Modeling ACC with Cloud, Clouldlet for Autonomous Vehicle Platoon using Petri nets

Tanmay Chakraborty[1], Shingo Yamaguchi[2], Mohd Anuaruddin Bin Ahmadon[2] and Soumya Kanti Datta[3]
[1]Student, Adamas University, India and Intern, Yamaguchi University, Japan.
[2]Graduate School of Sciences and Technology for Innovation, Yamaguchi University, Japan.
[3]Digiotouch OU, Estonia.
Emails: tanmay.chakraborty@ieee.org, shingo@yamaguchi-u.ac.jp,
anuar@yamaguchi-u.ac.jp, soumya@digiotouch.com.

***Abstract*—Adaptive Cruise Control (ACC) is an important component of any Advanced Driver Assistant System (ADAS) present in highly Autonomous Vehicle. In this paper, we have modeled and simulated a complex platooning system of such vehicles based on vehicular IoT, Cloudlets, and their ACC algorithm for determining the optimal Petri net that can be used for such modeling purposes. The architectures were modeled with timed stochastic Petri net as well as priority based Petri net for performing a comparative study. As our result, we could establish priority based Petri net as a better suited modeling tool for complex IoT system over timed stochastic Petri net.**

***Index Terms*—Adaptive Cruise Control; Cloud; Cloudlet; V2X; Petri net; Autonomous Vehicles; Platooning.**

I. Introduction

Connected Vehicles (CV) are vehicles that have communication capabilities with other vehicles or an infrastructure [1] and Autonomous Vehicles (AV) are vehicles with a number of systems that work together in harmony to achieve autonomy in driving. This requires on-board sensors LIDAR, camera, GPS/GNSS, accelerometer, gyroscope and more. They typically produce gigabytes of data every minute [1]. Connected Autonomous Vehicles (CAV) are the new generation of vehicles that use vehicle to everything (V2X) communications as well as on-board intelligence for a more efficient and safer operation. The on-board sensors in CAVs are being utilized as connected resources for an Internet of Things (IoT) ecosystem [2]. As a result, CAVs are able to provide services like shortest route to destination, cooperative platooning, road and traffic information. These vehicles are an example of complex IoT systems that needs more computation and real time solutions over conventional "Things".

Adaptive Cruise Control (ACC) is one of the key functionalities required for working of an autonomous vehicle. This system is responsible for controlling the vehicles speed, acceleration, retardation, braking etc. There have been a lot of research work laid down in this system to improve it's precision and robustness. In spite of having a number of literature a proper system modeling and process validation technique is missing. Another important fact about connected autonomous vehicles is that they generate huge amount of data. Considering the number of autonomous vehicles that will operate in the near future, telecommunication core networks will get congested if all the data needs to be transported to the Cloud servers. There will be huge latency in the Cloud server that must process the data and push decisions in real time for the car to perform an action. Thus, it is clear that present Cloud based servers are not enough to generate real time decision and there will be system failure. A highly desirable solution is to apply Multi-Access Edge Computing (MEC) based architectures. They are suitable for providing a consumer-centric IoT services in the CAVs [3]. A probable solution to the above mentioned network jamming is proposed in [4]. The authors proposed a novel IoT system based on Cloud and Cloudlet. They also established the advantage of using such an architecture over conventional architectures in cases where real time solutions are required with high computation power.

Petri nets or Place-Transition Nets are used for mathematically representing systems or processes [5]. They are useful tools that help in modeling different time and event based processes. The models have circles that represent places and rectangular boxes known as transitions. An arc runs from a place to a transition or a transition to a place. The Fig. 1 shows a basic Petri net example where $p0$ and $p3$ are places and $t1$ and $t2$ are transitions. A token in order to move from one place to another must pass through a transition. The arcs are weighted and thus Petri nets are also called weighted bipartite directed graphs, where the nodes are places and arcs are transitions. The weight of any arc is one by default if no weights are mentioned in a Petri net model. Models generated using Petri nets can be analyzed using linear-algebraic methods and they are an efficient tool for modeling large complex systems or processes. They are easily simulated and less expensive in terms of computation thus can be simulated in any low power computer. They help in debugging complex processes easily.

