

A Time-Scaling Data Generation Method For Internet of Things Simulation

Zhuo Hu, Yuhong Li, Rahim Rahmani, Shengnan Han

Abstract—Simulation plays an important role on studying the Internet of Things (IoT) traffic, which has an increasing impact on network infrastructures. The existing simulation tools and mechanisms on IoT mainly focus on simulating large-scale deployment of IoT and heterogeneous IoT applications. This paper concentrates on how to simulate the lengthy, burthty and multi-level time-scale IoT applications quickly. A time-scale data generation (TSDG) method is proposed to reduce the used simulation time of different IoT scenarios while keeping the real world characteristics of the IoT applications. Implementations of TSDG in ns-3 and simulation experiments of smart home and smart shopping centre are described in the paper. The evaluations results show that the proposed TSDG method can effectively reduce the time used to simulation IoT applications in ns-3 while reflect the IoT application traffic in the real world.

Keywords—IoT, Simulation, Time-scaling, Smart City

1. Introduction

With the continuous development of the Internet, an unprecedented number of Internet of Things (IoT) devices are gradually being added to the Internet [1]. The connected devices form a huge physical network, providing Smart City, Smart Health and Smart Industry etc. services by supplying and using information. With the increasing popularity of IoT devices and technologies, the IoT data traffic has a negligible impact on network infrastructure. Therefore, understanding the IoT data traffic is becoming critical [2].

Some research work have revealed the traffic patterns generated by single IoT devices and applications. For example, work [3] showed that the data intervals of payment and retail system follows exponential and Pareto distribution

Zhuo Hu, Yuhong Li
State Key Laboratory of Networking and Switching Technology, Beijing
University of Posts and Telecommunications
China

Zhuo Hu, Yuhong Li
Key Laboratory of Embedded System and Service Computing, Ministry of
Education, Tongji University
China

Rahim Rahmani, Shengnan Han,
Department of Computer and System Sciences, Stockholm University
Sweden

respectively for smart shopping center; the humidity and temperature detector and electric meter etc. devices in the smart home generate data periodically [4], but with different periods [5]. However, the large number of heterogeneous IoT devices and applications, complex application scenarios and IoT network topology made the integrated IoT data traffic much more complicated than the single type of IoT device data, which has not been well studied. The influence of IoT data traffic on the network infrastructure is a hot topic for researchers and research organizations.

Simulation has been proven to be an efficient method for studying and predicting data traffic [6][7], and some simulations tools have been developed, such as ns-3[8], OPNET[9]. Different networking environment, traffic patterns and mobility models etc. have been developed, which can be used to generate data traffic of various kinds of networking applications and scenarios and to evaluate the metrics of performance. Especially, various simulation tools specialized for IoT have been developed, such as[10][11][12][13]. Compared with the well-known general purpose network simulation tools, these tools mainly concentrating on solving the problems of massive number of IoT devices and large scale of IoT network. However, the characteristics of the IoT traffic have not been well considered.

In general, the IoT data traffic consists of four types of data in the machine-to-machine (M2M) commination [14] [15], namely heart beat, event trigger, payload data and node update. The heart beat is normally used by the devices in order to keep themselves connected with the infrastructure [16]. Thus, this type of data has a certain period, and has a constant size. However, depending on the sensors and type of applications, the period can be very different. Event trigger is initiated by a device (e.g., the device as a server) to trigger an action of an M2M device, for example, to report the humidity of a room to the network. The trigger process is irregularly generated. Although the data of the trigger itself is simple, the following event data might be a burst, which depends heavily on the applications. Payload data are those data used for exchanging information between devices. The data traffic is variable heavily and can either be of constant size like in telemetry, or of variable size like images in video surveillance. Also, the data can be either sent regularly but with different period or irregularly. Node update are programs or configuration parameters used to maintain the normal operation of devices, for instance, a server may push configurations or firmware to devices, and the devices will run the update from the servers dynamically or after reboot.

Generally speaking, the IoT traffic varies heavily with the application scenarios and is independent of human behaviour. In particular, the time of the information exchange is no longer

defined by humans. This puts forward challenges to the simulation tools. It might take a long time to simulate a IoT scenario due to the irregular period of IoT devices [17][18][19]. Obviously, the data sent by devices with long period need to be considered.

