Class Based Dynamic Priority Scheduling for Uplink to Support M2M Communications in LTE

Mukesh Kumar Giluka[†], Nitish Rajoria[†], Ashish C. Kulkarni[††], Vanlin Sathya[†]and Bheemarjuna Reddy Tamma[†]

Indian Institute of Technology Hyderabad, India[†]

PESIT Bangalore, India[††]

Email: [cs11p1002, cs11m01]@iith.ac.in, ashishkulkarni09@gmail.com, [cs11p1003, tbr]@iith.ac.in

Abstract—Machine-to-Machine (M2M) communication has emerged as a key technology with huge market potential for cellular service providers deploying LTE networks. Addition of enormous number of M2M devices into the cellular networks poses a heavy competition to existing Human-to-Human (H2H) devices for getting radio resources, thereby affecting the performance of the H2H communications. But, one can not treat all M2M flows as low priority and schedule them after H2H flows, as there are many M2M applications like healthcare and tracking which are of high importance and delay-intolerant. Hence, there is a need for class based priority scheduling of the traffic of M2M and H2H sessions in the network. In this paper, we propose a class based dynamic priority scheduling algorithm for uplink transmission of M2M and H2H traffic in LTE. The performance of the algorithm is evaluated by various metrics such as H2H throughput and system throughput and also compared with existing schedulers.

I. INTRODUCTION

The Internet of things (IoT) [1] can be characterized by interconnecting uniquely identifying objects. Presently, IoT can be realised with the help of an Internet-like-structure. Machine-to-Machine (M2M) communication is an emerging technology which deals with networking part of IoT [1]. Typically, in IoT/M2M applications, M2M devices monitor events (*e.g.,* temperature, inventory level), which is relayed through a wireless communication network infrastructure to an application (software program running on a Server connected to the Internet), that translates the monitored event into meaningful information to be able to take collaborative decisions with limited or no human intervention. Examples include smart transportation, environment monitoring, smart grid, smart healthcare, and fleet management which will entirely transform how humans interact with physical world. According to the predictions [1] by the end of year 2020, approximately 20 billion IoT/M2M devices will be deployed on the planet Earth and offer various noble M2M applications by connecting to the Internet.

Because of ubiquitous coverage and global connectivity, cellular networks can play a major role in the deployment of M2M applications. Cellular networks can be used in two different ways for supporting M2M communications: (a) an M2M device having embedded cellular radio sends data directly to the M2M server located on Internet through a cellular network. (b) an M2M device having embedded Wi-Fi or ZigBee radio first sends data to an M2M gateway and then gateway forwards data (typically after aggregation) to the

M2M server through a cellular network. Former is called as cellular M2M communication while later is called as capillary M2M communication [1]. In capillary M2M, gateway needs dual radios for supporting connection to cellular and Wi-Fi/ZigBee networks. In this paper, we will deal with only cellular M2M communication scenario.

Presently, cellular networks are optimized for Human-to-Human (H2H) and Human-to-Machine (H2M) communications and in future, these seem to be uninterrupted because of the percentage of revenue contributed by these applications. At the same time, characteristics of H2H/H2M communication are different from that of M2M in terms of traffic volume, uplink to downlink traffic ratio, density of devices in a particular geographical area and mobility. With these differences, supporting M2M communication in cellular networks is a major challenge to the telecom network operators. Among the several challenges to be dealt due to the incorporation of M2M applications in cellular networks, efficient allocation of radio resources between M2M applications and H2H/H2M applications is one of the biggest challenges. Due to the limited available radio spectrum and enormous number of M2M devices (*i.e.,* device running M2M applications), existing resource scheduling algorithms do not scale well. In this work, we propose a Class Based Dynamic Priority (CBDP) scheduling algorithm for uplink (*i.e.,* device → base station) communication in LTE systems, in which, based on the priority of the class to which M2M or H2H/H2M application belongs to, the radio resources are allocated.

In LTE (Long Term Evolution) networks [2], bandwidth resources are allocated in terms of Resource Blocks (RBs). A RB is a smallest unit of bandwidth resource to be allocated to users. In LTE, time is measured in terms of frames. A frame consists of 10 subframes. Duration of a subframe is $1ms$. RBs are allocated by the scheduler for a TTI (Transmission time interval) where TTI is the duration of a subframe.

Rest of the paper is organized as follows: In Sections II, related work is discussed. In Section III, proposed work has been explained. Simulation setup and performance evaluation of the algorithm are given in Section IV and finally, Section V concludes the paper.

