Class Based Priority Scheduling to Support Machine to Machine Communications in LTE Systems

Mukesh Kumar Giluka, Sharath Kumar N, Nitish Rajoria and Bheemarjuna Reddy Tamma
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad, India
Email: [cs11p1002, cs09b021, cs11m01, tbr]@iith.ac.in

***Abstract*—Due to the ubiquitous coverage and seamless connectivity, cellular systems are very promising to support Machine-to-Machine (M2M) communications. But, all of the cellular networks are designed and optimized for Human-to-Human (H2H) or Human-to-Machine (H2M) communications and therefore facing several challenges due to incorporation of M2M communications. One of such challenges is efficient resource allocation to M2M applications without affecting or least affecting H2H applications. In order to address this challenge, we need application specific priority based scheduling algorithms in which based on the QoS of the application, radio resources are allocated.**

In this paper, we have classified and prioritized all H2H and M2M flows based on their QoS requirements. Resources are allocated first to higher priority classes and in a given class, they are allocated to H2H flows first. In order to ensure the QoS of H2H flows, a threshold is kept on the maximum number of radio resource blocks to be assigned to M2M flows in a scheduling interval. Performance of the proposed scheduling algorithm is evaluated using various metrics such as system throughput and average utility per class and compared against existing scheduling schemes.

I. Introduction

The concept of Internet of things (IoT) [1] characterizes the interconnection of uniquely identifying objects. In present scenario, IoT can be realised with the help of an Internet-like-structure. Machine-to-Machine (M2M) communication is an emerging technology which deals with communication networking part of IoT system [1]. It provides ubiquitous networking to connect devices, running some specific applications, so that they can communicate with each other to take collaborative decisions with limited or without any human intervention.

In a typical M2M scenario, an end user machine called M2M device, communicates with an another machine called M2M server, situated very far from it, through some communication network. Cellular networks are best choice as communication network, to support M2M communication, because of their ubiquitous coverage and seamless connectivity. In M2M communications, cellular networks can be used in following ways:

1) An M2M device sends data to M2M server directly through cellular network. It is called as the cellular M2M communication.
2) An M2M device first sends data to an M2M Gateway which inturn forwards the data (typically after aggregation) to the M2M server through a cellular network. It is called as the capillary M2M communication [1], [2].

978-1-4799-2361-8/14/$31.00

Presently, cellular networks are designed to support Human-to-Human (H2H) or Human-to-Machine (H2M) communications. But, characteristics of M2M applications are different from H2H applications in terms of high uplink to downlink traffic ratio, low traffic volume, limited mobility of devices and larger density of devices in a particular geographical area. Because of these differences, supporting M2M in cellular networks without affecting or least affecting H2H communications is a very challenging problem.

Due to presence of enormous number of M2M devices in a particular geographical area and limited bandwidth resources, existing resource scheduling algorithms do not perform well. There is essential need of scheduling algorithms which not only support M2M applications but also try to keep QoS of H2H applications unaffected. In this paper, we propose a Class Based Priority (CBP) scheduling algorithm for uplink (i.e., device base station) communication in LTE systems in which, based on the QoS requirements, applications are kept into different priority classes. Scheduling of radio resources is done in the order of high priority class to low priority class applications. Rest of the paper is organized as follows: In Sections II, III related work and motivation of this work are discussed, respectively. In Section IV, we have mathematically formulated the resource allocation problem. In Section V, scheduling algorithm for resource allocation is presented. Simulation setup and performance evaluation of the algorithm are given in Section VI and finally, Section VII concludes the paper.

II. Related Work

In this section, we review existing works addressing scheduling issues due to incorporation of M2M in LTE systems. In [6], four classes are defined into which all applications are categorized. An utility function is associated with each class where user utility is a function of achievable data rate. The main aim of this approach is to maximize the aggregate throughput by maximizing the aggregate utility. But, in this approach fairness of a device is ignored. As a result of this, if a device has delay intolerant data but scheduling this device does not increase aggregate throughput (if channel quality of a device is bad) then this device may not be scheduled. Similarly, device having weak signal strength may also be not scheduled.

