# Deriving Machine to Machine (M2M) Traffic Model from Communication Model

Basel Barakat[1], *Student Member, IEEE*, Simeon Keates[1], Kamran Arshad[3] *Senior Member, IEEE* and Ian J. Wassell[3]
[1] Faculty of Engineering and Science, University of Greenwich, Kent, ME4 4TB, UK
[2] Ajman University of Science and Technology, Ajman, Ajman, UAE
[3] Computer Laboratory, University of Cambridge, Cambridge CB3 0FD, UK
bb141@gre.ac.uk, s.keates@gre.ac.uk, k.arshad@ajman.ac.ae, ijw24@cam.ac.uk

*Abstract*—**The typical traffic models proposed in literature can be considered as heuristic models since they only reflect the stochastic characteristic of the generated traffic. In this paper, we propose a model for M2M communications that generates the traffic. Therefore, the proposed model is able to capture a wider picture than the state-of-the-art traffic models. The proposed model illustrates the behaviour of M2M uplink communication in a network with multiple-access limited information capacity shared channels. In this paper, we analyzed the number of transmitted packets using the traffic model extracted from our proposed communication model and compared it with the state-of-the-art traffic models using simulations. The simulation results show that the proposed model has a significantly higher accuracy in estimating the number of transmitted packets compared with the liteature model.**

***Keywords— Machine to Machine Communication; Communication Model; Communication System Traffic; Traffic Model; Stochastic Process.***

## I. INTRODUCTION

Machine-to-Machine (M2M) communication characteristics differ significantly from conventional Human-to-Human (H2H) communication (e.g., voice calls). The main differences are: (i) generally M2M applications generate short bursts of periodic data packets; (ii) most of the time M2M communications are established through uplink channels; (iii) M2M traffic is machine generated and does not usually rely on human intervention; (iv) Quality of Services (QoS) requirements for M2M communication differ significantly from H2H applications such as in the mobility, delay tolerance, volume of data, priorities, power consumption, security requirements, etc. Therefore, serving M2M Devices (M2MDs) over conventional networks built for H2H communications raises several challenges [1]–[3].

An accurate model of the M2M communication traffic is critical in the planning and implementing the network of M2MDs i.e., the Internet of Things (IoT). In particular, it plays a significant role in the following aspects: (i) evaluating the Radio Resources Allocation (RRA) algorithm's energy efficiency [4] and throughput performance; (ii) providing a better understanding of M2M communications and traffic to enable the networks designers to equip the network with the efficient resources to handle the predicted massive number of devices; (iii) several security threats can be avoided if we could identify the difference between the regular traffic transmitted by M2M devices and a malicious traffic generated by, e.g., a security breach, such as traffic generated by a Distributed Denial of Service attack.

In [5], the traffic models are classified into two main classes i.e. (i) *Source traffic models* and (ii) *Aggregated traffic models*. A *Source traffic model* considers the traffic of an individual M2MD, while an *Aggregated traffic model* considers the resulting traffic characteristics across a broadband network. The *Source traffic models* commonly used in the literature are extensions of the H2H traffic models [3], [4]. Hence, they cannot be used directly to model M2M traffic, because of the differences between the two types of communication discussed earlier. In particular, the models in the literature typically consider very particular scenarios. Consequently, they cannot be generalized for M2M traffic. Further discussion of the common models proposed in the literature and their shortcomings are presented in section II.

The main contribution of this paper is to propose a model for M2MD communication based on non-deterministic finite state machine communication. Unlike the heuristic traffic models proposed in the literature, in which the traffic arrivals rely only on the statistical characteristics of the traffic, the traffic for M2M communications in this paper is extracted from the proposed communication model (presented in section III). Therefore, the traffic derived from the proposed M2M communication model can capture a wider range of settings.