As the degree of complexity of IoT architectures are increasing [6], it is absolutely necessary to look into simpler and faster modelling tools for analysis of these systems and their

[1]https://www.networkworld.com/article/3147892/internet/one-autonomous-car-will-use-4000-gb-of-dataday.html

978-1-5386-4980-0/19/$31.00

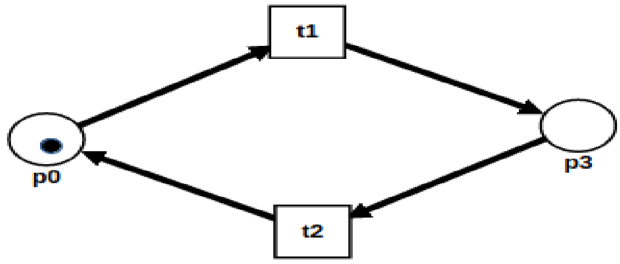

Fig. 1. An example of a Petri net.

architectures. Thus, we look into the applications of Petri net in modelling the complex IoT architectures. We use the complex architecture presented by [4] and [7] for our experiments. We successfully model their architecture with two types of Petri nets: timed stochastic Petri net and priority based Petri net. From our simulations, we could also establish priority based Petri net as a better suited modelling tool for modelling and simulating complex IoT architectures and systems.

The rest of the paper is arranged in the following way. Section II describes State-of-the-art. Section III describes the ACC algorithm and it's Petri net model. Section IV describes the Cloud, Cloudlet based model for autonomous vehicle platoon and it's simulation by Petri net. the paper is concluded in Section V.

II. State-of-the-art

Petri nets as a tool for modeling complex systems is a domain that is still under study. A lot of researchers have looked into its application for system validation.

Chandramohan [8] gives a basis on how ACC can be modeled using Petri nets. The author analyses the ACC with fault tolerant control mechanisms. But does not focus on the other systems like V2V, V2I, IoT and the connected vehicle platoon. Our works focus on the analysis of ACC with IoT, V2X and a vehicle platoon. The robust controller system designed by Filho et al. [9] after taking into consideration the braking capacity and system delays, guarantees that the minimum safety distance in between the vehicles is not violated in a platoon of vehicles. Li et al. [10] analyzed a Cooperative Adaptive Cruise Control (CACC) model based on modified car-following model on the basis of time to collision and linear stability. Their work portrays the immunity in CACC from rear end collision. Chang et al. [11] proposed a Cooperative Adaptive Driving system for a platoon. The length of the platoon was adaptively decided taking into consideration the traffic information from a local and global Cloud. Their CACC outperformed all previous systems. Hu et al. [12] proposed an ACC with proportional derivative controller that displayed an improved string stability and better performance taking into account the sensor delays, actuator lag and communication delay. Celaya et al. [13] describes a methodology consists of defining a simple multi-agent system based on the abstract architecture for intelligent agents. The abstract architecture is modeled as a discrete-event system using Petri nets and structural analysis of the net provides an assessment of the communication and coordination properties of the multi-agent system. Cistelha et al. [14] applied discrete event system concepts to model robotic tasks providing a systematic approach to modeling, analysis and design from specifications. Ding et al. [15] proposed a sorting algorithm applied to a traffic network model that was created with Petri nets to analyze its compositional structures, which represent different traffic lines, with the aim of optimizing the traffic network. Abellard et al. [16] modeled an adaptive-control for a wheel chair using data flow Petri nets. Their solution was an efficient control for wheel chairs and optimized the hardware/software co-design methodology. Further safety analysis of the systems modeled with Petri nets are also possible and it has been established that Petri nets perform better fault determination over fault trees [17]. Thus a desired method for applying in complex IoT systems. From this study we could establish that application of Petri nets for modeling and validating complex IoT processes has not yet been proposed.

