



ISSN NO. 2320-5407

Journal Homepage: - www.journalijar.com
**INTERNATIONAL JOURNAL OF
 ADVANCED RESEARCH (IJAR)**

Article DOI: 10.21474/IJAR01/1682
 DOI URL: <http://dx.doi.org/10.21474/IJAR01/1682>



RESEARCH ARTICLE

VERIFICATION OF CONGESTION CONTROL PROTOCOL IN VANETs USING FORMAL METHOD

*Ayana S and Shimmi Asokan.

Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology Kochi, India.

Manuscript Info

Manuscript History

Received: 12 July 2016
 Final Accepted: 13 August 2016
 Published: September 2016

Key words:-

Congestion, Model Checking,
 Timed Automata, UPPAAL, VANET

Abstract

Congestion control protocols developed in order to resolve the issue of congestion in VANETs must ensure reliable communication of safety-related messages. In this work we focused on formally verifying congestion control protocol in VANETs using model checking technique. The protocol was formally modeled using a network of timed automata. The model was verified against reachability, safety and liveness properties to analyze the reliability and functional aspects of the protocol. A model checking tool UPPAAL was used for formally modelling, simulating and verifying the protocol.

Copy Right, IJAR, 2016., All rights reserved.

Introduction:-

VANETs were introduced for the purpose of enabling a wireless based communication between vehicles as well as between vehicles and roadside equipment. Messages communicated between vehicles can be safety-related as well as non-safety-related information. When the rate of traffic on roads increases messages are generated with an increase in frequency. Overload of messages in communication channel would result in degradation of communication channel thereby creating a delay in transmission of messages. As a solution to congestion, congestion control protocols were developed.

A congestion control mechanism should be developed taking into consideration the different types of messages in VANET and their priorities. It should also take into account the limited bandwidth as well as mobility of vehicles into consideration. A congestion control protocol should also ensure an efficient and reliable communication of messages.

An exhaustive analysis of congestion control protocol needs to be done as a failure of the protocol may question the safety of human, vehicle as well as road-side equipment. In this paper, we have proposed to verify the congestion control protocol using formal methods. Model checking, an automated formal verification technique was used to carry out verification process. Use of formal verification technique helps in mathematically proving the correctness of the congestion control protocol.

In this paper, we have aimed to formally verify Congestion Control Protocol in VANETs using statistical model checking. A statistical model checking tool was used to carry out the formal verification process. The protocol was formally modeled. The properties that the congestion control protocol must satisfy were formally specified. These properties were then checked over the model to check if the properties were satisfied by the model or not.

Corresponding Author:- Ayana S, Shimmi Asokan.

Address:- Department of Computer Science and Engineering, Rajagiri School of Engineering and Technology Kochi, India.

Related Works:-

There have been few works addressing congestion control protocol verification and analysis. In the work by Boussida and Shawky [2], a congestion control mechanism was proposed. The congestion control protocol was based on scheduling and transmitting messages dynamically. The protocol was then formally verified using the model checking technique. The model was specified using timed automata. UPPAAL tool was used for modeling, simulating and verifying. The verification was done in two phases. The First phase was through simulation using graphical interface. All possible transitions were simulated here. The second phase of verification was done by specifying queries. This step did not succeed due to memory limitations. However, considering the first step of verification the approach was taken to be correct.

In the work [4], a formal analysis of CC protocol proposed in [2] was done through probabilistic verification. A probabilistic model checking tool, PRISM was used to carry out verification process. The effectiveness and correctness of the protocol were analyzed quantitatively. Slight improvisation was proposed to the protocol. Improvisations included allowing safety messages to use the entire available bandwidth and introduced preemption in the transmission of high priority messages to reduce the delay in transmission. It also proposed that service messages have to use only the service channel for transmission. On evaluation of the protocol, a better loss rate of safety messages was obtained.

In the work [3], a congestion control protocol was proposed which was based on event driven and measurement based congestion control mechanism. In order to verify the protocol, UPPAAL framework was used. The protocol was modeled using a network of timed automata. All possible transitions were randomly simulated in order to verify the protocol. Verification using model checking failed due to memory limitations. Validation of the protocol was then carried out through simulation experiment using Veins simulator. The simulation results showed that the congestion control protocol in terms of packet delay ratio performed the best.