In particular, the extracted traffic from the proposed communication model is assumed to be affected by several factors (as shown in Fig.1). The first factor is the *channel information capacity*. That plays a major role in the data transmission time. Most traffic models available in the literature do not consider the information capacity as they are mainly based on the Erlang and Engest Models [2]–[5]. The Erlang and Engset models (i.e., Markov Modelled Poisson Process) were proposed for telephone networks (i.e., circuit switched networks) and are arguably not valid for M2M traffic. The second factor not accounted for in the existing M2M traffic models is the *Blocking Incident* in which the users require access to the shared channels, but the channels are already fully occupied [1]–[3], [5]. Additionally, the multiple-access mechanism is missing in the existing M2M traffic models [1]–[3], [5]. For a shared channel, there are two main multiple-access approaches as classified in [6] i.e., (i) *Centralized Scheduled Access,* in which a centralized device determines what part of the channel is allocated to each user e.g., cellular scheduling algorithms [7], and (ii) *Distributed Access,* in which each user locally decides the channel to access e.g., an ALOHA mechanism. Using the proposed communication model derived traffic achieves a significant enhancement (compared to traffic models proposed in the literature) in predicting the number of packets as shown in section IV. This paper is concluded in section V.

**978-1-5386-6224-3/18/$31.00 ©2018 Crown**

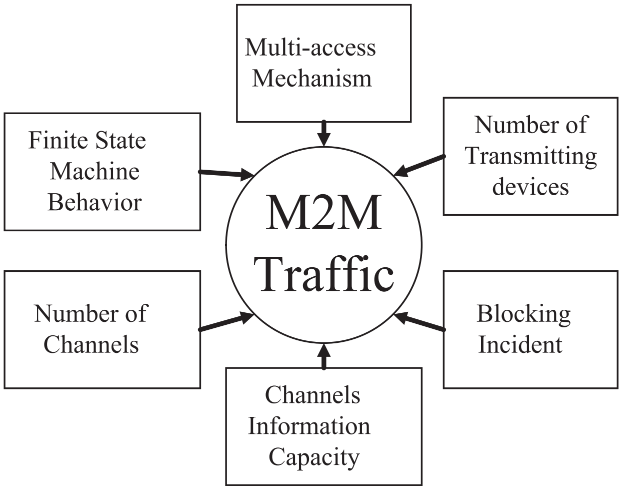
Multi-access Mechanism
Number of Transmitting devices
Blocking Incident
Channels Information Capacity
Number of Channels
Finite State Machine Behavior
M2M Traffic

Fig. 1 Factors affecting M2M communication traffic.

## II. Literature Survey of Existing M2M Traffic Models

### A. Fixed Scheduling and Event Driven M2MD traffic model

The authors in [1] proposed to classify the M2MDs traffic in two distinct models according to the periodicity of the transmissions. The first model studied the periodic update traffic referred to as Fixed Scheduling (FS) nodes e.g., regular reports of sensor measurements. The traffic generated by an FS node is typically assumed to follow a deterministic process representing the periodic transmissions. The second model focused on the non-periodic data traffic, referred to as Events-Driven (ED) nodes e.g., the report of an emergency alarm event. The traffic packets generated by the ED notes were modelled as a Poisson process.

Although the authors in [1] proposed a traffic model that highlights the distinctive characteristics that the M2M traffic, they made some simplifying assumptions that do not reflect the behaviour in a generalized sense. The first assumption was that the M2MDs can be either FS nodes or ED nodes. This assumption makes the model only applicable for some specific M2M devices. These devices are able to perform only a particular job, such as to periodically report the temperature, but it cannot report an incident in which the temperature is higher than a certain threshold. At the moment, in most practical systems most of the M2MDs can be both an FS node and an ED node. The assumption that all the FS nodes are synchronized could be considered as an over-simplifying assumption. The authors in [8] investigated the synchronization of machine-generated traffic, such as the routers status update messages which report the current link state. They demonstrated (analytically and empirically) that behaviour transition from asynchronized to synchronized is abrupt even if it was affected by an external influencer (such as turning them ON simultaneously). Synchronization in the case of M2MDs is a bigger challenge since M2MDs are typically connected to the network through a wireless connection, consequently, the inequality of propagation delays and bit errors play a significant role in inhibiting successful synchronization.

