerman, H. J., Paruzel A., & Barroso, A., Congestion Control in Wireless Networks for Vehicular Safety Applications. *In Proc. of The 8th European Wireless Conference*. (2007), 7.

[9] He, J., Chen, H.C. et al., Adaptive Congestion Control for DSRC Vehicle Networks, IEEE Communications Letters. (2010), Vol. 14, No. 2.

[10] Lomuscio, A., Strulo, B. et al. (2009). Model Checking Optimisation Based Congestion Control Models. *Research notes from University College London*, 2009

[11] Sommer, C., Tonguz, O. and Dressler, F., Traffic Information Systems: Efficient Message Dissemination via Adaptive Beaconing. *IEEE Communication Magazine*.49, (2011), 173 – 179. IEEE.

[12] Tsao, S.L. and Cheng, C.M., Design and Evaluation of A Two-Tier Peer-To-Peer Traffic Information System. *IEEE Communication Magazine*. 49, (2011), 165 – 172. IEEE.

Mohamad Yusof Darus is currently a PhD student in the Department of Computer Systems and Communications of the Faculty of Computer Science and Information Systems at the Universiti Teknologi Malaysia. He obtained M.Sc. Information Technology from Universiti Utara Malaysia (Malaysia) in 2003. He has been involved in lots of academic research since then; presently he is a member of Pervasive Computing Research Group at UTM, while his research interest Vehicular Networks (VANETs). He has published in many national and international learned journals.

Kamalrulnizam Abu Bakar obtained his PhD degree from Aston University (Birmingham, UK) in 2004. Currently, he is an Associate Professor in Computer Science at Universiti Teknologi Malaysia (Malaysia) and member of the "Pervasive Computing" research group. He involves in several research projects and is the referee for many scientific journals and conferences. His specialization includes mobile and wireless computing, information security and grid computing