Open Science Index, Electronics and Communication Engineering Vol:15, No:5, 2021 publications.waset.org/10012016.pdf

Random Access in IoT Using Naïve Bayes Classification

Alhusein Almahjoub, Dongyu Qiu

***Abstract*—**This paper deals with the random access procedure in next-generation networks and presents the solution to reduce total service time (TST) which is one of the most important performance metrics in current and future internet of things (IoT) based networks. The proposed solution focuses on the calculation of optimal transmission probability which maximizes the success probability and reduces TST. It uses the information of several idle preambles in every time slot, and based on it, it estimates the number of backlogged IoT devices using Naïve Bayes estimation which is a type of supervised learning in the machine learning domain. The estimation of backlogged devices is necessary since optimal transmission probability depends on it and the eNodeB does not have information about it. The simulations are carried out in MATLAB which verify that the proposed solution gives excellent performance.

***Keywords*—**Random access, LTE/LTE-A, 5G, machine learning, Naïve Bayes estimation.

I. Introduction

THE IoT is a vast cluster of devices linked together over wired or wireless networks that make it easier to combine physical and computer communication networks. Over the last decade, IoT has been increasing rapidly and the applications based on mobile devices, sensors, and actuators have grown smarter that enable the ease of networking between physical devices and cloud platforms. Smartphones, embedded systems, and almost every other gadget are linked with the internet. Generally, the data obtained from these devices are aggregated and processed to make possible decisions and correlations, progressing through machine learning (ML) algorithm to Artificial Intelligence (AI) [1].

Our everyday life is getting dependent on the IoT extending from smart household devices such as smoke detector, temperature sensors, smart meter, oven, refrigerator, IP camera, smart bulb, to more advanced devices such as accelerometer, parking lot sensors Radio Frequency Identification (RFID) devices and a variety of different sensors, etc. There are also too many applications of IoT in medical, transportation, manufacturing, energy, and many others, massive IoT networks around the world caries new challenges related to the management of these devices, communication, protection, and privacy, and computing [2]-[5]. IoT devices are communicating with each other and gather a massive amount of data every day. Some IoT applications are based on feedback or predefined conditions and to analyze the data some smart applications with some human involvement are required. IoT devices not only need to collect the data but also need to make decisions based on the feedback and acquire benefits from their collected data. A vast amount of data is generated by IoT devices, so conventional data collection, storage, and processing techniques are not suitable at this stage. Also, the variability of IoT-generated data provides another frontier for the existing mechanisms of data processing. Therefore, new mechanisms are needed to leverage the value of the IoT-generated data. For this, ML is considered one of the most acceptable computational models for the provision of embedded knowledge in IoT devices [6]. Google's Nest Learning Thermostat and Amazon Echo are examples of the ML in IoT. Google's Nest Learning Thermostat records the temperature and then applies some algorithms for the knowledge of the user temperature pattern and preferences, but voice and visual images cannot be perceived for by this device. Amazon echo works on the voice of the user and then converts it into words and further uses this information for searching appropriate information [7].

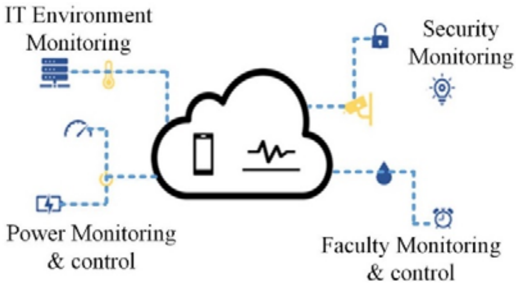
IT Environment Monitoring
Security Monitoring
Power Monitoring & control
Faculty Monitoring & control

Fig. 1 IoT devices everywhere

II. System Model

A. Random Access Procedure

The model that used in this paper is based on the random access procedure of LTE/LTE-A because this model is currently employed by 3GPP. The research is going on to have granted free random access procedure, and there are some procedures available in the literature, but they are not adopted by 3GPP. There are two types of random access processes currently adopted by LTE/LTE-A and in 5G.