III. Adaptive Cruise Control

ACC is the main control system of the autonomous vehicle. This system is responsible for the inertial control, braking control, keeping safe distance from obstacle, and providing a comfortable riding experience for the passengers. It's one of the key components for ADAS and self-driving cars. To ensure higher level of autonomy and safety for both vehicle and passenger, ACC system must be free from errors, fault-tolerant and very precise. A distributed system model for ACC is highly desirable for fault detection, system simulation and getting system response at different stages of algorithm before actually applying it in a real world scenario. In this section, the Petri net model of the ACC algorithm is proposed. We analyze the model validity by simulation of the Petri net.

A. ACC Algorithm Modeling

We have considered the ACC algorithm presented in [4] for our experiments. The ACC algorithm presented in Fig. 2 can be considered as a distributed system with a specific control flow. The algorithm has been modeled with a Petri net for analysis, simulation and verification of the system. The Petri net model is given in Fig. 3. The transitions of the model are given in Table I and the places are given in Table II. The model has been designed to simulate different functionality of the Petri net. The Petri nets are analyzed with their incidence matrix, reachability matrix and cut-set matrix. This gives a simple linear relation for mathematically validating systems. The model verified the ACC system process under experiment and provided a novel direction for verification of different complex CAV processes.

IV. Cloud and Cloudlet based Model for Autonomous Vehicle Platoon

The Cloud, Cloudlet and V2X architecture is one of the key technologies behind a more safer and better autonomous

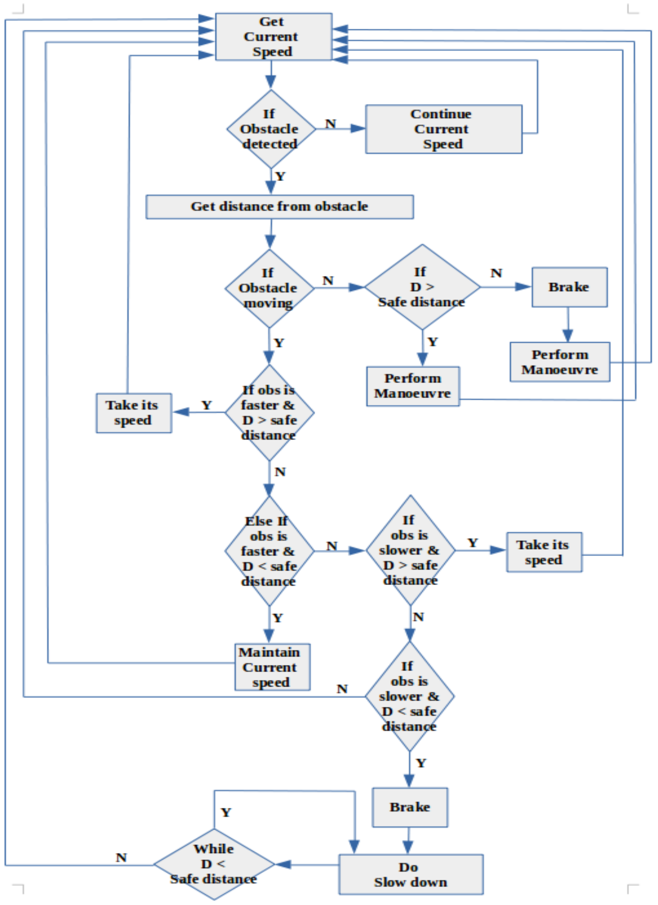

Fig. 2. ACC algorithm running on Cloudlet.