In work [7], formal verification of CC protocol was carried out using PRISM probabilistic model checker. The Quality of Service (QoS) metrics were analyzed in this work. A more realistic model of the protocol was created. Markov Population Process (MPP) was used to formally model the protocol. Use of stochastic model checking technique produced results with high degree of confidence.

Congestion Control Protocol in VANET:-

In a Vehicular Ad-Hoc Network, messages are transmitted between vehicles that fall within a communication range. Messages are prioritized based on the application type of messages. This paper discusses three types of messages which are event driven messages, beacon messages and service messages. Messages that are generated due to the occurrence of an event such as, an accident or vehicle breakdown needs to be transmitted without any delay. These messages are treated with the highest priority. Periodic safety messages are generated so as to communicate speed, location and direction of vehicles. These messages are used by vehicles to compute the distance between other vehicles from itself. The lowest priority is assigned to service messages which are communicated by road-side equipment to the vehicle. The high priority messages, event driven messages and the beacon messages are transmitted through the control channel (CCH). Low priority service messages are transmitted via service channel (SCH). When the vehicular density is high (during peak hours), the frequency of periodic message generation is more because which may cause the communication channel getting congested. This situation is hazardous as this might result in a delay in transmission of high priority messages. Congestion control protocols were introduced therefore to ensure reliable and efficient communication of messages.

The congestion control protocols as proposed in ^[3] were divided into two categories, namely event driven detection and measurement based detection. According to event based detection, transmission of lower priority message is frozen on detection of a higher priority message. Priority is given to high priority message transmission so that these messages are transmitted to the destination without any delay. In measurement based detection approach the channel usage level is sensed and when the channel usage level goes beyond a certain threshold the messages are discarded. These two phases were formally modeled using timed automata. The measurement based detection phase is incorporated in enqueueing automata (Figure 3) and event driven based detection is modeled using dequeuing automata (Figure 4).

Formal Modeling and Verification:-

The congestion control algorithm was formally modeled using timed automaton. The timed automaton was specified using UPPAAL tool. For modeling, the protocol was divided into five subsystems namely, the vehicle automaton, infrastructure automaton, en-queuing automaton, de-queuing automaton and engine automaton.

Modeling Phase:-

Vehicle automaton: Vehicle automaton is considered to generate high priority messages. Event driven messages and beacon messages are generated here. Beacon (periodic) messages are considered to be generated at a higher rate than the event-driven messages. A message is considered to be generated every ‘time’ interval. A clock variable ‘x’ is used for generation of message. This module is synchronized with the en-queuing automata. ‘U’ is used in locations to indicate that there should be no delay in transition to the next state. Figure 1 shows the vehicle automaton.

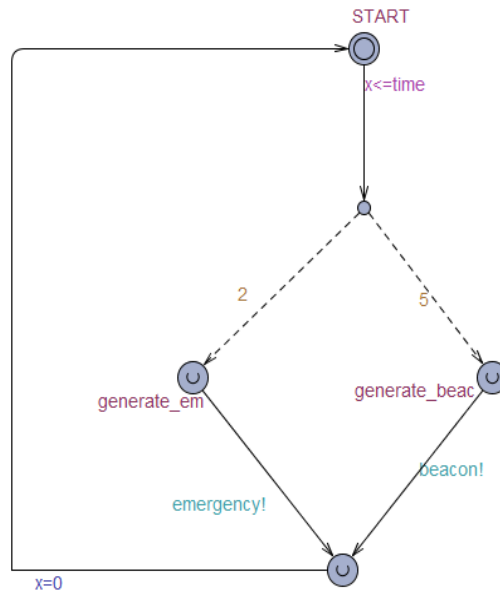


Fig.1:- Vehicle Automaton

Infrastructure automaton: This automaton is responsible for generating the periodic service messages. This subsystem is shown in Figure 2. Service messages are generated more frequently than the safety messages. It is considered a service message is generated every three units of time. This automaton is synchronized with the enqueuing automaton. A clock variable ‘z’ is used to generate a service message every three units of time.