### B. Markov Modelled Poisson Process Models (MMPP)

A two-state Markov process (as shown in Fig. 2) is commonly used in the literature for modelling queue arrivals. The MMPP model was also used to model the M2M traffic arrivals [3], [4]. The first state i.e., '*Transmit*' or *'On'* represents an M2MD transmitting data. The second state i.e., '*Idle' or 'Off'* represents an M2MD not transmitting data, also known as *Sleep Mode*. $P_{I,T}$ represents the probability of changing from the *Idle* state to the *Transmit* state and $P_{T,I}$ changing from the *Transmit* state to the *Idle* state.

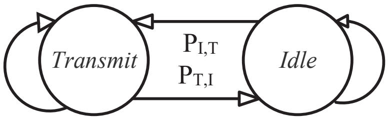

Fig. 2 MMPP model used in [2]–[5].

The data transmission in the MMPP model is obtained by the steady state probability (i.e., the probability of existing in a specific state when time $(t)$ approaches infinity) of the *Transmit* state. Consequently, the *Transmission Probability* $(P_T)$, can be expressed [9] as:

$$P_T = \lim_{t\to\infty}\left(P_{I,T}\right)^t = \lim_{t\to\infty}\left(1 - P_{T,I}\right)^t = \frac{P_{I,T}}{P_{I,T} + P_{T,I}} \qquad (1)$$

Although the two-state Markov chain model has been commonly used in the literature [2]–[5], it is not an optimum model for M2M traffic through a shared channel. This is because it depends on the Poisson inter-arrival process, and it does not hold true here as explained earlier. Further, M2MD data packets are typically non-homogenous and cannot be treated as identical nor represented as a state in a two-state Markov chain. For instance, a periodic update reporting an M2MD battery status and a picture of an intruder passing a security check cannot be treated as identical data transmissions, because of the size and nature of data to be transmitted. Further, this model does not consider *Blocking Incidents*, which occur frequently in networks of high user density such as IoT. A *Blocking Incident* occurs when the transmitter (i.e., the M2MD) has data to be transmitted, but all the channels designated for data in the network are fully occupied.

### C. Empirical Model

The empirical models rely on experiments and tests for their evaluation. Typically, the models proposed using this methodology start by running the experiment and afterwards trying to fit the data collected into an appropriate statistical distribution. Recently, the authors of [10] and [11] used an empirical approach and measured the M2M traffic in a cellular network. They concluded that the M2MDs' communications would have a significant impact on the smartphones' connectivity. In particular, an M2MD would compete with the smartphones for the available channels and, as a consequence, the *Blocking Probability* would increase.

Although empirical models illustrate the precise traffic measured in M2M communications, they have their shortcomings. For example, they are a reactive approach which can only describe a precise scenario for which data was collected. They are not able to suggest a generalized model. Additionally, the approach used in [10] can only model the aggregated traffic observed through the network.

## III. Proposed M2M Communication Model (*MCM*)

### A. Overview

As mentioned in Section I, in this paper we propose a communication model for M2M and use it to derive communications traffic that considers several factors. We assume M2MDs typically are low computational complexity finite state machines that mainly consist of sensor(s), a microprocessor/controller and a communication unit. The

978-1-5386-6224-3/18/$31.00 ©2018 Crown

M2MD 's main function is to monitor the environment and sends a report to a centralized node to analyze data. Fig. 3 illustrates a generic M2MD data communication flow chart. Initially, the M2MDs at startup monitors the environment (e.g., senses the motion in a room). After a predefined period, the M2MD sends a periodic update (i.e., Round Robin states update) to the base station or a centralized node. In the incident of a triggered interrupt (an event occurs e.g., a movement is detected), the M2MD also transmits exceptional, i.e., non-periodic, data to report it.