II. RELATED WORK

In this section, we review existing works addressing scheduling issues due to incorporation of M2M in LTE sys-

tems. In [3], all the applications are divided into four classes. For each class, an utility function is defined where user utility is a function of achievable data rate. The main objective of their scheduling algorithm is to maximize the aggregate utility of the function so that aggregate throughput can also be maximized. But, fairness and priority of a device are not taken into account during scheduling process. If a device has very less achievable data rate with respect to some RB then this RB will not be allocated to that device because it will not lead to increase in the aggregate utility. At the same time, if a device is running a delay-intolerant application but assigning a RB is not maximizing the utility then the device may remain unallocated. In [4], in order to distribute radio resources between H2H and M2M flows two algorithms have been proposed. H2H flows are given high priority over M2M flows in both algorithms. So, resources are allocated first to all H2H flows and if remaining, resources are allocated to M2M flows. The first algorithm gives higher priority to the signal to interference and noise ratio (SINR) value at a RB with respect to M2M device, in comparison to delay tolerance level during the allocation of RBs to M2M devices. The second algorithm gives higher priority to delay tolerance level than the SINR value. The main drawback of these algorithms is that they do not allocate RBs to M2M devices based on the applications they are running. It does not differentiate the delay tolerant and delay intolerant M2M devices and therefore efficient allocation of RBs in not done.

In [5], authors define a cluster of M2M devices based on the parameters: packet arrival rate and maximum tolerable jitter. An M2M device belongs to a cluster if both the device and cluster have identical values of these parameters. A cluster with larger packet arrival rate has high priority. Depending on the traffic rate and priority of cluster, a fixed access grant time interval will be allocated to clusters. The main drawback of the paper is that it consider traffic arrival rate as constant while in reality, M2M traffic can be random in nature. In [6], authors have attempted to remove the drawbacks of [5] but they did not consider the cases when M2M applications should be given more preferences over H2H applications.

In this paper, we have proposed a scheduling algorithm which considers delay tolerance and minimum guaranteed bit rate of applications and preference of H2H flows over M2M flows to schedule RBs. Apart from this, we propose variable chunk size based method for contiguous allocation of RBs to an user for uplink transmission.

III. Proposed Work

Different applications have different QoS requirements in nature. In [7], authors have classified all applications (both H2H and M2M) based on the parameters delay tolerance/delay sensitivity, accuracy and priority. Except the parameter delay tolerance/delay sensitivity, other parameters can be taken care by higher layers and can be ensured by mutual communication of end devices. But as far as scheduling of radio resources in wireless medium is concerned, delay sensitivity of the application is the most important parameter to deal with. With

respect to a radio resource scheduler, types of applications can be described as following: (1) Type 1: Applications that send very small amount of data but delay intolerant in nature. For example, emergency alerting, signalling messages. (2) Type 2: Applications that need a minimum guaranteed bit rate (MGBR). These application have a delay tolerance time (DTT) within which a certain amount of data must be scheduled by the scheduler to meet the MGBR. For example, streaming video, VOIP, and IP enabled surveillance cameras. (3) Type 3: Applications that have bulk data to send but are delay tolerant in nature. For example, file downloads. (4) Type 4: Applications that send very small amount of data and delay tolerant in nature. For example, environment monitoring M2M applications.

In this paper, we propose delay aware radio resource scheduler to support M2M traffic without affecting or least affecting H2H traffic. The goal of designing such a scheduler is to address following issues: (1) to support delay intolerant H2H or M2M flows. (2) to ensure MGBR of type 2 applications discussed in last paragraph. (3) to give preference to H2H flows over M2M flows in case of similar QoS requirements. (4) to postpone the scheduling of delay tolerant flows if enough radio resources are not available.

In order to achieve above goals, we first classify all flows based on their remaining time to serve (RTTS) within which the flow must be served by the scheduler to meet the DTT of the flow. If DTT of a flow is 30 ms then initially its RTTS will also be 30 ms but if the flow does not get served till 5 ms then its RTTS will become 25 ms. All delay intolerant applications (type 1) are kept in class 1 while all delay tolerant applications (type 3 and type 4) are kept in class n (last class). Other applications or flows are kept between class 2 to class *n-1*. Except class n, a class is defined by a range of RTTSs (say [20-30] ms). If RTTS of a flow lies within this range, then it means that the flow belong to that class. So, if RTTS of a flow is 25 ms then it lies in this class. A class is given higher priority than other class if higher value of its RTTS range is lesser than the lower value of RTTS range of other class. So, if class 1 has RTTS range [10-20] ms and class 2 has RTTS range [20-30] ms then priority of class 1 is higher than class 2. A class with high priority will be served first by the scheduler than a low priority class. If a flow belonging to low priority class is not served within the RTTS range of the class then the flow will automatically get shifted to next high priority class. In a TTI, if a flow is not being served then its RTTS value decreases by 1 ms. Since type 3 and type 4 applications are delay tolerant in nature, in class n, all delay tolerant applications are kept. These applications or flows will not have any RTTS value. So, a flow belonging to class n will remain in the same class.