In [7], two scheduling algorithms were proposed for allocating resources between H2H and M2M flows in LTE. Both algorithms give first priority to H2H flows. After the allocation of radio Resource Blocks (RBs) to the H2H flows, the remaining RBs are allocated to M2M flows. The first algorithm gives higher priority to the 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 flows based on the applications they belong to. It does not differentiate the delay tolerant and delay intolerant M2M flows and therefore efficient allocation of RBs in not done.

In [8], authors proposed the concept of clustering of M2M devices. The parameters used to assign a cluster to an M2M device is packet arrival rate and maximum tolerable jitter. It means that if both cluster and device have identical values of above parameters then device will belong to that particular cluster. A cluster will be given higher priority if its packet arrival rate is more. If priority of a cluster is high then it will get preference during allocation of resources. But this approach ignores other QoS characteristics such as delay requirement and reliability.

In this paper, we have done the classification of all M2M and H2H applications based on various parameters and assigned a priority to each class. Then, we have proposed a scheduling algorithm which schedules the resources (*i.e.,* RBs) based on the priority of class.

III. Motivation

Different M2M applications have different QoS requirements in nature. Some applications are delay tolerant like environment monitoring applications while some are delay-intolerant like emergency alerting. Similarly, some other QoS parameters are also there, based on which classification of M2M applications can be done. In [4], the authors broadly classified all the M2M applications into eight classes. Classification was done based on the QoS parameters *viz,* priority of data, its reliability requirements and its real time nature (delay tolerance level). Authors also proposed that all H2H/H2M applications too fall under these eight classes. In Table I, value of 1 represents that the parameter is required/important and value of 0 represents that parameter is not required by the application of the class. In this work, based on the classification shown in Table I, we assign priorities to classes in the decreasing order from class 1 to class 8. We have assumed that an M2M device can have only one application running in it but H2H/H2M devices can have multiple applications running in them. A device belongs to a class if the application running in it belongs to that class. Hence, a H2H device may belongs to multiple classes at the same time because a H2H device may have more than one application running in it. In the proposed CBP scheduling algorithm for LTE systems, resource allocation is done based on the priority of the class to which application running on M2M or H2H/H2M device belongs to. The motivation behind class based scheduling algorithm is two fold: (a) to give preference to H2H devices/applications, belonging to high priority class, over others. (b) to give preference to M2M devices/applications, belonging to high priority classes, over H2H devices/applications belonging to low priority classes. Since, the current cellular networks like LTE are optimized for H2H/H2M, above point (a) becomes relevant here. An M2M flow belonging to low priority class may have compromised QoS, so by giving preference to higher class M2M flows over lower class H2H/H2M flows we can efficiently use the available RBs to support M2M. In this case, point (b) becomes relevant. But, if number of M2M devices is quite large in comparison to H2H/H2M in a geographical area then the performance of H2H may degrade. Hence, to ensure good performance of H2H/H2M, we have kept a threshold on maximum number of RBs to be assigned to M2M in a particular scheduling interval. In our experiments, given in section VI, we have analyzed the performance of H2H by varying this threshold value. Apart from this, in the proposed CBP algorithm, fairness of devices in terms of their CQI (channel quality information) value is also considered. Hence, a device with lower CQI value but belongs to higher class will also get RBs allocated. Figure 1 shows a class based LTE network scenario where both M2M and H2H devices co-exist in the network and get RBs allocated by the scheduler running at eNodeB. Each class contains both H2H and M2M flows. Classes are shown in decreasing order of priority in clockwise direction. In a class, dotted arrow shows the high priority of H2H over M2M.

TABLE I
Classification of M2M and H2H/H2M applications

	Class 1	Class 2	Class 3	Class 4	Class 5	Class 6	Class 7	Class 8
Priority	1	0	1	0	1	0	1	0
Reliability	1	1	0	0	1	1	0	0
Real Time	1	1	1	1	0	0	0	0

IV. Problem Formulation

In this section, we have formulated the problem of RB allocation in terms of total utilities of classes in a scheduling interval aka TTI (Transmission Time Interval). Utility of a class is defined as the total number of M2M and H2H requests satisfied in that class in a TTI. A H2H/M2M request is a request generated by a H2H/M2M device/application to get RBs from eNodeB. A H2H or M2M request is satisfied if it gets at least some minimum number of RBs allocated. This amount can be calculated by multiplying the minimum guaranteed bit rate (MGBR) of the traffic of the application (who has generated the request) with the value of TTI. For example, in order to support applications like video streaming and VOIP, a minimum bit rate must be guaranteed. MGBR of an M2M application traffic and of a H2H application traffic belonging to the same class might be different.