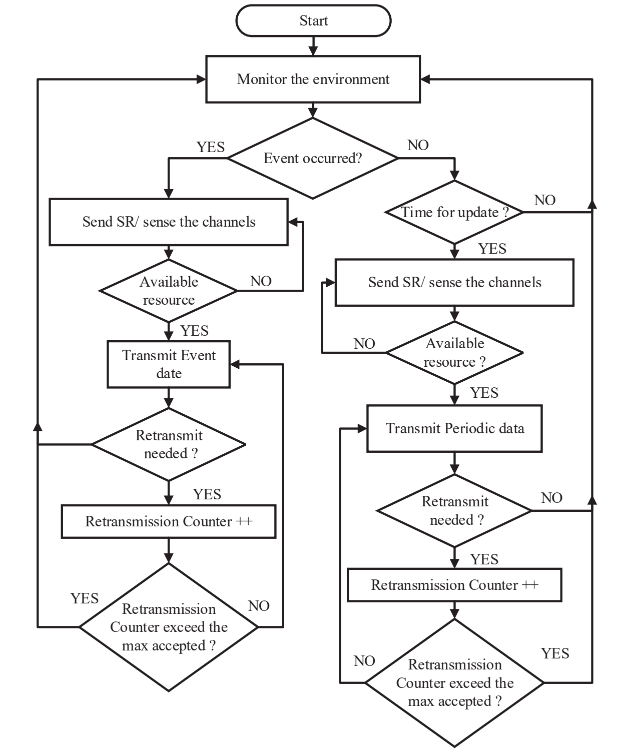

Fig. 3 Generic M2MD data communications flow chart. The flowchart shows the two types of data generated by an M2MD i.e. periodic updates and aperiodic event reporting data.

The proposed *M2MD's Communication Model* (*MCM*) is shown in Fig. 4. *MCM* is a discrete stochastic process that consists of four states: *Sleep (s)*, *Round Robin (r)*, *Interrupt* ($i$) and *Buffer (b)*. At any time the M2MD is considered to be in one of these four states and would change to another state with a certain probability refered to as the Transition Probability (TP). The TPs shown in Fig. 4 represent the *Starting State* and *Finishing State*. For example, a TP $P_{s,b}$ has *Starting State* $s$ and *Finishing State* $b$.

The *Sleep* state represents the starting state of the finite state machine in which the M2MD does not transmit any data. The *Round Robin* state represents the epoch in which the M2MD is transmitting routine periodic updates data, e.g., a periodic report of room temperature. During the *Buffer* state, the M2MD has data to be sent, but it is still waiting to access the shared channels to transmit it. Additionally, in the case of fully occupied channels, the M2MD buffers the data packets until it can access a channel. The *Interrupt* state represents an event occurring in the M2MD in which it sends data representing the event, e.g., a burglar alarm is activated.

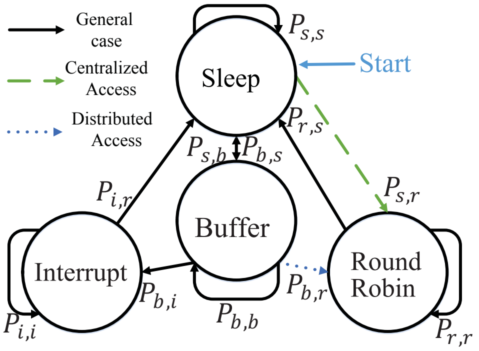

Fig. 4 Proposed M2MD's communication Model i.e. *MCM*.

In *MCM*, the data traffic is transmitted during two distinct states i.e., *Interrupt* and *Round Robin*. We differentiate between the two states for the following reasons: (i) typically, the data that has to be sent in the *Round Robin* updates is short data bursts; while in the *Interrupt* state data packet size is comparatively large. For instance, motion detectors would periodically send comparatively short data bursts (e.g., data sent containing the device identifier and, say, the battery state information). On the other hand, in the case of an exceptional event (e.g., a moving object had been detected), the M2MD would send a longer data burst that contains information of the event (e.g., a picture or the coordinates of the moving object); (ii) in the *Round Robin* state the communication is synchronized while communication in the *Interrupt* state is not; (iii) consequently, they would differ in their channel access approach, which relies on the network access technique.