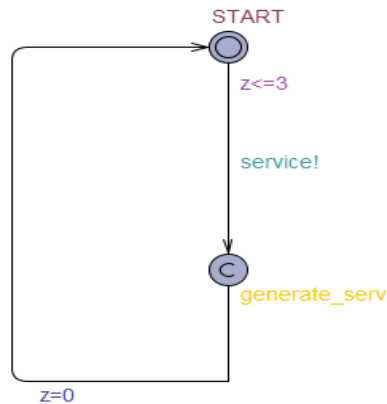


Fig. 2:- Infrastructure Automata.

Enqueuing automaton: This module is responsible for receiving the messages generated by the vehicle and infrastructure automaton. The messages received are queued to the respective queues. Messages generated by the vehicle automaton are sent to the control queue and messages from the infrastructure automaton are sent to the service queue, provided, the number of messages in the queue is less than a predefined threshold value. Measurement based congestion control mechanism is modeled here. Channel usage level is sensed and when the value goes beyond a threshold the messages are discarded. This subsystem is illustrated in Figure 3. A variable ‘cch’ is updated to 1 if the generated safety message is an event driven message. The variable is 0 if the generated message is a beacon message. On receiving beacon? signal transition to the state accept_CCH takes place if the number of messages is within 5 else a transition to deny_CCH takes place. The variable b_dropped stores the value of messages dropped.

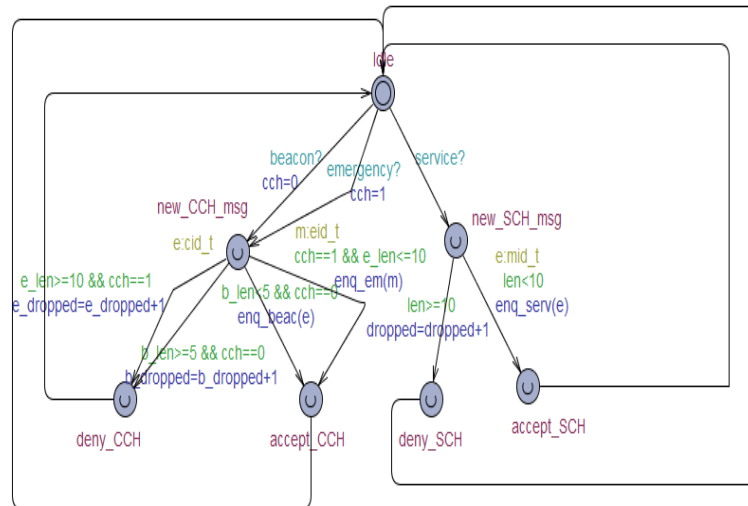


Fig 3:- Enqueuing Automata

Dequeuing automaton: Here, the messages are withdrawn from the queues and are sent to the transmission engine. Messages are transmitted based on their priority. Event- driven based congestion control mechanism is modeled by means of this subsystem. Transmission of low priority messages is frozen when an event-driven message is detected. When e_len >0 event-driven message transmission is enabled. Transmission of beacon and service messages are frozen until the value of event driven messages in the queue becomes zero. Thereby queue freezing is incorporated in the model. The dequeuing subsystem is shown in Figure 4. A clock variable y is used to enable dequeuing every two units of time. This subsystem is synchronized with the transmission engine automaton. Synchronizing signals used here are em_transmitted!, beac_transmitted!, and service_transmitted!

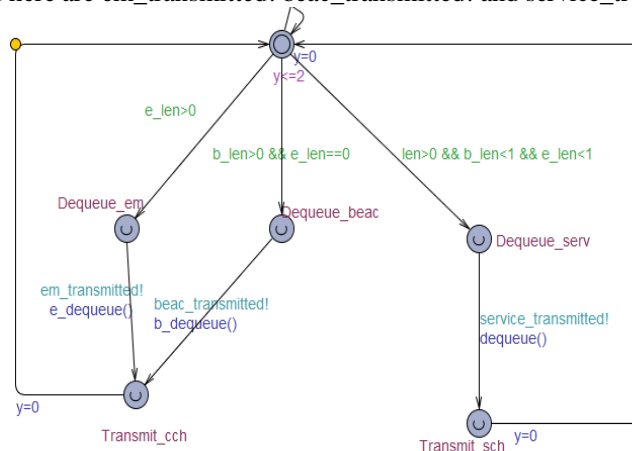


Fig. 4:- Dequeuing Automaton

Engine automaton: This module is responsible for messages successful transmission onto the respective channels. Event-driven and the beacon messages are transmitted to the control channel (CCH). Service messages are sent to the service channel (SCH). Occ_Ctrl and Occ_Serv indicate that a successful transmission onto the channel has occurred. Figure 5 shows the engine automaton.