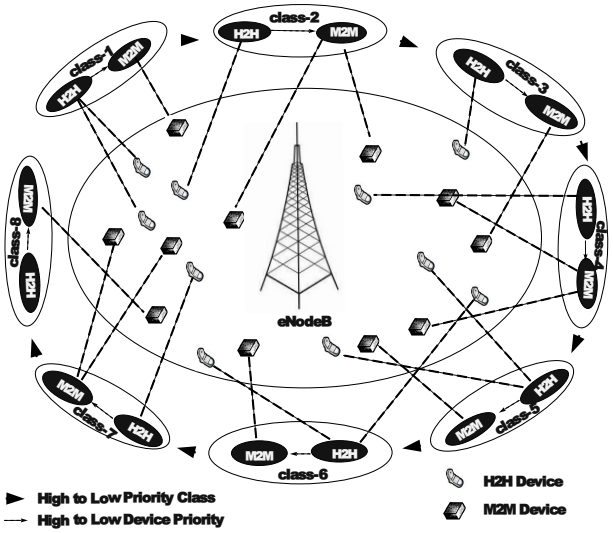

Fig. 1. Priority class based scheduling of M2M and H2H flows

Utility of a class C_n is represented by the following equation:

$$C_n = \sum S(H) + \beta_n \sum S(M) \tag{1}$$

where S(H) denotes the satisfiability function of a H2H request and S(M) denotes the satisfiability function of an M2M request.

$$S(H) = \begin{cases} 1 & \text{if } R_H \geq MGBR * TTI \\ 0 & \text{Otherwise} \end{cases} \tag{2}$$

where R_H is number of RBs allocated to a H2H request in a TTI. Similarly,

$$S(M) = \begin{cases} 1 & \text{if } R_M \geq MGBR * TTI \\ 0 & \text{Otherwise} \end{cases} \tag{3}$$

where R_M is number of RBs allocated to a M2M request in a TTI. β_n ensures that M2M request will be allocated RBs only after allocating all H2H request in class n. Therefore,

$$\beta_n = \begin{cases} 1 & \text{if } H_{Tn} = H_{Sn} \\ 0 & \text{Otherwise} \end{cases} \tag{4}$$

where H_{Tn} is the total number of H2H requests belong to class n which requested for RBs in a TTI and H_{Sn} is the total number of H2H requests served in class n. Now, total utilities of all the classes (1 to N) in a TTI can be written as follows:

$$(C_1 + \alpha_1(C_2 + \alpha_2(C_3 + \cdots + \alpha_{N-1}(C_N)))) \tag{5}$$

where α_n ensures that class $n+1$ will be served only after class n has been served satisfactorily. Constraint on number of RBs to be allocated to all M2M requests in a TTI is defined as follows:

$$\sum_{i=1}^{N} RB_{mi} \leq L_m \tag{6}$$

where RB_{mi} is the total number of RBs assigned to M2M requests in a class i in a TTI and L_m is the threshold on maximum number of RBs to be assigned to M2M requests in a TTI. If some of L_m RBs are still left unassigned after allocation to all of the current M2M requests in a TTI, then they can be allocated to H2H/H2M by CBP scheduler.
Constraint on α_n can be defined as follows:

$$\alpha_n = \begin{cases} 1 & \text{if } D_{Tn} = D_{Sn} \text{ or } (\sum_{i=1}^{N} RB_{mi} = L_m \text{ and } \beta_n = 1) \\ 0 & \text{Otherwise} \end{cases} \tag{7}$$

where D_{Tn} is the sum of number of H2H and M2M requests belong to class n and D_{Sn} is the sum of number of H2H and M2M requests served in class n. So, the value of α_n will be 1 if, either all M2M and H2H requests of class n have been satisfied or all H2H requests of class n have been satisfied *i.e.,* $\beta_n = 1$ in case of constraint on maximum number of RBs to be allocated to M2M requests is reached.