In this paper, a new method is proposed on how to simulate the lengthy, bursty and multi-level time-scale IoT applications quickly. A time-scale data generation (TSDG) method is proposed to reduce the time used for simulating different IoT scenarios while keeping the real time of the related IoT applications and the detail data traffic of the applications.

The rest of the paper is organized as follows. In section 2, we will discuss the existing work about simulations for IoT. In Section 3, we will introduce our simulation method TSDG, elaborating how it can reduce the simulation time while keeping the details of data traffic. Following this in Section 4, we will describe the implementation of our method in the network simulator ns-3. We will illustrate the test results by simulating the smart home and smart shopping centre scenarios in Section V and finally make conclusions in Section VI

II. Related Work

The large scale of network, numerous applications and heterogeneous devices put forward challenges to the use of existing simulation tools. To meet this challenge, Stefano Ferretti et al. [20] proposed a multi-level simulation method for modeling large IoT environments. They use "advanced" emulators to provide coarse-grained detail. A simulator coordinates the execution of a set of domain-specific "low-level" simulators that are generated and executed where the fine-grained details are needed. Two levels of simulators interact and synchronize their execution to calculate the correct status update.

In [21], the authors identified requirements for the next generation IoT experimental facilities, discussed some shortcomings of the simulation-based approach. They investigated the existing test platforms, which support also simulation. They claimed that a joint test platform for simulation was possible. However, such a simulation tool cannot satisfy the requirements of IoT simulations. The network scale and the level of details required by IoT applications and scenarios cannot be easily simulated.

In work [22], the technique of "hybrid simulation" is proposed in order to enhance the scalability and allow real-time execution of large-scale IoT environments (for example, large smart cities). "Hybrid simulation" means a simulation method that glues multiple simulation models together [23]. Each simulator has a specific task orchestrated by several coordinators. When these simulators work at different levels of detail, it is referred to multi-level simulation. This solution allows the creation of multiple interactive instances of different simulations specifically designed to particular aspects of the reduced portion of the simulated region, or to simulate a simplified subset of entities.

The authors in [24] studied the scalability of IoT applications of Wireless Sensors and Actuator Networks (WSAN) through simulations. The Ptolemy II simulation tool and the real MQTT message broker are used. The authors presented a case study involving the deployment of sensors and actuator networks as part of an IoT application for monitoring and controlling urban parks. Their experimental results show that the scalability of this type of application can be analyzed using simulations. However, they did not consider simulations of other types of IoT traffic.

The authors in [25] proposed a cost-effective software online simulation method that allows testing software to be deployed on simulation devices and environments. To maximize code reuse, the authors' effort was to create adapters to integrate the deployment code with the underlying platform, including the generic simulation engine, providing more specific but reusable packages, for example, simulation mobility, communication, logs. The main advantage of this method is that the actual code have already run once before being deployed. The simulation engine replicated it in the virtual device and environment.

Simulation was used to provide a means to study the real IoT-fog cloud systems in [15]. By analyzing the sensor data formats and data sets of smart cities, a tool called SUMMON was proposed to provide a means of collecting and filtering open data sets. However, this work focused on only how to provide IoT data to the simulation environment and illustrated its use in the open source IoT device simulator.

In general, the current work on IoT simulations are mainly trying to solve the problems of scalability and various IoT application models. Methods with different levels of granularity are used to model IoT network, environment and applications at different levels of detail. Parallel and multi-level simulation are introduced to simulate the large-scale IoT scenarios. To a certain degree, a kind of space scalability has been achieved during simulations. However, there is few work dealing with the time scalability during the simulations, i.e., to reduce the time used for simulating IoT applications and scenarios generating data with long and irregular periods. This is the focus of our work in the paper.

III. A Time-Scaling Data Generation Method(TSDG)

The purpose of the time-scaling data generation (TSDG) is to reduce the simulation time whereas keep the original time relationship of the data packets of one or multiple applications, in other words, to keep the traffic model or the original applications. Therefore, the simulation efficiency for various applications in a known scenario can be improved. It allows us to finish collecting data traffic in a shorter time than that happens in the real world, where heterogeneous IoT devices send data simultaneously in the simulated IoT scenario.

The basic idea of TSDG is to add a timestamp to the transmitted data to record the sending time of the data. At the same time, a traffic processing procedure is added to re-assemble and synthesize the data packets according to the

recorded sending time and transmission time in the network links. In order to reduce the simulation time, four techniques are used in TSDG, namely uninterrupted operation, continuous packet sending, parallel startup and process splitting.