- Four steps random access process
- Two steps random access process

A two-step random access process is derived from the four-step random access procedure, in which two steps are combined to reduce the number of steps from four to two. In this paper, we

Alhusein Almahjoub and Dongyu Qiu are with the Electrical and Computer Engineering, Concordia University, Montreal, Canada (e-mail: al.almahjoub@gmail.com, dongyu@ece.concordia.ca).

Open Science Index, Electronics and Communication Engineering Vol:15, No:5, 2021 publications.waset.org/10012016.pdf

consider four steps random access procedure in our model, since it can always be reduced to two steps. Before the start of the steps, the eNodeB (or gNodeB, as it is called in 5G) broadcasts the transmission probability p and the preambles. The IoT devices, each one of them, generate a random number locally and if the generated random number is less than the broadcast probability, they choose the preambles and transmit them to the eNodeB as message 1 in the first step of the random access process. The devices whose generated random numbers are greater than broadcast probability will wait for the next opportunity for transmission. The pool of preambles from which the IoT devices select their respective preambles is small and hence choosing and transmitting preambles based on the above procedure reduces the bottleneck of the system. After successfully receiving the transmissions from the IoT devices, the eNodeB responds with message 2 which is called random access response. Each IoT device gets a unique random access response message from eNodeB accept in the situation where more than one IoT devices select the same preamble. In this case same random access response message is sent to the IoT devices that selected the same preamble in message 1. After receiving random access response message from eNodeB, the IoT devices send another message, message 3, to the eNodeB which is called a connection request message. If eNodeB successfully decodes connection request messages from IoT devices, it responds with a contention resolution message, message 4, to the IoT devices. When IoT devices successfully receive contention resolution message from eNodeB, the random access process is said to be complete after which scheduled access starts. More than one IoT device may select the same preamble at the first step. Then the eNodeB is not able to decode connection request messages from those IoT devices, and hence it does not send contention resolution messages to those IoT devices. The pictorial form of the random access process is shown in Fig. 2.

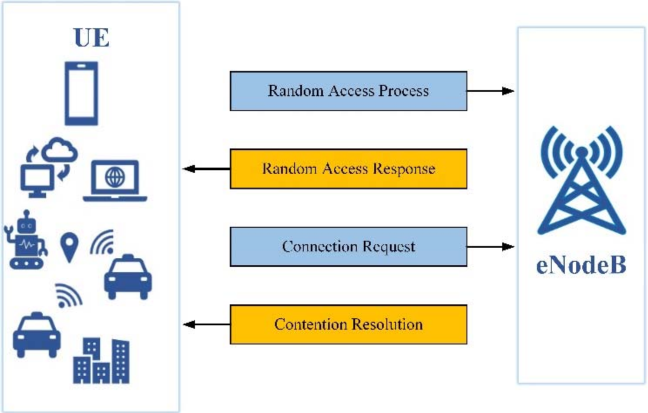

Fig. 2 Random access procedure in LTE/LTE-A

In the two-step random access process, messages 1 and 3 are combined and messages 2 and 4 are combined and the rest remains identical. Let us consider that M denotes the number of preambles and N represents the total number of IoT devices to be served. Time is divided into slots and the transmission of the messages takes place at the beginning of the slot just like slotted Aloha. The arrival process of IoT devices follows Beta distribution which is as follows:

$$f(t) = \frac{t^{\alpha}(T_A - t)^{\beta-1}}{T_A^{\alpha+\beta-1}\mathrm{B}(\alpha,\beta)} \tag{1}$$

where $\alpha = 3$, $\beta = 4$, and T_A is the activation time in which all the devices arrive in the system [8]. $\mathrm{B}(\alpha,\beta)$ denotes the Beta function which can be represented as:

$$\mathrm{B}(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt \tag{2}$$

Since T_A is the activation time which consists of slots, then the number of devices arriving in a one-time slot can be written as:

$$\lambda_i = N\int_{t_{i-1}}^{t_i} f(t)dt \tag{3}$$

It is important to note that any arrival distribution can be considered for IoT devices such as Poisson or uniform distributions. However, Beta distribution is chosen to represent the bursty nature of traffic as is the case when after the power failure, all the IoT devices try to connect to the network to transmit their data.