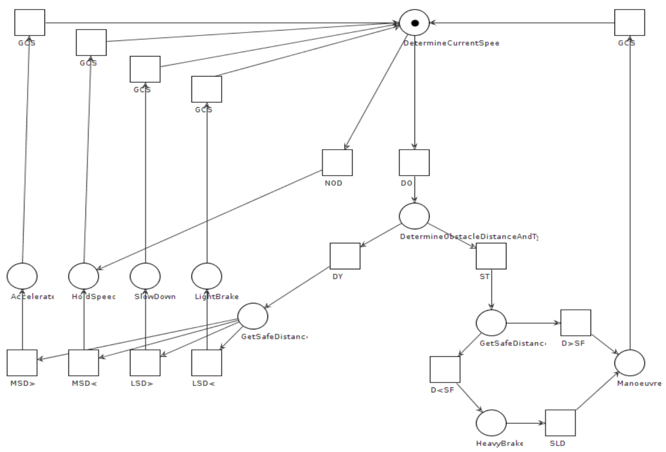

Fig. 3. ACC model using Petri net.

vehicle [18]. The IoT architecture in [4] highlights the integration of ACC in an on-board Cloudlet that is present in the autonomous vehicle. The Cloudlet is responsible for exchanging message with external infrastructure i.e Cloud and also receiving analytics from the external infrastructure which is displayed in a dashboard inside the vehicle. The Cloudlet offers cooperative localization and allows platoon

TABLE I
TRANSITION LABELS FOR PETRI NET IN FIG 3

Transition	Label
DO	Detect obstacle
DY	Dynamic obstacle
ST	Static obstacle
NOD	No obstacle detected
$D < SF$	Distance- vehicle & obstacle less than safe distance
$D > SF$	Distance- vehicle & obstacle more than safe distance
SLD	Slowing down
$LSD <$	Obstacle less speed & distance less safe distance
$LSD >$	Obstacle less speed & distance more safe distance
$MSD <$	Obstacle more speed & distance less safe distance
$MSD >$	Obstacle more speed & distance more safe distance
GCS	Get current ACC speed

TABLE II
PLACE LABELS FOR PETRI NET IN FIG 3

Place	Label
DetermineCurrentSpeed	Determines present ACC speed
DetermineObstacleDistanceAndType	Distance from obstacle & it's type
GetSafeDistance	Calculates braking distance
Manoeuvre	Pass a car or change lane
HeavyBrake	Near stop
LightBrake	Instant speed control and retardation
Slowdown	Reduce speed from current speed
Holdspeed	Maintain current speed
Accelerate	Increase speed from current speed

formation that reduces the chances of accidents and helps in having a safer ride experience for the passenger. The V2V communication which allows the sharing of the AV data via short range communication protocols is also managed by the on-board Cloudlet. In this section, we model their complex IoT system with Petri nets and validate their architecture.

A. ACC in Cloudlet

The IoT architecture of which Cloudlet is an integral part is given in Fig. 4. The Fig. 5 shows the functional components of the Cloudlet. The Cloudlet hosts the required software components for multiple communication technologies and communication protocols required for V2X communication. The architecture is carefully designed with different infrastructure blocks for Road Side Unit's (RSU), positioning, and a secure Cloud server. The central Cloud provides the Cloudlet with analytics like current platoon position, average platoon speed, ETA to destination, Shortest route to destination, and road information.

B. Petri net model of Cloud, Cloudlet & Autonomous Vehicle Platoon

The Cloud, Cloudlet based complex IoT architecture for Connected Autonomous Vehicle platoon system is modeled using Petri nets for analysis and validation of the IoT system. The system block diagram used for Petri net modeling is given in Fig. 4. The process is modeled using two different kinds of Petri net for a better analysis. Given in Fig. 6 is