A. Uninterrupted operation

As mentioned above, generally IoT traffic consists of four types, i.e., heart beat, event trigger, payload data and node update. No matter when and which type of traffic a device is transmitted, typically the traffic is intermittent, in other words, there will be an idle interval, which may be much longer than the actual traffic transmitting time. In this case, the idle interval for all the devices can be removed during the simulation. Fig. 1 illustrates the traffic sent by a single device. traffic_1 to traffic_4 are four periods in which the device has data to send. During the simulation, the idle intervals are discarded. After the traffic processing at the receiver nodes, the idle interval can be added and the traffic model will be recovered.

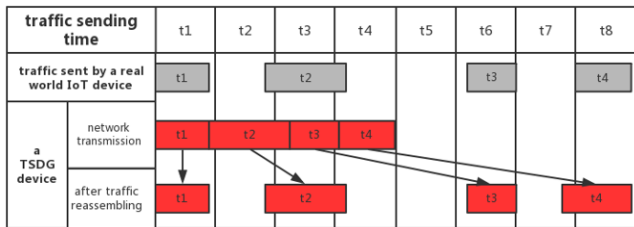


Figure 1. Uninterrupted operation for the data sent by a single device

Fig.2 illustrates the format of the data packet sent by the temperature sensor in our implementation. A new timestamp field “sending time” with the length of 26 bytes is added to each data packet. At the receiver node, the traffic will be re-assembled according to the timestamp and the transmission time.

```
"id": 5,
"temperature": 28.9273120410964,
"altitude": 5.35,
"latitude": 39.99139,
"longitude": 116.31154,
"sending time": "2018-07-
```

Figure 2. Timestamp in the data packet sent by the temperature sensor

B. Continuous packet sending

In the real world, some devices send data according to a certain statistic rule or distribution. For example, the data interval of payment follows exponential distribution and the retail system follows Pareto distribution in a smart shopping centre [3]. In order to reduce the time used for simulation, the data packet at the sender node will be generated and sent continuously, ignoring the time interval. A traffic generating timer is used, which generates and records the time offsets for the timestamps according to the known statistic distribution of the data packets.

As shown in Fig.3, after the continuous packet sending, the time used for generating and sending traffic packets using

TSDG may be reduced. Furthermore, the reduced time depends on the statistic distribution of the real world traffic model of the device.

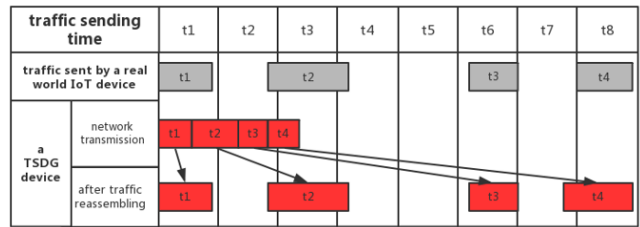


Figure 3. Continuous packet sending for the traffic of a single device

C. Pseudo parallel startup

In the complex IoT scenarios, the start and run time of individual IoT devices are different, as shown in Fig. 4. In this example, seven devices (i.e., s1-s7) start to transmit data at different time. Among them, the device s6 began to send data at latest. Although the time used for sending data is the shortest, the whole process cannot end until s6 finishes sending data. Obviously, if s6 can start sending data earlier, the time used for simulating this scenario will not be determined by the data sending time of device s6.

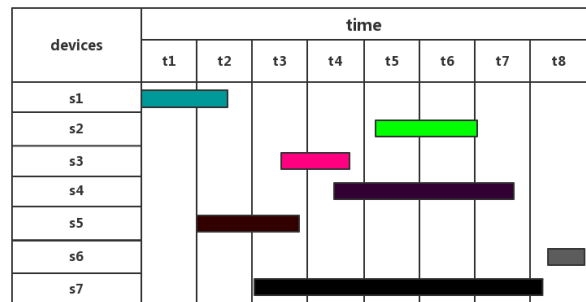


Figure 4. Data transmission of devices in the real world

As shown in Fig. 5, in TSDG, each device begins to run at the “beginning” of the entire simulation scenario according to a scheduler, which is related with the timestamp manager too. The idle time without data traffic generation of each device is discarded, and the actual start time are notified to the receiving device by inserting the timestamp in the generated data packets.