B. Optimal Transmission Probability

As mentioned above, before the first message transmission, each device compares its generated random number with broadcast transmission probability p . This transmission probability is very important as it controls the resources of the system. If p is very small, then a small number of IoT devices are allowed to transmit and some resources may go idle because the larger number remains silent in this scenario. However, if p is large, then a large number of IoT devices are allowed to transmit with the possibility that more than one IoT devices select a single preamble. Thus, with small p, the number of idle preambles increases whereas the number of collided preambles increases in case of large p. It is necessary to maximize the success probability and for that optimal probability needs to be found out.

As it is already implicitly mentioned in (3), the number of IoT devices in every time slot may change. This means that at any given time, there are some devices in the system, new and existing, but always less than N. Let us denote the number of such devices by n. The number of devices, whose generated random numbers are less than p and are allowed to transmit, is denoted by m. We can write the probability, $P(m|n)$, that m out of n devices are allowed to transmit as:

$$P(m|n) = \binom{n}{m}p^m(1-p)^{n-m} \tag{4}$$

The probability that a single preamble is successful is given as:

$$P(S_1) = \binom{m}{1}\frac{1}{M}\left(1-\frac{1}{M}\right)^{m-1} \tag{5}$$

The probability that any number of preambles are successful

Open Science Index, Electronics and Communication Engineering Vol:15, No:5, 2021 publications.waset.org/10012016.pdf

is given by:

$$P(S) = \sum_{m=0}^{n} MP(S_1)P(m|n) \tag{6}$$

Equation (6) can be simplified by using (4) and (5) as:

$$P(S) = np\left(1 - \frac{p}{M}\right)^{n-1} \tag{7}$$

We need to maximize the success probability and we can do it by taking derivative of (7), equating it to zero, and solving it for p. Then the optimal value of p can be written as:

$$p = \min\left(1, \frac{M}{n}\right) \tag{8}$$

We can see from (8) that the optimal p depends on the n which varies over time and the eNodeB cannot have information on n. So, it is necessary to estimate n and we employ ML

C. Estimation Using ML

ML [11] has been an active area of research quite recently. It is being used in almost every field in sciences and engineering. ML can be broadly divided into two broad categories:

- Supervised learning
- Unsupervised learning

Supervised ML builds a model that makes predictions based on evidence in the presence of uncertainty. In supervised ML, the data are trained on known input and known output which can help predicting a good response for new data. On the other hand, unsupervised learning finds hidden patterns or intrinsic structures in data. It is used to draw inferences from datasets consisting of input data without labeled responses. Classification and regression are subclasses of supervised learning and clustering is the subclass of unsupervised learning. Classification, regression, and clustering can further sub-divided into several techniques respectively. The complete picture representing ML, its associated subclasses, and their techniques is given in Fig. 3.

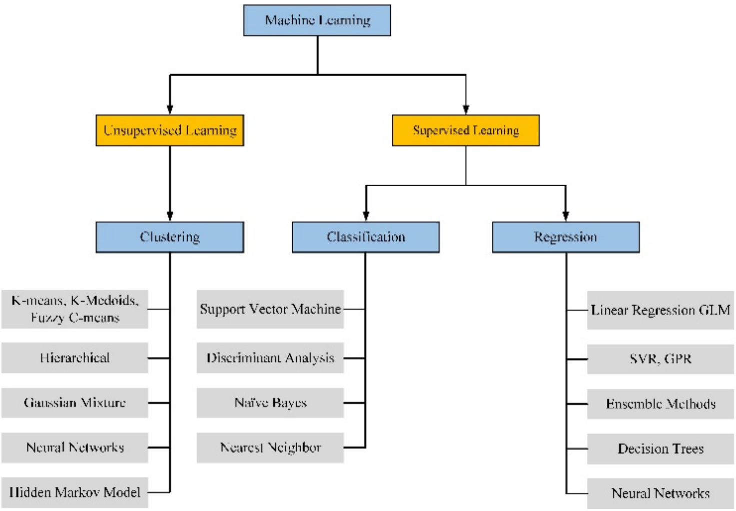

Fig. 3 ML and its subclasses [10]