### B. *MCM* Transitions

The *MCM* is modelled as a discrete stochastic process in which at each time unit a state transition occurs. The transition can be to any possible state (including the starting state itself). The TPs determine which state is the most likely to be moved to in the next time slot. The summation of the TPs going out of any state must equal to unity, as follows:

$$\begin{gathered} P_{s,s} + P_{s,r} + P_{s,b} = 1 \\ P_{b,b} + P_{b,i} + P_{b,s} + P_{b,r} = 1 \\ P_{r,r} + P_{r,s} = 1 \\ P_{i,i} + P_{i,s} = 1. \end{gathered} \tag{2}$$

The self-transition probabilities i.e., staying in the same state, rely on several factors. In particular, $P_{s,s}$, which represents the probability of remaining in the *Sleep* state, depends on the frequency of both the periodic updates and the event occurring. The availability of channel resources directly affects the value of $P_{b,b}$. In particular, $P_{b,b}$ value is equal to the channel instantaneous *Blocking Probability*. The length of the M2MD data packet and the channel quality, e.g., Signal to Noise Ratio (SNR), determines the value of both $P_{r,r}$ and $P_{i,i}$. Currently, we are only considering the SNR to be affecting the information data rate. Hence, the maximum achievable information rate by the $k^{\text{th}}$ M2MD in the $j^{\text{th}}$ channel ($R_{k,j}$) can be obtained by the Shannon capacity formula and can be obtained as:

$$R_{k,j} = BW \log_2\left(1 + SNR_{k,j}\right) \tag{3}$$

$$\begin{gathered} P_{(r,r)/(i,i)} = 1/\gamma(t)_{(r,r)/(i,i)} \\ \gamma(t)_{(r,r)/(i,i)} = \left\lceil DR_{(r)/(i)}/R_{k,j}(t) \right\rceil \end{gathered} \tag{4}$$

978-1-5386-6224-3/18/$31.00 ©2018 Crown

where $BW$is the channel bandwidth, $\gamma$ represents the number of time units the data needs to be transmitted, $\lceil . \rceil$ refers to the ceiling function, $t$ refers to the instantaneous time, and $DR$ is the state data requirements.

In a network with shared channels, there are two main multi-access techniques, as classified in [6]. The first technique is *Centralized Scheduling*, in which the M2MD must send a Scheduling Request (SR) to a centralized device (say, a Base Station) to access the channel. The Base Station controls the M2MDs channel's multiple-access scheduling [4]. The second technique is *Distributed Scheduling* where each M2MD makes a local decision whether it should access any particular channel based on channel sensing techniques, such as in [12].

In a *Centralized Scheduling network*, a central device (e.g., a Base Station) schedules the M2MD shared channel access. Consequently, the M2MD is required to send a Scheduling Request (SR) before starting to transmit data. After the Base Station receives the SR, it schedules a specified Resource (such as time and bandwidth pair) for the M2MD. Thus, when an interrupt occurs (i.e., asynchronized data transmission is required) the M2MD needs to store the data in its buffer (i.e., the *Buffer* state). The time duration that the data packets spend in the buffer represents the time of sending the SR to the Base Station, and for a resource to be scheduled. On the other hand, in *Round Robin* update, data packets are transmitted in a predefined epoch (i.e., at an explicitly defined time). Accordingly, the M2MD sends the SR to the Base Station in advance and M2MD periodic updates do not require data buffering.