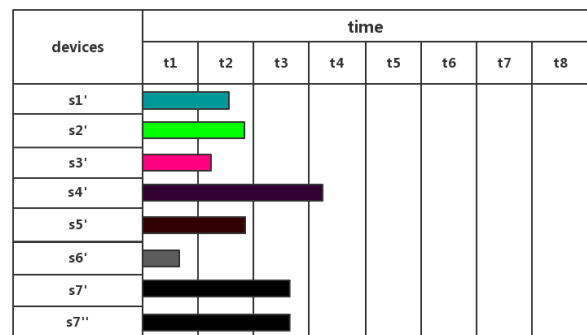


Figure 5. Parallel startup and data transmission process splitting in TSDG

D. Transmission process splitting

Based on the pseudo parallel startup, a variety of devices start simulation at the almost the same time under the control of the scheduler. According to the anti-barrel principle, the overall simulation time is determined by the IoT device with the longest simulation time. Thus, minimizing the simulation time of the device with the longest running time is the key to reduce the whole simulation time.

In the example shown in Fig. 4, device s7 has the longest data transmission time. Therefore, we split the data transmission of device s7 into two data transmissions. As shown in Fig. 5, s7' is responsible for the first half of the data transmission process, and s7'' is responsible for the second half of the process.

The difficulty of this technique is how to split the transmission process into several processes, for example, how to make s7'' operate exactly the second half of the s7. First, we make sure that the data sent by the device s7' and s7'' have the same flow identifier, namely, the source address, destination address, source port and destination port. Second, the timestamps of the packets sent by the virtual device s7'' are calculated and adjusted according to the timestamp of s7' in order to guarantee the whole time sequence of s7.

In our current implementation, we split processes with relative long periodical cycles and with symmetric traffic models, such as normal distribution. In addition, a scheduler is used to schedule the data sending of all the devices, including virtual devices.

In addition, since the simulation is realized in one local machine, the synchronization between the sender nodes and receiver nodes do not needed to be considered. According to the topology of the IoT networks, the transmission time can be achieved. Hence, at a certain receiver node (e.g., the gateway node in Fig. 7), the traffic can be re-assembled according to the timestamps in the data packets.

iv. Implementation of TSDG based on ns-3

We have implemented the TSDG method on the network simulator ns-3. The TSDG mainly consists of 6 functional modules, namely timestamp manager, scheduler, transmission process splitting, idle period removing, traffic re-assembling and traffic synthesizing. Among them, the timestamp manager stores the time offsets and timestamps of each flows (i.e., data packets sent by each device or virtual device) and is used for inserting the timestamps during the packet sending. Scheduler is used to schedule the start time of sending packets for each device.

Fig. 6 (a) and (b) illustrate the processing at the sender and receiver side respectively. At the sender node (Fig. 6 (a)), TSDG first splits the process of sending data, calculates the corresponding time offsets according to the traffic model of the node (i.e., device), and stores it in the timestamp manager. Then the idle periods will be checked and removed according to the traffic model. Similarly, all the time offsets will be

recorded in timestamp manager. Then, depending on the traffic model, the data transmission rate will be calculated to realize the continuous packet sending. After these procedures, the ns-3 node, and the corresponding link, protocol etc. will be configured and packets will be sent under the control of the scheduler.

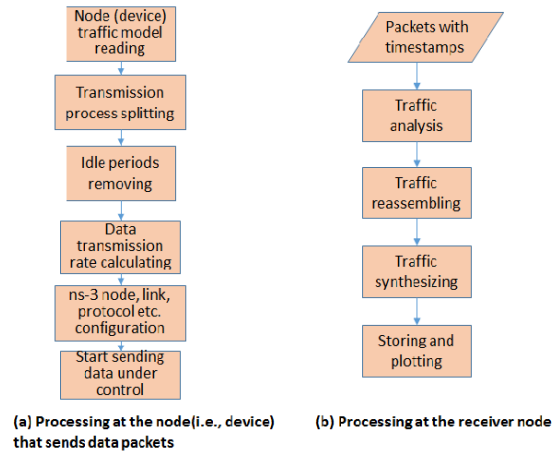


Figure 6. Processing at the sender and receiver node

At the receiver (Fig. 6 (b)), the node may receive data from different devices simultaneously (e.g., at Sink a, Sink b and the Gateway in Fig. 7). Through the traffic analysis, packets and flows from different senders are identified and classified. Then the data from the same device will be re-assembled according to the timestamps. Finally, the traffic from different devices are synthesized and stored in the database and plotted depending on the needs.

v. Test Results and Analysis

In order to evaluate the TSDG method, we implemented two IoT scenarios based on the implemented TSDG. Fig.7 illustrates the network topology we used. The integrated traffic at the sinks of the sub-networks of smart homes and smart shopping centres are simulated.