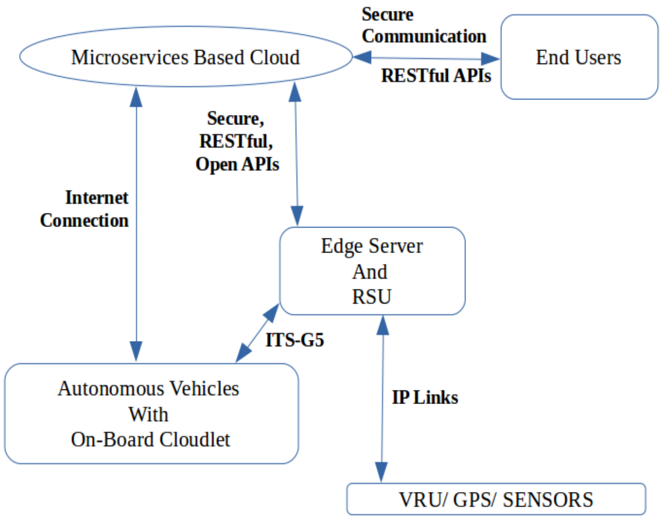

Fig. 4. IoT Architecture including Cloudlet and AV.

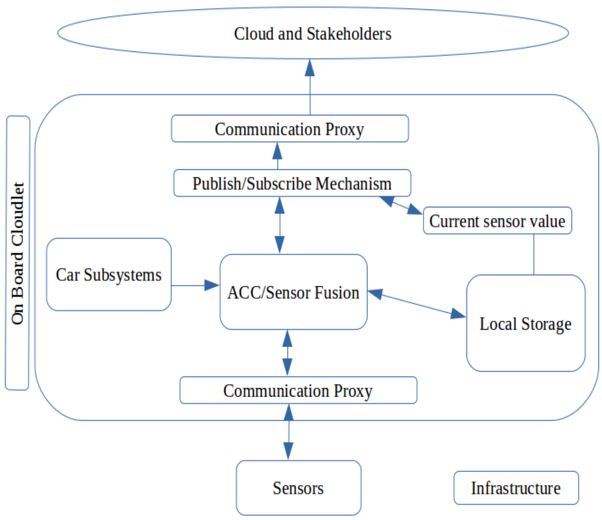

Fig. 5. ACC integration with Cloudlet.

the model using priority based Petri net and given in Fig. 7 is the stochastic timed Petri net model. The transition for both the models is given in Table III. The place for both the models is given in Table IV. The model represents a platoon with number of Cloudlets capable of exchanging data via V2V communication by publish/subscribe mechanism. Here, data is considered as tokens and each vehicle behind the lead vehicle has been subscribed to the data of it's previous vehicle. Each vehicle in the platoon is also allowed to send data to the Cloud via a V2I communication model and receive data from the Cloud in the on board dashboard. We have implemented the concept of new data generation via a transition called Sensors&DataGenerators. The process in the model is simulated using PIPE 5[2].

[2]http://www.doc.ic.ac.uk/~wjk/publications/bonet-llado-knottenbelt-puijaner-clei-2007.pdf

The timed stochastic Petri net model uses timed transitions which fire at a specified rate. This model needs the rate to be specifically calculated and it becomes complex to calculate the rate when the number of vehicle agents increase. In this model, the transitions fire in a random way thus making it difficult to maintain any specific order of the system and it is important for an IoT architecture to maintain its process firing order. While in the priority based Petri net model the transitions fire based on their priority in the system. In this type of model adding new vehicle agents become easier and also making the system fire transitions in a particular order can be achieved with different levels of priority values.

C. Simulation and Model Analysis

Simulation of the two models were performed with a timed transition firing method. The time lag in a on-board Cloudlet is near about 120ms and the time lag in DSRC communication is near about 75ms [4]. So, while simulating the system we took these time lags into consideration and a 195ms time lag was used between each transition firing. This method of simulation allowed us to get closer to the actual system. On vigorous simulation of the two given models we observed the priority based Petri net model outperform the timed stochastic Petri net model for autonomous vehicles process modeling. Maintaining the proper sequence of transition firing is an important aspect in IoT process modeling. It makes debugging easier as well as makes the system more stable for real world application. The priority based model provided the desired sequence of transition firing and the sequence of firing did not alter with increasing iteration while the timed stochastic model did not maintaining the proper sequence of transition firing.