V. Proposed CBP Scheduling Algorithm

Since, in M2M communication uplink traffic is more, we have designed the algorithm for uplink scheduling. Algorithm 1 lists out the proposed CBP scheduling algorithm. The algorithm allocates RBs to different M2M and H2H requests arriving in a TTI. In the algorithm, we have assigned a priority to each request based on the class it belongs to, whether it is generated by H2H or M2M device and CQI value of the device who generated it. So, a request will have highest priority if it belongs to class 1, it is generated by a H2H device and CQI value of this H2H device is highest among all H2H devices in class 1. In the algorithm, we have taken min-heap as the data structure to implement the priority queue of incoming requests. The request with highest priority (lowest priority number) is the root of the heap. In first iteration, if a request gets opportunity to be served, the scheduler allocates only $MGBR * TTI$ resources to it. If all the requests get served but unassigned RBs (U_{RB}) are still available then algorithm

enters into second iteration. In this iteration, algorithm will again create a min-heap of same requests. But in this case, all M2M requests will also be treated as H2H requests. i.e., there will not be any limitation on number of RBs to be allocated to M2M devices because in first iteration itself, all requests have received their $MGBR * TTI$ amount of RBs.

Algorithm 1 CBP Uplink Scheduling Algorithm

Input: Set S of all requests for RBs that came in a TTI
Output: Allocation of RBs to requests

U_{RB} = Number of unallocated RBs
L_m = Threshold on maximum number of RBs available for all M2M requests in a TTI
Assign priority to all requests
$i = 0$ { to keep track of iterations}
while $U_{RB} \neq 0$ **do**
  Build a *min-heap* of requests with highest priority request at root
  while Heap is not empty **do**
    R =Extract Min
    $R_{RB} = MGBR * TTI$
    if $R_{RB} > U_{RB}$ **then**
      $U_{RB} = 0$
      break; {Come out of inner while loop}
    else
      if $i > 0$ **then**
        Allocate R_{RB} RBs to request R
        $U_{RB} = U_{RB} - R_{RB}$
      else
        if R is a H2H request **then**
          Allocate R_{RB} RBs to request R
          $U_{RB} = U_{RB} - R_{RB}$
        else
          if $R_{RB} > L_m$ **then**
            $L_m = 0$
            Continue; {Go to begin of the inner while loop}
          else
            Allocate R_{RB} RBs to request R
            $U_{RB} = U_{RB} - R_{RB}$
            $L_m = L_m - R_{RB}$
          end if
        end if
      end if
    end if
  end while
  $i = i + 1$ {going to next iteration}
end while

VI. Simulation Results and Performance Evaluation

In this section, performance of proposed CBP scheduling algorithm is evaluated using system level simulations in NS-3 simulator [9]. Simulation parameters are specified in Table II and parameters not specified here are assumed to be default ones mentioned in 3GPP specifications.

TABLE II
Simulation parameters

Simulator	NS-3.14.1
Cellular Layout	Single-Cell with Omni-directional Antenna
No. of RBs	50
No. of Devices	9, 12, 18, 24, 30, 36, 42, 48, 90
Application Traffic	UDP, TCP
Inter Packet Interval	50ms
Packet Size	400 Bytes (M2M) 1024 Bytes (H2H)