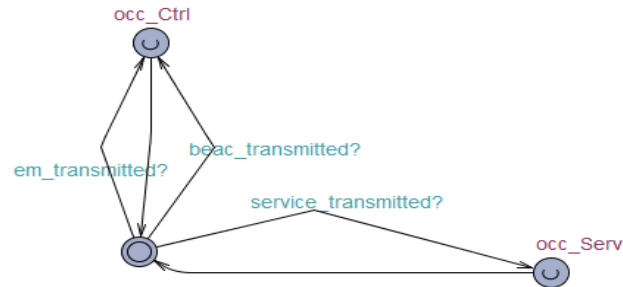


Fig 5:- Engine Automaton.

Verification Phase:-

In order to verify the reliability and the functional aspects of congestion control protocol, the following properties of congestion control protocol were identified and formalized using Computation Tree Logic(CTL) and Probabilistic Computation Tree Logic(PCTL) for verification.

1. Reachability- $E \langle \rangle \text{Engine.occ Ctrl}$ was used to verify the reachability property. This query is used to verify if there exists a path for safety messages to reach the control channel. Since safety messages are of the highest priority it is essential to ensure that these messages are effectively sent to the destination.
2. Safety- In order to analyze the safety property of protocol the following properties need to be verified
 - 2.1 The Probability of safety messages being accepted- The Query, $\text{Pr} [\leq t](\text{Enqueue .accept CCH})$ was used to verify this property. It checks the probability of control messages being accepted in 't' units of time.
 - 2.2 Maximum number of emergency messages lost- This property was verified using the formula $E[\leq 10; 10000](\text{max:e_dropped})$. This formula estimates the number of event -driven messages lost.
 - 2.3 Priority for transmission of safety messages- This property was used to verify if queue freezing was adopted in the model. The query $E \langle \rangle \text{Dequeue.Dequeue. imply e_len} \geq 0 \ \&\& \ \text{len} == 0 \ \text{or} \ \text{len} == 1$ was used to verify the property.
 - 2.4 Delay in transmission of safety messages over the channel is lesser than that of beacon messages- This property was verified using the query, $\text{Pr} [\leq 20](\text{Engine.occ_Ctrl} \geq \text{Engine.occ_Serv})$
 - 2.5 No deadlock-The query $A[] \text{not deadlock}$ was used to check if the model never fell into a deadlock state.
3. Liveness- In this work, it is analyzed to check if service messages are eventually transmitted over the service channel even if it is initially blocked from transmission due to queue freezing.

Results and Discussion:-

UPPAAL 4.1.19 version was used for creating the timed automata of the protocol, validating and verifying it. Vehicle, infrastructure, enqueueing, dequeueing and the engine timed automata were modeled using UPPAAL. Through model simulation ambiguities, incompleteness in the model was identified and resolved.

The identified properties were verified using verifier. Verification of reachability property against the model succeeded showing that the messages can be sent to the control channel thereby indicating that the states are reachable and that there is a path in which safety messages would be sent. As a communication protocol includes a sender and a receiver it is essential that the states be reachable. Verification of liveness property also succeeded indicating that even if service message transmission to the channel is delayed, it will eventually get transmitted. Liveness property is verified so as to ensure that something good eventually occurs.