The graph in Fig. 8 shows the data accumulation in the Cloudlets with increasing number of transition firings for timed stochastic model. Data accumulating in the Cloudlets will eventually decrease the performance of the system and increase time lag as now new data cannot be processed due to the previous data being present in the Cloudlets. The graph in Fig. 9 shows the data accumulation in the Cloudlets for priority based Petri net model. Comparing the two graphs we can infer that priority based model is capable of modeling real time IoT systems over the timed stochastic model. The incidence matrix and the minimum cut set matrix can be generated after this modeling and analysed if the system is failing at any point.

Thus the priority based Petri net is being proposed as a tool for complex IoT system modeling like the connected autonomous vehicles platoon. The priority values of the system are chosen empirically by keeping in mind the desired sequence the system should work in. As the priority values are not mentioned in the architecture chosen for modeling. The ratio of priorities between each transition of V2V&V2I is being kept constant but the value of individual priorities were reduced with a constant value without disturbing the ratio. The priority of the sensor data generator is kept low and the analytics transition priority is kept high. So the desired sequence is maintained. The priority value of the model is given in table V for Fig. 6. For a system under test the architecture should define the priority of their processes in order to model it using this type of net. This modeling will help the architecture to validate if the priority is working like it should. With this model, finding the optimal priority value of each process in a architecture is also possible by vigorous simulations.

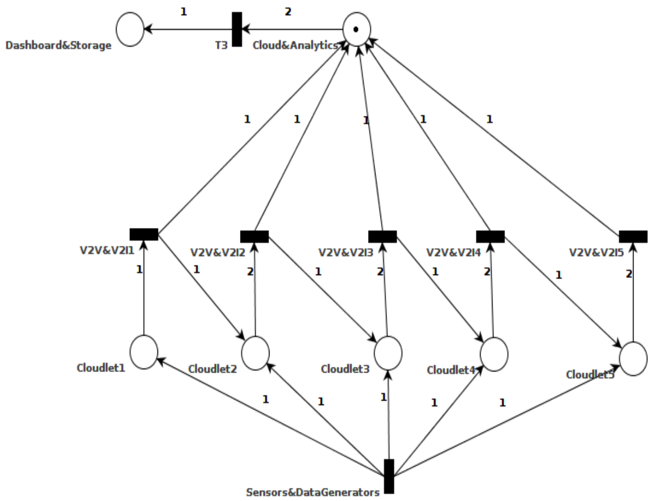

Fig. 6. Model of Autonomous Vehicle Platoon with Cloud and Cloudlet using priority based Petri net.

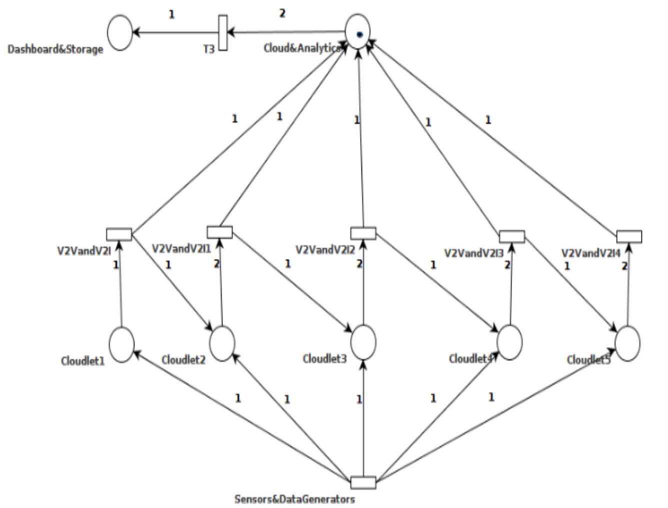

Fig. 7. Model of Autonomous Vehicle Platoon with Cloud and Cloudlet using timed stochastic Petri net.