Algorithm 1 lists out the proposed Class Based Dynamic Priority (CBDP) scheduling algorithm. The algorithm running at the base station (BS) allocates RBs to flows on per TTI basis. The algorithm takes following parameters as input: (i) A set of all flows which are waiting for uplink RB scheduling by the BS. All flows will have RTTS except flows generated

Algorithm 1 CBDP Scheduling Algorithm
_____
Input: Set S of all unsatisfied flows from last TTI and new
    flows in current TTI with their RTTS.
Output: Allocation of RBs to flows, Set of unsatisfied flows
  1: $U_{RB} \leftarrow$ Number of free RBs
  2: Classify all H2H and M2M flows, based on their RTTS,
     into classes 1,2,...,n.
  3: **while** $U_{RB} \neq 0$ **do**
  4:    Allocate RBs to all H2H flows of class 1 as per RTTS
  5:    Allocate RBs to all M2M flows of class 1 as per RTTS
  6:    Allocate RBs to all H2H flows from class 2 to class $n-1$
        as per RTTS
  7:    Allocate RBs to all M2M flows from class 2 to class $n-$
        1 as per RTTS
  8:    Allocate RBs to all H2H flows of class n
  9:    Allocate RBs to all M2M flows of class n
 10: **end while**
 11: Update RTTS values of all unsatisfied or partially satisfied
     flows except flows of class n
_____

by type 3 and type 4 applications. (ii) Range of RTTS of
each class (except class n). With the help of these parameters,
the scheduler assigns classes to all flows in each TTI (before
scheduling the RBs). A class not only contains those flows
which originally belong to that class but also contains those
flows who originally belong to some low priority classes but
because of dynamic priority concept, after sufficient decrement
in their RTTS values (because of not being served for some
TTIs), they moved to this class. In a class, a flow having
minimum RTTS will be served first.

The algorithm first serves all the flows of class 1 so that
delay intolerant applications can meet their delay constraints.
In class 1, all H2H flows are served before M2M flows in
the ascending order of RTTS values. The motivation behind
giving preference to H2H flows is not to affect them because
of M2M flows of same QoS requirements. Now, from class 2
to class $n-1$, all H2H flows will be served first. After serving
H2H flows, again from class 2 to class $n-1$ all M2M flows
will be served. Since, a flow (M2M or H2H) belonging to any
class in between class 2 to class $n-1$ can tolerate some amount
of delay in allocation of resources, in order to give preference
over M2M flows, all H2H flows belonging to class 2 to $n-1$
are allocated RBs first and then allocation for M2M flows
takesplace.

The scheduler will schedule at least $MGBR*DTT$ amount
of data in DTT for all flows belonging to class 2 to $n-1$.
If RBs are available after serving all classes, more can be
allocated to flows of these classes. For the flows of class n,
neither they have any delay constraint like class 1 flows nor
they have to meet any MGBR. So, the scheduler allocates RBs
to these flows in best-effort manner.

Figure 1 shows an example of CBDP scheduling order for
four classes of H2H and M2M flows in the network. Here,
class 1 contains flows of type 1 applications. Class 2 and

class 3 contain flows of type 2 applications. Class 4 contains
flows of type 3 and type 4 applications. The arrow shows the
order in which scheduler allocates RBs to flows belonging to
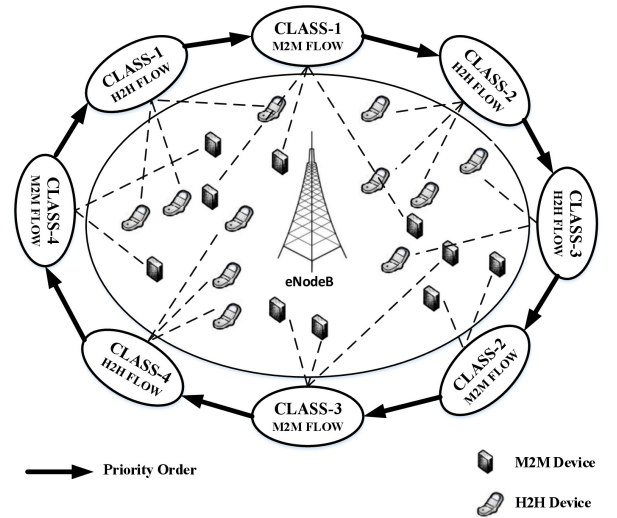various classes of M2M and H2H flows.