Verification of the safety properties was also done. Property 2.1 was checked for different intervals of time. For 1 unit of time, the acceptance rate was found to be between the interval [0.62, 0.72]. The Acceptance rate is found to

increase as the time interval increases. For 3 units of time and beyond, the probability of messages being accepted was higher. The acceptance rate was between the range [0.9026, 1]. Property 2.2 was checked by explicitly specifying the number of simulated runs. 2000 was the number of simulated runs specified. Verification of this property showed that up to 15 units of time the messages lost are nil. As there is an increase in time the message loss gradually increased. Verification of property 2.3 succeeded. The Success of the property showed that queue freezing was implemented in the model. Verification of property 2.4 succeeded. Verification of this property indicates that probability of safety messages to reach the control channel is greater than that of service messages. A graph showing the probability comparison of the property is shown in Figure 7. The probability value is shown in y-axis and time is indicated on the x-axis. When the probability value is greater than or equivalent to 1 it indicates that the faster is the probability for safety messages to reach the control channel than the service message to reach the service channel. Figure 6 shows the probability comparison graph. The probability value is greater than 1 for all units of time. When the value is greater than or equal to 1 it indicates that comparison greater is true. Verification of property 2.5 using model checker failed due to machine memory limitation. Therefore simulation via graphical interface was carried out to verify if the model fell into a deadlock state. A Deadlock state did not occur on simulation of the system. It is important to verify the safety properties to verify that nothing bad occurs in the protocol.

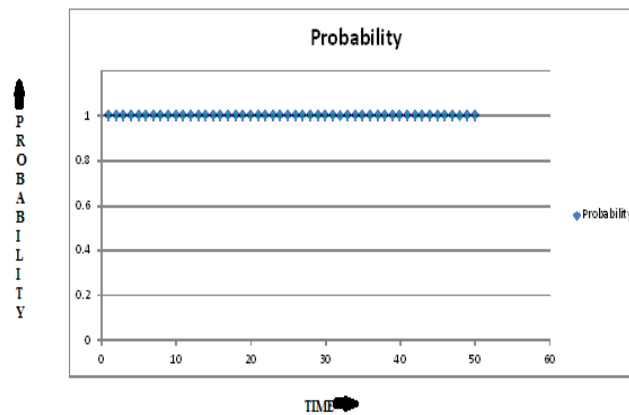


Fig 6:- Probability Comparison.

Conclusions:-

In this work an exhaustive analysis of congestion control protocol was carried out. This work presents a statistical model checking technique as an efficient and reliable way for analyzing the reliability and safety aspects of the congestion control protocol. The congestion control protocol was stochastically modeled using a network of timed automata namely the vehicle automata, infrastructure automata, enqueue automata, dequeue automata and transmission engine automata. The important properties that a congestion control must ensure were identified and expressed using query language. Both computation tree logic and probabilistic computation tree logic was used to express the properties identified.

UPPAAL SMC tool was used to model, validate and verify the protocol. Use of the SMC tool enabled stochastic representation of the protocol and the protocol could be analyzing both quantitatively and qualitatively.

As future work, the modeling of the protocol can be improved further. Transmission of messages from the channel to the vehicles within communication range has to be modeled and also the work here concentrated only on three types of messages, event driven, beacon and service messages. As future work more types of messages can be modeled and verified.

References:-

1. Xue Yang, Jie Liu, Feng Zhao, Feng Zhao, "A Vehicle-to-Vehicle Communication Protocol for Cooperative Collision Warning" First annual international conference on mobile and ubiquitous systems: networking and services, Aug 2004, pp.114-223.
2. Bouassida, M.S., and Shawky, M., A cooperative and Fully Distributed Congestion Control Approach within VANETs, 9th International Conference on Intelligent Transport Systems Telecommunication, 2009, pp. 526-531.
3. Mohamad Yusof Darus, Kamalrulnizam Abu Bakar. "Formal Verification of Congestion Control Algorithm in VANETs", I. J. Computer Network and Information Security, 2013, pp. 1-7.
4. 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.
5. Bouassida, M.S., and Shawky, M., "On The Congestion Control Within Vanet". 1st IFIP Wireless Days.(2008), 1-5.
6. Zang, Y.P., Stibor, L., Cheng, X., Reumerman, H. J., Paruzel A., & Barroso, A., Congestion Control in Wireless Networks for Vehicular Safety Applications".
7. Andrei, M. Calder, M. Fisher, S. Konur, "Quantitative Modelling and Automated Analysis of a Cooperative Congestion Control Protocol for Dynamic Vehicular Ad-hoc Networks"