In this paper, we use Naïve Bayes as our classification technique and we will use it to estimate the total number of devices in the system, n. Naïve Bayes is based on probabilities which are called apriori and aposteriori probabilities. Generally, aposteriori probabilities are based on the observation after the outcome of the experiment or simulation. In our work, the observation is taken to be the number of idle preambles in each slot. Without the loss of time let us write the Bayes rule as:

$$P(n|I) = \frac{P(I|n)P(n)}{P(I)} \tag{9}$$

where $P(I|n)$ is called aposteriori probability and $P(n)$ is called apriori probability. As explained above that at the third step, the eNodeB can tell that whether the preamble is selected by a single IoT device or more than devices have selected a preamble. So, in every slot, eNodeB cannot detect success or collided preambles. However, it can observe the number of idle preambles, and hence it can calculate the probabilities $P(I|n)$ and $P(I)$. Since the value of n varies over time, $P(I|n)$ may have different values for different n, i.e., $P(I|n_1, n_2, \cdots, n_n)$. $P(n)$ is assumed to have Poisson distribution as it is the number of backlogged IoT devices [11]. Then based on Bayes estimation, $P(n_1, n_2, \cdots, n_n|I)$ needs to be determined based on apriori and aposteriori probabilities. Then we select n for which the probability turns out to be maximum. Then we slightly modify n to have actual value since it is negatively Binomial distributed. We can have changed n, represented as $\bar{n}$ as [12]:

$$\bar{n} = \frac{n}{p_i} \tag{10}$$

Then this $\bar{n}$ can be used to calculate the transmission probability in the next time slot, p_{i+1}.

Algorithm 1: The Proposed Algorithm

```
1. Prepare the dataset of $P(I|n)$, $P(I)$, and $P(n)$
2. Initialize $n = 0$, $p = \max\left(1, \frac{M}{n}\right)$, $I = 0$
```

Open Science Index, Electronics and Communication Engineering Vol:15, No:5, 2021 publications.waset.org/10012016.pdf

```
3.  If the number of idle preambles is $I$, then
        Look in the dataset in $P(I|n)$ table
        Using $P(I)$ and $P(n)$, extract $n$ for maximum
        probability
4.  End if
5.  $\bar{n} = n/p_i$
6.  $p_{i+1} = \min\left(1, \frac{M}{\bar{n}}\right)$
```

III. Simulations and Discussions

For using the Naïve Bayes method as a classifier, we need the dataset which we prepared using extensive computer simulations. The value of N is taken as 1000 and the value of M is varied from 5 to 20. T_A is taken to be 500-time slots, as T_A is the activation time in which all the devices come into the system. The performance metric is taken as TST which is the total time required to serve all the N devices in the system. The analytical expression for TST is given as [13]:

$$TST = \frac{N}{M \cdot e^{-1}} \tag{11}$$

where, e^{-1} is the number of successful devices using one preamble and $M \cdot e^{-1}$ is the total number of successful devices using all the preambles.

For the training of data for $N = 1 \rightarrow 10$, we prepared the data set of the aposteriori probability $P(I|n)$ by taking each value of n. Similarly, the values of $P(I)$ and $P(n)$ are also calculated, and hence the dataset is prepared based on these three types of probabilities. Then the function of the classifier is to classify in which class the output falls. That is, based on the observation on I in the current slot and n in the previous slot we need to find the class n (current time slot) and then take the value with maximum probability. In simple words, after the training process, three tables are formed, one for each probability for all values of n. During the classification phase, by observing I in the current slot and n in the previous slot, look for the corresponding probability in $P(I|n)$ table. Also, we observe the probabilities $P(I)$ and $P(n)$. Multiplying these probabilities will give us the value of $P(n|I)$. Extract the value of n for which the probability is maximum. The process is summarized in Algorithm 1, and this algorithm runs every time slot until all the devices get served.