Performance evaluation is done within a single-cell environment with an omnidirectional antenna so that no inter-cell interference is present. The operating bandwidth on the uplink is 20 MHz, sub-divided into 100 RBs with each RB spanning a bandwidth of 180 kHz. The simulator assumes that each device (M2M/H2H) sends CQI spanning the entire bandwidth periodically, hence the eNodeB is assumed to have a full knowledge of the channel conditions per device for every TTI. All devices are located around eNodeB with a distance ranging from 10 meters to 1000 meters using constant position mobility model, meaning they are statically fixed. Performance of H2H devices degrade with the introduction of M2M devices, due to lack of enough RBs to meet QoS of H2H devices. This adverse effect can be controlled by introducing a threshold on number of RBs allocated to M2M requests, as proposed in the algorithm. This threshold limit is determined by using parameter λ. Therefore, maximum number of RBs that can be allocated for M2M requests will be $L_m = \lambda$ * total number of RBs. It means that total M2M requests will get resources from 0 to L_m. The performance of H2H and M2M devices with various values of λ is shown in Figure 2. In this experiment, network scenario is as follows: 90 devices are attached to eNodeB out of which 30 devices are H2H each running two applications. Remaining 60 devices are M2M each running single application. From Figure 2, upto λ= 0.5, throughput of

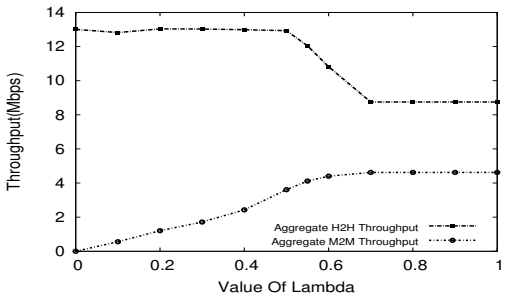

Fig. 2. Effect of varing λ on throughputs of H2H and M2M.

M2M devices is increasing while for H2H devices, it is almost unaffected. But, when value of λ goes beyond 0.5, throughput of H2H devices starts decreasing fastly but throughput of M2M devices increases slightly. When value of λ reaches to 0.7,

throughputs of both M2M and H2H devices start converging to some constant values. The main reason behind this behavior is that when less number of RBs are assigned to M2M requests, H2H requests/devices are unaffected but when number of RBs assigned to M2M requests are more, performance of H2H requests/devices degrade. For better performance, value of λ can be chosen between 0.4 and 0.5. Figures 3, 4, 5 show the performance comparison between CBP scheduler and Round-Robin (RR) scheduler in terms of system throughput. Network scenario in these cases are as follows: if N devices are attached to eNodeB, (N/3) devices are H2H each with two applications running. Remaining ($2N$/3) devices are M2M each with a single application running. Values of N considered in the experiments are 9, 12, 18, 24, 30, 36, 42, and 48. Here, we have taken the value of λ as 0.45.

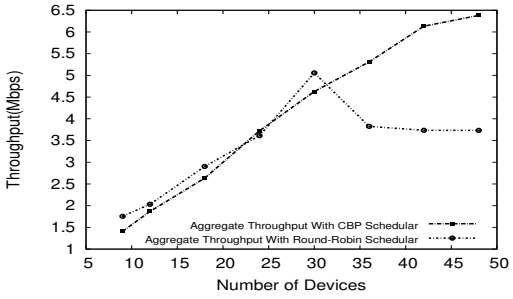

Fig. 3. Comparison of system throughput of RR and CBP schedulers for UDP Traffic

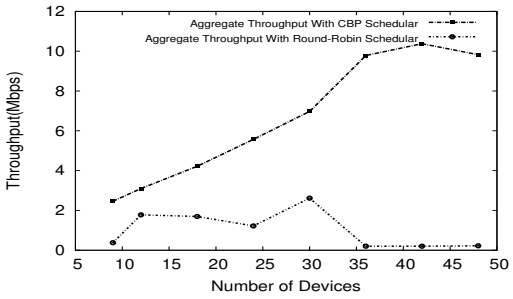

Fig. 4. Comparison of system throughput of RR and CBP schedulers for TCP Traffic

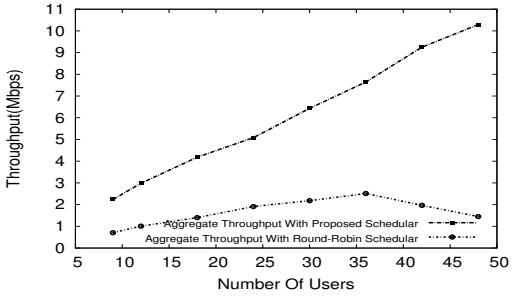

Fig. 5. Comparison of system throughput of RR and CBP scheduler while TCP-UDP Traffic