Fig. 1. An example of CBDP scheduling order for four classes of H2H and
M2M flows

A. Contiguity Constraint for Uplink

The major constraint with LTE uplink resource scheduling
is that the number of RBs assigned to a flow must be con-
tiguous because of Single-carrier Frequency Division Multiple
Access (SC-FDMA)[8] scheme for uplink channel access.
Maximization of throughput with the constraint of contiguous
RB allocation is a NP-Hard problem. Hence, in this paper, we
propose variable chunk size based uplink resource allocation
heuristic scheme where a flow is assigned best available chunk
of contiguous RBs to upload its data. Size of a chunk is the
number of contiguous RBs required to send the desired amount
of data. Size of a chunk, to meet demand of a flow, will be
different for different sets of RBs because of change in CQI
(channel quality information) with respect to RBs and user lo-
cation. The Algorithm 2, the Contiguity Constraint Algorithm
(CCA), is used to ensure the contiguous allocation of RBs
to requests generated by flows in Algorithm 1. Algorithm 2
is called by Algorithm 1 in steps $4-9$. In CCA, R denotes
a set of unallocated, contiguous RBs and $NEED$ denotes a
two dimensional matrix where $(NEED)_{ij}$ is the chunk size
needed to meet the demand of flow i, if first RB of the chunk
is j^{th} RB of the set R. So, row i contains the value of required
chunk sizes to send data of flow i with respect to each RB
(*i.e.,* as a first RB of the chunk) of set R. Now the algorithm
will choose a chunk of minimum size with which the flow i
can send its data. If the chosen chunk is already assigned to
some request then the algorithm will choose next minimum

size chunk and will check its availability. If the chosen chunk is free then it will be assigned to flow i and RBs of the chunk will be marked as *allocated* in the set R. After allocating RBs to flow i, the algorithm will move to schedule $(i+1)^{th}$ flow.

Algorithm 2 Contiguity Constraint Algorithm

Input: $NEED$ matrix
Output: Allocation of RBs to requests, Set of unserved H2H requests with updated RTTS
1: R is the set of not yet assigned RBs { Initially it contains the whole set of RBs in the network}
2: $NEED$ is a two dimensional matrix where $(NEED)_{ij}$ is number of RBs required by flow i to send its data if RBs are allocated starting from j^{th} RB of set R
3: **while** All RBs of R not allocated **do**
4:     Find minimum value of $(NEED)_{ij}$ for flow i { Minimum value in the ith row}
5:     Assign all RBs between j^{th} and $(j+(NEED)_{ij}-1)^{th}$ RBs of set R to flow i
6:     Mark all RBs between j^{th} and $(j+(NEED)_{ij}-1)^{th}$ RBs of set R as *allocated*
7:     Assign ∞ to all elements of columns of $NEED$ matrix from j^{th} to $(j+(NEED)_{ij}-1)^{th}$ column
8:     $i \leftarrow i+1$ {Schedule next flow in the order}
9: **end while**

IV. Simulation Results and Analysis

In this section, the performance of proposed CBDP algorithm is evaluated using system level simulations in NS-3.16 network simulator [9]. The simulation parameters are specified in Table I and parameters not specified here are assumed to be the default ones mentioned in 3GPP specifications [2]. The performance evaluation is done with a single-cell environment with an omni-directional antenna. The bandwidth on the uplink is 10 MHz, sub-divided into 50 RBs. All the devices are randomly located in a radius of 1km using Random Disc Position Allocator Model. Table II and Table III show the simulation parameters of H2H and M2M flows, respectively.