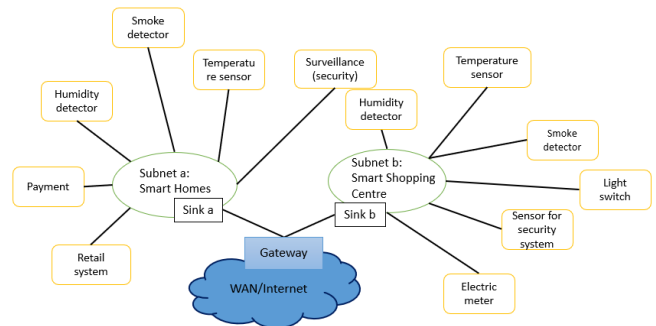


Figure 7. Network topology

For the Subnet a, the traffic patterns of the devices in each household [5] is shown in Table I. Altogether data from 100000 households will be gathered at Sink a. In this example, all the devices sending data periodically but with different periods, from 6 seconds to 18 seconds.

TABLE I. DEVICES AND TRAFFIC PATTERN IN EACH HOUSEHOLD

Device type	Device number	Packet size (byte)	Period (s)
Humidity detector	5	90	18
Temperature sensor	4	80	6
Smoke detector	2	80	6
Light switch	5	100	18
Sensor for security system	1	80	6
Electric meter	1	90	6

Table II shows the devices and the corresponding traffic model of each shopping centre [3]. Here the data intervals of payment and retail system follows exponential and Pareto distribution respectively. Altogether data traffic of 8 shopping centres are generated and will be collected at Sink b.

TABLE II. DEVICES AND TRAFFIC PATTERN IN EACH SHOPPING CENTRE

Device type	Device number	Packet size (byte)/distr	Period(s)/Interval
Humidity detector	40	90	18
Temperature sensor	40	80	6
Smoke detector	60	80	6
Surveillance (security)	70	Log normal distr.(5.9,1.2)	15
Payment	50	205	Exp. distr. ($\mu=2.5$ sec.)
Retail system	50	72	Pareto distr.(1,10)

Fig. 8 and Fig. 9 illustrates the data collected at Sink a and Sink b respectively for the period of 6000 seconds without using TSDG.

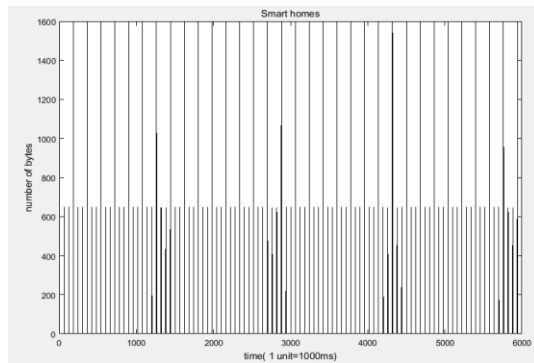


Figure 8. Data traffic at Sink a of smart homes

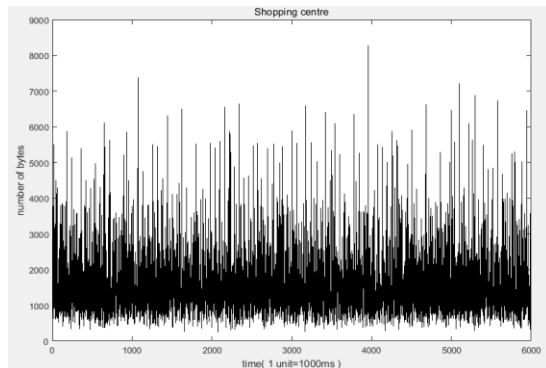


Figure 9. Data traffic at Sink b of smart shopping centres

We extended the simulation time with and without using TSDG. Fig. 10 shows the time of the real world application running time (i.e., the simulation time without using TSDG) and the simulation time (i.e., using TSDG). Here we can see that for the smart home scenario, the simulation time has been reduced greatly. This is because all the devices belonging to one household send data periodically, whose simulation time can be well compressed. However, since there are 100000 households, and the devices belonging to different households begin to send data randomly with 18 seconds (i.e., the longest period of the devices) certain running time are needed. On the contrary, the simulation time for the smart shopping centres have not been reduced very much. This is because the traffic of both the payment and the retail system do not follow symmetrical distribution, and the data sending process have not been split for the moment. In addition, different devices began to send data randomly within 18 seconds, therefore the simulation time have not been reduced greatly.