TABLE III
TRANSITION LABELS FOR PETRI NET FIG. 7 AND FIG. 6

Transition	Label
SensorDataGenerators	Data Generator/Sensors
V2V&V2I	Communication module
T3	Data analytics

TABLE IV
PLACE LABELS FOR PETRI NET FIG. 7 AND FIG. 6

Place	Labels
Cloudlet1	Lead vehicle of platoon
Cloudlet2 - Cloudlet5	Following vehicles
Cloud	Central Cloud
Dashboard&Storage	Cloud dashboard for visualization of data & storage

TABLE V
TRANSITION PRIORITY FOR PETRI NET IN FIG 6

Transition	Priority
SensorDataGenerators	10%
V2V&V2I1	90%
V2V&V2I2	85%
V2V&V2I3	80%
V2V&V2I4	75%
V2V&V2I5	70%
T3	100%

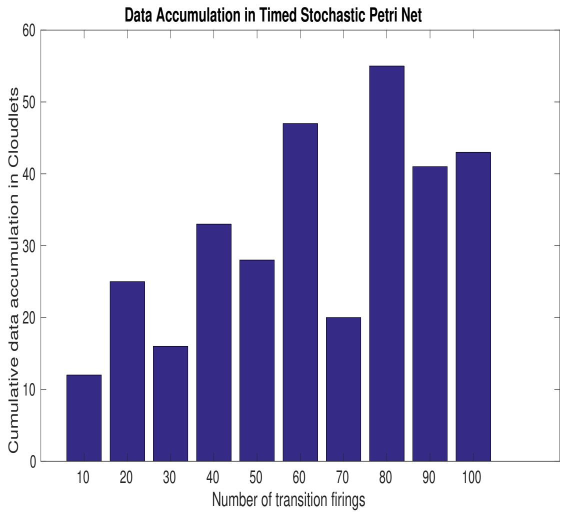

Fig. 8. Graph for cumulative data token accumulation in the Cloudlets for timed stochastic Petri net model.

D. Advantage, Novelty, and Applications

Most IoT systems are categorized as distributed systems. This work proposes the application of Petri nets in complex IoT system modeling and validation. Petri nets being a powerful graphical modeling tool is capable of modeling complex IoT systems with ease and allows a linear matrix method for fault analysis. This linear, simple and graphical modeling tool is advantageous to use when the IoT system is complex and fault determination is cumbersome. This type of complex IoT architecture modeling using priority based Petri nets have not been proposed earlier. This opens a new direction of study towards validating IoT architectures with simple graphical simulation. We can see that there are a lot of IoT architectures in the literature but not a very suitable method for validating them without implementation. With the presented tool one can check the architecture before adapting in their work.

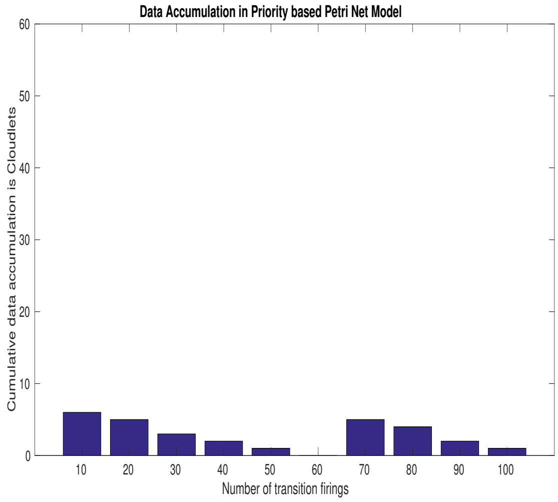

Fig. 9. Graph for cumulative data token accumulation in the Cloudlets for priority based Petri net model.