In Fig. 4, we plot the TST versus the number of preambles. $N = 1000$, and M is varied from 5 to 20. We see that as the number of preambles increases, the value of TST decreases. This result is obvious since more preambles mean the devices have a larger pool to select a preamble from. The collisions decrease, success increases, and hence TST decreases. There are three plots in this figure. The plot with the yellow color represents the scenario when optimal transmission probability is used. Optimal transmission probability means that the eNodeB has full knowledge of the backlogged devices and does not need any estimation to calculate p. The plot in the red color represents the scenario where the eNodeB estimates the number of backlogged devices using Naïve Bayes estimation, and the plot in the blue color represents the pseudo-Bayesian based estimation of backlogged devices discussed in [8]. We see that the estimation in [8] gives larger TST values as compared to the proposed technique. Moreover, overestimation is quite good as there is a negligible difference between optimal p and estimated p. Hence, we can say that the estimation works well even if compared with [8].

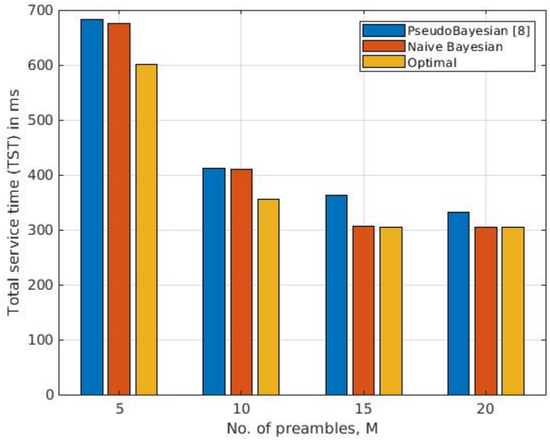

Fig. 4 TST vs M for N = 1000

In Fig. 5, we plot the number of backlogged IoT devices versus time. We see that the backlogged devices continue to increase until it reached the end of T_A after which the arrival of the new deices stops and the backlogged devices get served. Again, we compare the actual and estimated backlogged devices in blue and red colors respectively. We see that the estimated backlog follows the actual backlog. Hence the Bayes estimation keeps track of the actual number of backlogged devices which again signifies the effectiveness of the proposed technique.

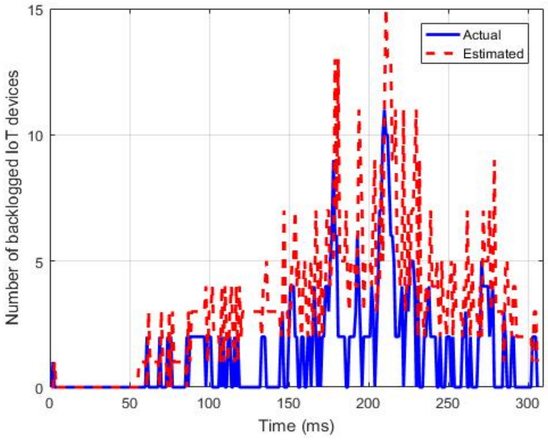

Fig. 5 Number of backlogged devices vs time for M = 15

In Fig. 6, we plot the transmission probability over time. We see that it forms a cup-shaped graph. At the start, the transmission probability is 1 because there are no backlogged IoT devices. As the time ticks, the number of backlogged devices increases, the transmission probability increases until it remains at its minimum value. After the arrival of new devices stops and the backlogged devices get success, the number of backlogged devices decreases, and the transmission probability increases until it becomes 1 again. The blue graph represents

Open Science Index, Electronics and Communication Engineering Vol:15, No:5, 2021 publications.waset.org/10012016.pdf

actual or optimal transmission probability while the red color represents the estimated one. Again, we see that the estimation is very close to the actual value and the estimated graph closely follows the actual graph.