However, in a *Distributed Scheduling* network all the data transmission (i.e., the data transmission is done by both the *Interrupt* and *Round Robin* states) has to be buffered until the M2MD senses the channel and determines an unoccupied channel, before transmitting the data. Table I illustrates both data communication types (i.e., *Round Robin* state data and *Interrupt* state data in both network channels multi-access approaches (i.e., *Centralized* and *Distributed Scheduling*).

Table I: Data procedures for both types of Network i.e., *Centralized* and *Distributed Scheduling*.

| **Data generating state** | **Data transmission procedure** |
|---|---|
| *Interrupt* | Initially, the M2MD is in the Sleep state. When the data is ready to be transmitted to the M2MD stores it in the Buffer. The M2MD remains in the buffer state until it either detects an unoccupied channel (for *Distributed Scheduling*) or it has been allocated a channel (for *Centralized Scheduling*). |
| *Round Robin* | In *Centralized Scheduling* the M2MD changes from the Sleep State (i.e. initial state) to the Round Robin State. In *Distributed Scheduling*, the M2MD changes from the initial state to the buffer state and stays there until it senses an unoccupied channel. |

As mentioned in II-B, the *Round Robin* updates occur in a predefined epoch, so in a *Centralized Scheduling* network $P_{s,r}$ follows a *Deterministic* distribution with a rate $\lambda$. It is worth mentioning that the assumption $P_{s,r}$ to be deterministic is only acceptable if the SR was sent in a sufficient time for the centralized device to allocate a channel resource to the M2MD. On the other hand, the interrupts occur in a random manner and hence $P_{s,b}$ can be modeled as a *Discrete Poisson* distribution with a mean $\mu$. $P_{b,s}$ represents the probability of the M2MD discarding the packets it has previously prepared to transmit. This incident occurs when the packets have been blocked for a period of time, therefore, the information represented in the packet is not relevant anymore.

The TPs in the *MCM* model can be represented as a *Transition Matrix (δ)*:

$$\delta = \begin{bmatrix} P_{s,s} & P_{s,r} & P_{s,b} & P_{s,i} \\ P_{r,s} & P_{r,r} & 0 & 0 \\ P_{b,s} & P_{b,r} & P_{b,b} & P_{b,i} \\ P_{i,s} & 0 & 0 & P_{i,i} \end{bmatrix} \tag{5}$$

where the probabilities in each row have the same *Starting State* and the probabilities in each column share the same *Finishing State*.

The steady-state probabilities of the *Sleep*, *Round Robin*, *Buffer* and *Interrupt* states are referred to as $\mathrm{P}_s, P_r, P_b$ and $P_i$ respectively. Accordingly, the steady-state probabilities can be expressed as a *Stationary Vector* ($Q$):

$$Q = [\, P_s \, P_r \;\; P_b \;\; P_i \,] \text{ where } P_s + P_r + P_b + P_i = 1. \tag{6}$$

The steady-state probabilities for the M2MD for the *MCM* can be obtained using the Balance equation:

$$\delta \times Q = Q \text{ or } Q(\delta - I) = 0 \tag{7}$$

where ($I$) is the identity matrix. Accordingly, from (5) and (6), the expression (7) can be represented as:

$$P_s(P_{s,s} - 1) + P_r(P_{r,s}) + P_i(P_{i,s}) = 0 \tag{8}$$

$$\mathrm{P_s}(\mathrm{P_{s,r}}) + \mathrm{P_r}(\mathrm{P_{r,r}} - 1) = 0 \tag{9}$$

$$\mathrm{P_s}(\mathrm{P_{s,b}}) + \mathrm{P_b}(\mathrm{P_{b,b}} - 1) = 0 \tag{10}$$

$$P_s(P_{s,i}) + P_b(P_{r,i}) + P_i(P_{i,i} - 1) = 0. \tag{11}$$

Finally, by solving (6), (8), (9), (10) and (11), the values of $Q$ and, hence, steady-state probabilities can be obtained.