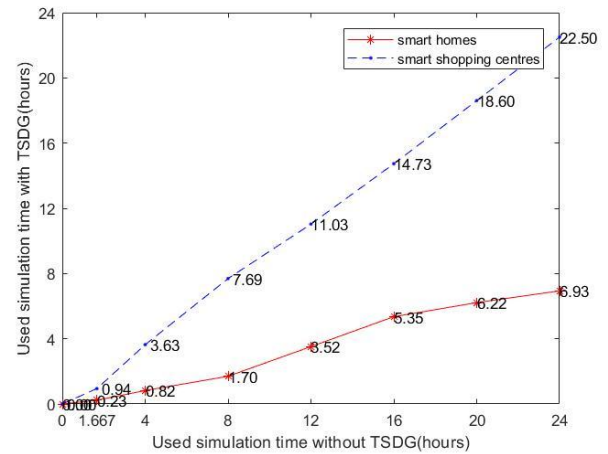


Figure 10. Simulation time using TSDG

To further analyze the effect of different type of devices, we measured the simulation time of each type of devices in the shopping centre scenario, as shown in Table 3. We set the running time of the devices to different values, and measure the used simulation time of each device. Here we can see that for the periodical traffic, the time reduction is roughly dependent on the period of the traffic model. The longer the period, the greater the reduction. However, from Table III, we also found that if the traffic model follows asymmetric distribution, the time reduction is not much. We are now work on reducing the simulation time in this case.

TABLE III. SIMULATION TIME OF DIFFERENT TYPES OF DEVICES IN THE SCENARIO OF SHOPPING CENTRE

Device type	Running Time (seconds)			
	60	300	1800	3600
Humidity detector	3.35	16.67	98.81	204.29
Temperature sensor	9.55	48.73	321.46	597.20
Smoke detector	9.89	50.42	310.02	584.11
Surveillance (security)	4.37	23.66	128.08	242.71
Payment	58.82	284.19	1577.34	3215.61
Retail system	58.53	286.42	1582.05	3300.22

vi. Conclusions

In this paper, we analyzed the bottleneck of the simulations for IoT applications and scenarios, and proposed a method to reduce the simulation time. Uninterrupt operation, continuous packet sending, pseudo parallel startup and transmission process splitting techniques are used in the method to reduce the simulation time. We have implemented the proposed method in the network simulator ns-3 and perform simulations using it for the IoT scenarios of smart household and smart shopping centres. Simulation results show that the proposed method can reduce the simulation time greatly. Our next step is to improve the method so that more time reduction can be achieved under various traffic models of IoT devices.