V. Conclusion

The paper highlights application of Petri nets in IoT system modeling. We utilized a complex connected autonomous vehicle platoon system for our experiments to validate our approach. We could establish the advantage in using Petri nets for IoT system modeling. Our simulations also could establish priority based Petri nets as a better tool for modeling complex IoT systems where the sequence of processes plays an important role.

References

[1] S. Kanti Datta, M. Irfan Khan, L. Codeca, B. Denis, J. Hrri, and C. Bonnet, "Iot and microservices based testbed for connected car services," in *2018 IEEE 19th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, pp. 14–19, June 2018.

[2] S. K. Datta, J. Haerri, C. Bonnet, and R. Ferreira Da Costa, "Vehicles as connected resources: Opportunities and challenges for the future," *IEEE Vehicular Technology Magazine*, vol. 12, pp. 26–35, June 2017.

[3] S. K. Datta, C. Bonnet, and J. Haerri, "Fog Computing architecture to enable consumer centric Internet of Things services," in *2015 International Symposium on Consumer Electronics (ISCE)*, pp. 1–2, June 2015.

[4] T. Chakraborty, S. Yamaguchi, and S. K. Datta, "Sensor fusion and adaptive cruise control for self driving platoon," in *2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)*, pp. 231–235, Oct 2018.

[5] J. L. Peterson, "Petri nets," *ACM Comput. Surv.*, vol. 9, pp. 223–252, Sept. 1977.

[6] P. Ray, "A survey on internet of things architectures," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, no. 3, pp. 291 – 319, 2018.

[7] T. Chakraborty and S. K. Datta, "Siiot: A shortest path estimation and obstacle avoidance system for autonomous cars," in *2018 Global Internet of Things Summit (GIoTS)*, pp. 1–6, IEEE, 2018.

[8] N. A. Chandramohan, *Design and modeling of adaptive cruise control system using petri nets with fault tolerance capabilities*. PhD dissertation, Purdue University, 2018.

[9] C. M. Filho, M. H. Terra, and D. F. Wolf, "Safe optimization of highway traffic with robust model predictive control-based cooperative adaptive cruise control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 3193–3203, Nov 2017.

[10] Y. Li, H. Wang, W. Wang, L. Xing, S. Liu, and X. Wei, "Evaluation of the impacts of cooperative adaptive cruise control on reducing rear-end collision risks on freeways," *Accident Analysis and Prevention*, vol. 98, pp. 87–95, 2017.

[11] B. J. Chang, Y. L. Tsai, and Y. H. Liang, "Platoon-based cooperative adaptive cruise control for achieving active safe driving through mobile vehicular cloud computing," *Wireless Personal Communications*, vol. 97, pp. 5455–5481, Dec 2017.

[12] S. G. Hu, H. Y. Wen, L. Xu, and H. Fu, "Stability of platoon of adaptive cruise control vehicles with time delay," *Transportation Letters*, vol. 0, no. 0, pp. 1–10, 2017.

[13] J. R. Celaya, A. A. Desrochers, and R. J. Graves, "Modeling and analysis of multi-agent systems using petri nets," in *2007 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1439–1444, Oct 2007.

[14] H. Costelha and P. Lima, "Modelling, analysis and execution of robotic tasks using petri nets," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1449–1454, Oct 2007.

[15] J. Ding, X. Chen, and R. Wang, "A compositional analysis method for petri-net models," *IEEE Access*, vol. 5, pp. 27599–27610, 2017.

[16] A. Abellard, M. M. B. Khelifa, and M. Bouchouicha, "A petri net modelling of an adaptive learning control applied to an electric wheelchair," in *2005 International Symposium on Computational Intelligence in Robotics and Automation*, pp. 397–402, June 2005.

[17] T. Liu and S. Chiou, "The application of petri nets to failure analysis," *Reliability Engineering System Safety*, vol. 57, no. 2, pp. 129 – 142, 1997.

[18] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Communications Magazine*, vol. 56, pp. 48–54, August 2018.