M2MD only transmits data in two states i.e., *Round Robin* and *Interrupt*. Therefore, the number of transmitted packets ($NP$) can be derived from the *MCM* by using the probability of a device transmitting data ($P_T$) and the number of devices in the area of interest ($n$):

$$NP = P_T \times n \text{ where } P_T = P_r \cup P_i \tag{12}$$

## IV. EVALUATING THE NUMBER OF TRANSMITTED PACKETS

For simulating the M2MDs a discrete event simulator [13] was used to evaluate the network behaviour. In [6] it was shown that the *Distributed Scheduling* approach can outperform the *Centralized Scheduling* approach, where there is delayed Channel State Information (CSI). In a high user density network (such as a network handling many M2MDs), the probability of delay CSI is high, therefore, in this paper, we studied the packet transmission in a *Distributed Scheduling* network. The channels access probability is assumed to be equiprobable access across the M2MDs, i.e., all M2MDs are considered to have the same priority. For the simulations, five M2MDs (i.e., $n$ =5) sharing three channels is considered. The parameters and the associated values used to obtain the numerical and simulation results are given in Table II. The number of packets transmitted by M2MDs with respect to the time units is shown in Fig. 6. As shown in the figure, the *MCM* is able to model the simulated M2MD traffic more accurately. In particular, in the case of $\gamma_{(r,r)}$ and $\gamma_{(i,i)}$ are equal to unity and ten respectively i.e., SNR1, the *MCM* are able to predict the number of transmitted packets with a significantly higher accuracy. For instance, in SNR1 the number of packets achieved by simulation is $3 \times 10^4$ for the $5 \times 10^4$ time unit, and using the *MCM* is $3.041 \times 10^4$, which is less than 1.4% error. On the other hand, in the SNR2 scenario is not as accurate. However, using the MMPP model which does not adapt with respect to the SNR (referred to as Poisson in Fig. 6) the predicted number is $2.5 \times 10^4$ which is about 16.7% error.

TABLE II: NUMERICAL PARAMETERS AND VALUES.

| Parameter | Value |
|---|---|
| Simulation duration | $10 \times 10^4$ Time Unit |
| Number of M2MD $n$ / Channels | 5/ 3 |
| SNR 1 $\gamma_{(r,r)/(i,i)}$ | 1 for $(r,r)$/ 10 for $(i,i)$ |
| SNR 2 $\gamma_{(r,r)/(i,i)}$ | 3 for $(r,r)$ / 30 for $(i,i)$ |
| Round Robin updates distribution in *MCM* | Deterministic with a mean of 10 |
| Interrupts Distribution in *MCM* | Poisson with mean 50 |
| Data Requirements $DR_r$ / $DR_i$ | 150 / 1500 Kbit |
| $P_T$ for the Poisson model | Exponential distribution with mean 10 |
| $P_{b,s}$ | 0 |

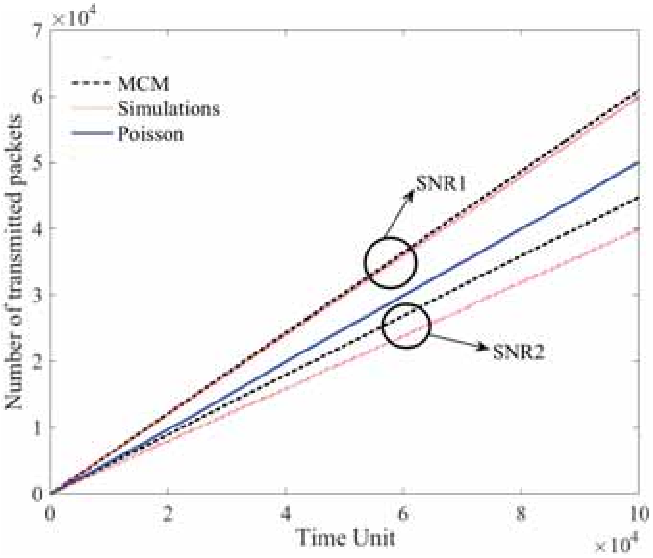

Fig. 6 Number of transmitted packets with respect to the time unit.