References

- [1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash. Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4):2347–2376, Fourthquarter 2015.
- [2] Gabriele D'Angelo, Stefano Ferretti, Vittorio Ghini. Modeling the Internet of Things: a simulation perspective. *IEEE 2017 International Conference on High Performance Computing and Simulation (HPCS 2017)*, September 2017.
- [3] Osoba S B, "Appraisal of parking problems and traffic management measures in central business district in Lagos,". *Journal of Sustainable Development*. 2012, 5(8): Pages 105-115.
- [4] Jeu Young Kim, Hark-Jin Lee, Ji-Yeon Son. Smart home web of objects-based IoT management model and methods for home data mining. 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS).
- [5] A. Orrevad, "M2M Traffic Characteristics-When machines participate in communication," Master of Science Thesis, Royal Institute of Technology, Sweden, 01/12-2009.
- [6] Hongjing Zhang, Ying Wang, Xuesong Qiu. Network operation simulation platform for network virtualization environment. 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS). Pages 19-21, August 2015.
- [7] Jinbao Li, Di Xue, Shouling Ji. H-Sim: A Concurrent Discrete-event Driven Sensor Network Simulation Platform. 2008 International Symposium on Information Science and Engineering. Pages 20-22, December 2008.
- [8] Anbao WANG, Wenrong JIANG. Research of Teaching on Network Course Based on NS-3. 2009 First International Workshop on Education Technology and Computer Science. Pages 629-632. 2009.
- [9] Allan Braga, Jasmine Araújo, Josiane Rodrigues. Implementation of a New Propagation Model for 5.8GHz systems in OPNET Simulator. 2013 7th European Conference on Antennas and Propagation (EuCAP). Pages 954-957. 2013.
- [10] Stelios Sotiriadis, Nik Bessis, Eleana Asimakopoulou. Towards Simulating the Internet of Things. 2014 28th International Conference on Advanced Information Networking and Applications Workshops. Pages 444-448. 2014.
- [11] V. Looga, Z. Ou, Y. Deng, and A. Yl'a-J'a'aski. Mammoth: A massivescale emulation platform for internet of things. In 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, volume 03, pages 1235–1239, October 2012.
- [12] Luis Jurado Pérez, Joaquín Salvachúa Rodríguez, etc. Simulation of Scalability in IoT Applications. 2018 International Conference on Information Networking (ICOIN), pages 577–582, January 2018.
- [13] G. D'Angelo, S. Ferretti, and V. Ghini. Simulation of the internet of things. In 2016 International Conference on High Performance Computing Simulation (HPCS), Pages 1–8.
- [14] Y. Sasson, D. Cavin, and A. Schiper. Probabilistic broadcast for flooding in wireless mobile ad hoc networks. In *Wireless Communications and Networking*, volume 2, pages 1124–1130 vol.2, March 2003.
- [15] T. Pflanzner; K. Zs. Leszko; A. Kertesz. SUMMON: Gathering smart city data to support IoT-Fog-Cloud simulations. 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC) , pages 71 – 78, April 2018
- [16] G. Brambilla, M. Picone, S. Cirani, M. Amoretti, and F. Zanichelli. A simulation platform for large-scale internet of things scenarios in urban environments. In *Proc. of the First International Conference on IoT in Urban Space, URB-IOT '14*, pages 50–55, ICST, 2014.
- [17] Gabor Kecskemeti, Giuliano Casale, Devki Nandan Jha. Modelling and Simulation Challenges in Internet of Things. *IEEE CLOUD COMPUTING*. pages 62-69. January 2017.
- [18] D. Singh, G. Tripathi, and A. J. Jara. A survey of internet-of-things: Future vision, architecture, challenges and services. In *Internet of things (WF-IoT)*, 2014 IEEE world forum on, pages 287–292. IEEE, 2014.
- [19] K. Perumalla. Tutorial: Parallel and distributed simulation: Traditional techniques and recent advances, 2007.
- [20] Stefano Ferretti, Gabriele D'Angelo, Vittorio Ghini, Moreno Marzolla. The Quest for Scalability and Accuracy: Multi-Level Simulation of the Internet of Things. *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2017)*. October 2017.
- [21] A. Gluhak, S. Krco, M. Nati, D. Pfisterer, N. Mitton, and T. Razafindralambo. A survey on facilities for experimental internet of things research. *IEEE Communications Magazine*, 49(11):58–67, November 2011.
- [22] Gabriele D'Angelo, Stefano Ferretti, Vittorio Ghini. Modeling the Internet of Things: a simulation perspective. *IEEE 2017 International Conference on High Performance Computing and Simulation (HPCS 2017)*, September 2017.
- [23] L. Magne, S. Rabut, and J.-F. Gabard. Towards an hybrid macro-micro traffic flow simulation model. In *INFORMS spring 2000 meeting*, 2000.
- [24] S. Sotiriadis, N. Bessis, E. Asimakopoulou, and N. Mustafee. Towards simulating the internet of things. In 28th Int. Conf. on Advanced Information Networking and Applications Workshops (WAINA), pages 444–448, May 2014. Conference Short Name: WOODSTOCK'18.
- [25] G. D'Angelo. The simulation model partitioning problem: an adaptive solution based on self-clustering. *Simulation Modelling Practice and Theory (SIMPAT)*, 70:1 – 20, 2017.
- [26] A. M. Law and D. M. Kelton. *Simulation Modeling and Analysis*. McGraw-Hill Higher Education, 3rd edition, 1999.