TABLE I
SIMULATION SETUP

Simulator	NS-3.16
System Bandwidth	10 MHz
Cellular Layout	Single-Cell with Omni-directional Antenna
No. of RBs in a TTI	50
TTI Duration	1 ms
Total No. of Devices	80
UE-eNodeB Min. Distance	10 m
Simulation Time	1 sec

We have compared the performance of proposed CBDP algorithm with proportional fair (PF) and round-robin (RR) scheduling algorithms. Initially, we have analyzed how throughput of H2H flows get affected with the incorporation of M2M flows in the network. For that we have compared

TABLE II
SIMULATION PARAMETERS FOR H2H FLOWS

Class		H2H			
Class Number	Range of RTTS	Application	MGBR	DTT	% of H2H Flows
1	0-20ms	IMS Signaling	-	10ms	10%
2	20-100ms	Voice, Gaming	10kbps, 10kbps	25ms, 50ms	30%,20%
3	100-500ms	Video	100kbps	110ms	20%
4	Rest	Web Browsing	-	-	20%

TABLE III
SIMULATION PARAMETERS FOR M2M FLOWS

Class		M2M			
Class Number	Range of RTTS	Application	MGBR	DTT	% of M2M Flows
1	0-20ms	Emergency	-	10ms	10%
2	20-100ms	-	-	-	-
3	100-500ms	Surveilance Camera	100kbps	110ms	15%
4	Rest	Regular Monitoring	-	-	75%

the throughput of H2H flows in CBDP and PF schedulers. After that we evaluated class wise throughput of all H2H and M2M flows for the proposed CBDP scheduler. At the end, we compared the system throughput or aggregate throughput with respect to all three schedulers.

In Figure 2, we have analyzed how H2H flows are getting affected in terms of aggregate throughputs when number of M2M flows increase. Here, we have taken 20 H2H flows for CBDP and PF schedulers. Number of M2M flows is increased from 0 to 60. We can observe that CBDP algorithm performs better in comparison to PF algorithm. This is because of the fact that in case of CBDP, H2H flows are served first and then M2M flows are served while in case of PF scheduler, resources are shared between H2H and M2M flows equally which results into the degradation of throughput of H2H flows.

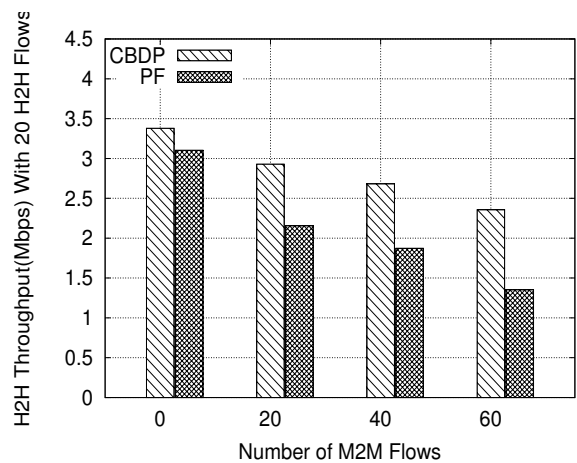In Figure 3, classwise throughputs for different number of

Fig. 2.   Comparison of aggregate throughputs for H2H flows

flows have been shown for proposed CBDP scheduler. A class may contain both H2H and M2M flows. Since, in class 1 both H2H and M2M flows have very small amount of data to send, throughput of class 1 is not visible in the graph. But practically, all flows of class 1 are satisfied. We can observe from the graph that throughputs of class 2 and class 3 increase with the increasing of total number of flows. But throughput of class 4 increases up to 40 flows but after that it decreases. Since, in class 2 and class 3 we are ensuring a minimum guaranteed bit rate, so throughput increases while in class 4, flows are served in best effort manner. Since, class 4 contains all delay tolerant flows, so they are scheduled later when RBs are free due to lack of higher class traffic.

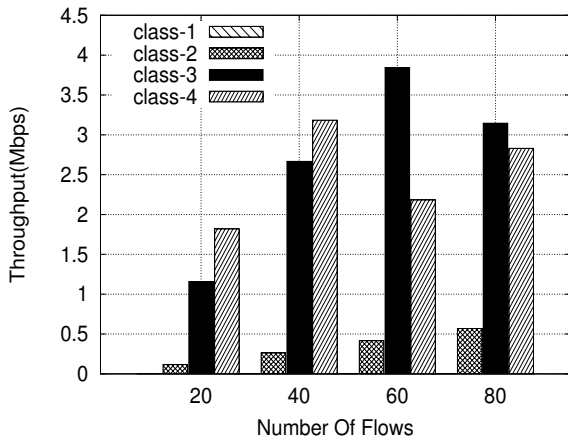In Figure 4, aggregate throughputs of all flows (H2H and