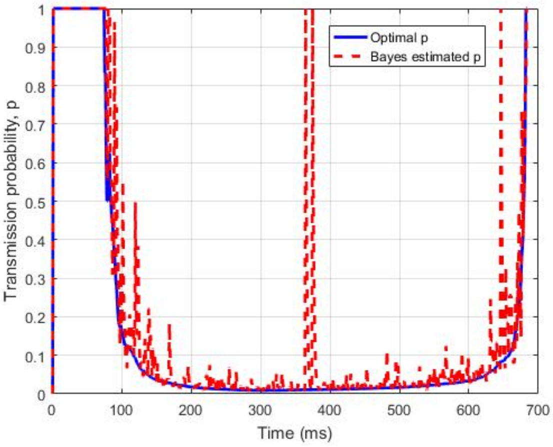

Fig. 6 Transmission probability vs time for M = 5

IV. Conclusion

In this paper, we consider the problem of random access of IoT devices in 5G and beyond. We consider the adopted random access procedure for our system model. TST is considered as a performance metric that increases as the number of IoT devices increases and can be reduced by using optimal transmission probability. The optimal transmission probability is based on the knowledge of the number of backlogged devices, however, unfortunately, the eNodeB does not have that knowledge. So, we propose an estimation based algorithm that estimates the number of backlogged devices and reduces TST by calculating and using estimated (close to optimal) transmission probability. The estimation is based on the Naïve Bayesian estimation technique which is a subclass of supervised learning in ML. We see in the results that our proposed algorithm works well.

Acknowledgment

We would like to thank anonymous reviewers for their valuable suggestions for improving the paper.

References

[1] Saadi, M., Noor, M.T., Imran, A., Toor, W.T., Mumtaz, S. and Wuttisittikulkij, L., 2020. IoT enabled quality of experience measurement for next generation networks in smart cities. *Sustainable Cities and Society*, *60*, p.102266.

[2] A. A. Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari and M. Ayyash, "Internet of Things: A survey on enabling technologies protocols and applications", *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347-2376, 4th Quart., 2015.

[3] J. Granjal, E. Monteiro and J. Sá Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues", *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294-1312, 3rd Quart. 2015.

[4] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "A survey on Internet of Things: Architecture enabling technologies security and privacy and applications", *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[5] K. A. da Costa, J. P. Papa, C. O. Lisboa, R. Munoz and V. H. C. de Albuquerque, "Internet of Things: A survey on machine learning-based intrusion detection approaches", *Comput. Netw.*, vol. 151, pp. 147-157, Mar. 2019.

[6] M. S. Mahdavinejad et al., "Machine learning for Internet of Things data analysis: A survey", *Digit. Commun. Netw.*, vol. 4, no. 3, pp. 161-175, Aug. 2018.

[7] S. T. Vieira, R. L. Rosa, D. R. Zegarra, A. Ramírez, M. Saadi, L. Wuttisittikulkij. Q-Meter: Quality Monitoring System for Telecommunication Services Based on Sentiment Analysis Using Deep Learning. *Sensors*, *21*(5), pp.1880, 2021

[8] H. Jin, W. T. Toor, B. C. Jung and J. B. Seo, "Recursive Pseudo-Bayesian Access Class Barring for M2M Communications in LTE Systems", *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8595-8599, Sept 2017.

[9] C. Tsai, C. Lai, M. Chiang and L. T. Yang, "Data mining for Internet of Things: A survey", *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 77-97, 1st Quart. 2014.

[10] https://www.mathworks.com/, accessed on January 2, 2021

[11] Alvi, M., Abualnaja, K.M., Toor, W.T. and Saadi, M., 2021. Performance analysis of access class barring for next generation IoT devices. *Alexandria Engineering Journal*, *60*(1), pp.615-627.

[12] M. Tavana,V. Shah-Mansouri, andV.W. S.Wong, "Congestion control for bursty M2M traffic in LTE networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2015, pp. 5815–5820.

[13] W. T. Toor and H. Jin, "Comparative study of access class barring and extended access barring for machine type communications", *Proc. ICTC*, pp. 604-609, Oct. 2017.