## V. CONCLUSIONS AND FUTURE WORK

In literature, several traffic models for M2M communications traffic have been proposed. Those models are able to represent M2M traffic for a specific set of scenarios, but, they do not cope well with a different set of scenarios. In this paper, we modelled M2M communications by looking more closely at M2MDs' behaviour. We then used the communication model to estimate the transmitted traffic. In the proposed method, the data traffic does not only rely on the statistical characteristics of the M2MD traffic. The extracted traffic has several other factors affecting it, such as the channel information capacity and multi-access technique used. We simulated the traffic commonly used in the literature using a discrete event simulator and compared it to the analytical results obtained by the proposed communication model. The results showed a significant improvement in predicting the number of packets with respect to time by using the proposed model.

Although this paper proposes a comprehensive model for M2M communications, further work can be done in the simulations. Hence, the simulation results presented in this paper reflect the data transmission in simplified theoretical channels. In particular, the information capacity is considered to be affected only by the SNR. Furthermore, the channels are assumed to have a static SNR. Further work can also be done in calculating the value of the TPs. In particular, we assume that the TPs are time-invariant; which might not be the case for systems that generate traffic according to a specific pattern.

## REFERENCES

[1] O. Al-Khatib, W. Hardjawana, and B. Vucetic, "Traffic modelling for Machine-to-Machine (M2M) last mile wireless access networks," *2014 IEEE Glob. Commun. Conf. GLOBECOM 2014*, pp. 1199–1204, 2014.

[2] C. Ide, B. Dusza, M. Putzke, C. Muller, and C. Wietfeld, "Influence of M2M communication on the physical resource utilization of LTE," in *Wireless Telecommunications Symposium*, 2012, pp. 1–6.

[3] N. Nikaein, M. Laner, K. Zhou, P. Svoboda, D. Drajic, M. Popovic, and S. Krco, "Simple traffic modeling framework for machine type communication," *10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013*, pp. 783–787, 2013.

[4] B. Barakat and K. Arshad, "Energy efficient scheduling in LTE-advanced for Machine Type Communication," in *2015 International Conference and Workshop on Computing and Communication, IEMCON 2015*, 2015.

[5] M. Laner, P. Svoboda, N. Nikaein, and M. Rupp, "Traffic models for machine type communications," *10th IEEE Int. Symp. Wirel. Commun. Syst. 2013, ISWCS 2013*, vol. 9, no. i, pp. 651–655, 2013.

[6] M. Johnston and E. Modiano, "A New Look at Wireless Scheduling with Delayed Information," *IEEE Int. Symp. Inf. Theory - Proc.*, pp. 1407–1411, 2015.

[7] B. Barakat and K. Arshad, "An adaptive hybrid scheduling algorithm for LTE-Advanced," in *2015 22nd International Conference on Telecommunications (ICT)*, 2015, pp. 91–95.

[8] S. Floyd and V. Jacobson, "The Synchronization of Periodic Routing Messages," *IEEE/ACM Trans. Netw.*, vol. 2, no. 2, pp. 122–136, 1994.

[9] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data networks*, vol. 2. Prentice Hall, 1992.

[10] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang, "A first look at cellular machine-to-machine traffic," in *ACM SIGMETRICS Performance Evaluation Review*, 2012, vol. 40, no. 1, p. 65.

[11] M. Z. Sha, L. Ji, S. Member, A. X. Liu, J. Pang, and J. Wang, "Large-Scale Measurement and Characterization of Cellular Machine-to-Machine Traffic," *IEEE/ACM Trans. Netw.*, vol. 21, no. 6, pp. 1960–1973, 2013.

[12] A. Aijaz and A. H. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet Things J.*, vol. 2, no. 2, pp. 103–112, 2015.

[13] Mathworks, "SimEvents®: User's Guide," *MATLAB Manual*. pp. 1–458, 2011.