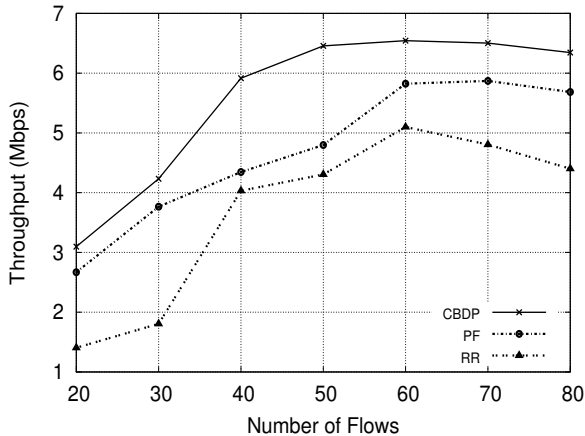Fig. 3.   Comparison of classwise aggregate throughputs in CBDP scheduler

Fig. 4.   Comparison of Sytem throughput of CBDP, PF and RR schedulers

M2M both) with respect to all schedulers considered have been shown. Here, the CBDP algorithm performs better than PF. The main reason for this is that in the case of PF, to ensure the contiguous allocation of RBs, a flow is assigned a chunk of constant size. So, it is possible that some of RBs in a chunk will remain unallocated if data to be sent by the flow is finished

after utilizing only a some of RBs of the chunk. But in the case of CBDP algorithm, chunk with variable size is allocated to ensure the contiguity of RBs. Because of that same amount of data can be sent with less number of RBs and therefore efficient RB utilization is achieved. Quantatively we can say that, depending on total number of flows, throughput shown by CBDP scheduler is 27-100% more than RR scheduler while in case of PF scheduler, it is 7-40%. From above results, we can conclude that CBDP algorithm performs best in terms of supporting M2M devices with least affecting H2H devices.

V. Conclusion

Support of IoT/M2M applications in existing cellular networking technologies is an interesting and challenging problem. If proper radio resource allocation schemes are not employed, these M2M applications could drastically impact the performance of H2H/H2M applications and the telecom operators may end up facing customer churn. The main goal of the proposed class based dynamic priority (CBDP) scheduling for uplink, was to support M2M communications while least affecting H2H communications. We compared the proposed CBDP algorithm with PF and RR scheduling algorithms and found that even in presence of M2M flows, CBDP algorithm not only satisfied the QoS requirements of H2H flows but also satisfied QoS requirements of M2M flows.

As a future work, we are planning to simulate the algorithm for thousand M2M flows with more specific M2M applications. Apart from this, we will calculate the extra signalling overhead incurred by the algorithms.

REFERENCES

[1] Min Chen, Jiafu Wan and Fang Li, "Machine-to-Machine Communications: Architectures, Standards and Applications", *KSII Transactions on Internet and Information Systems*, vol. 6, no. 2, February 2012.
[2] 3GPP TS 36.211 V8.7.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8)", June 2009.
[3] Kan Zheng, Fanglong Hu, Wenbo Wang, Wei Xiang and Mischa Dohler, "Radio Resource Allocation in LTE-Advanced Cellular Networks with M2M Communications", *IEEE Communications Magazine*, vol. 50, no. 7, pp. 184-192, July 2012.
[4] Athanasios S. Lioumpas and Angeliki Alexiou, "Uplink Scheduling for Machine-to-Machine Communications in LTE-Based Cellular Systems", *in Proc. of IEEE GLOBECOM Workshop on Machine-to-Machine Communication*, December 2011.
[5] Shao-Yu Lien, Kwang-Cheng Chen and Yonghua Lin, "Toward Ubiquitous Massive Accesses in 3GPP Machine-to-Machine Communications", *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66-74, April 2011.
[6] Antonis G. Gotsis, Athanasios S. Lioumpas and Angeliki Alexiou, "Analytical Modeling and Performance Evaluation of Realistic Time-Controlled M2M Scheduling over LTE Cellular Networks", *European Transactions on Telecommunications*, vol. 24, no. 4, pp. 378-388, June 2013.
[7] Rongduo Liu, Wei Wu, Hao Zhu, Dacheng Yang, "M2M-Oriented QoS Categorization in Cellular Network", *in Proc. of WiCOM 2011*, September 2011.
[8] Jeongchan Kim, Donggeun Kim, Youngnam Han, "Proportional Fair Scheduling Algorithm for SC-FDMA in LTE Uplink", *in Proc. of IEEE Global Communications Conference (GLOBECOM)*, December 2012.
[9] Network Simulator. Available online at:*"http://www.nsnam.org"*.