The scenario uses UDP traffic in Figure 3, uses TCP traffic in Figure 4 and uses mixture of TCP and UDP in Figure 5. The reason behind poor performance of RR can be explained as follows: (i) CBP scheduler puts an upper limit on number of RBs allocated to M2M unlike RR where more RBs are allocated to them. Since, most of the M2M applications produce small-sized data payloads, the allocated RBs of M2M are incompletely filled leading to bandwidth wastage. (ii) CBP scheduler allocates RBs based on application's satisfactory needs (Minimum Guaranteed Bit Rate * TTI) unlike RR, where RBs are allocated uniformly without the concern of application needs. When number of running applications are less, as shown in Figure 3, performance of RR is similar to that of CBP scheduler but for large number of applications it degrades but in Figures 4 and 5, irrespective of number of running applications, performance of RR always degrades. The reason behind such performance of RR is that when application traffic is TCP, application maintains a TCP congestion window. Size of congestion window will exponentially increase till a certain threshold if the application receives ACKs continuously, otherwise it will reduce to half. RR allocates resources uniformly even when the number of applications are more. Because of this, congestion window will fluctuate continuously which degrades throughput whereas CBP scheduler allocates RBs based on satisfactory needs. So, the congestion window will increase continuously.

As shown in Figures. 6, 7, we examined the average utility of different priority classes in two different scenarios listed in Table III. For a class, utility is defined as the total number of applications (H2H/M2M) in that class whose aggregate received bytes are greater than or equal to satisfaction limit of the class (MGBR $\times$ simulation time). Figures 8, 9, compare the average throughputs of each class in these two scenarios. As RR treats all classes equally, the average utility and average throughput are same for all classes. Both metrics show that the proposed scheduler performs better than RR with respect to high priority classes. Clearly, this is because the proposed scheduler works on the same lines of satisfaction limit (MGBR $\times$ TTI) and priority of classes unlike RR which is priority insensitive. Network scenario for these experiments is as follows: 50 devices are attached to eNodeB which are running applications of five priority classes (1-5) in the given proportion in Table III. Satisfaction limits for scenarios 1 and 2 are taken as 63.12 Kbps and 36.82 Kbps, respectively.

TABLE III
DISTRIBUTION OF CLASS

Priority Class	Priority	Scenario 1 (% of UEs)	Scenario 2 (% of UEs)
Class 1	1	20%	40%
Class 2	2	20%	30%
Class 3	3	20%	10%
Class 4	4	20%	10%
Class 5	5	20%	10%

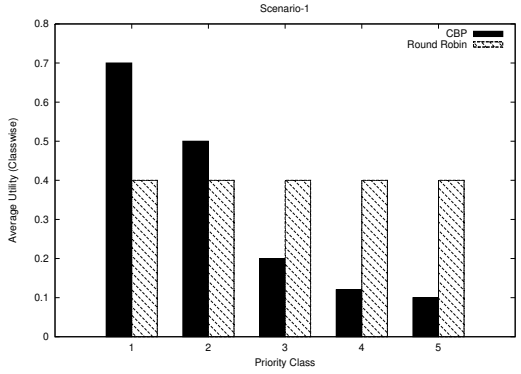

Fig. 6. Comparison of Average Utility of RR and CBP schedulers.

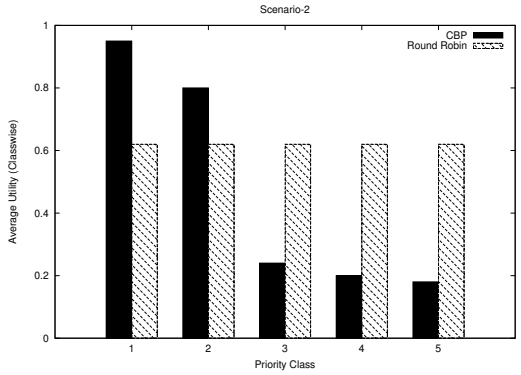

Fig. 7. Comparison of Average Utility of RR and CBP schedulers.

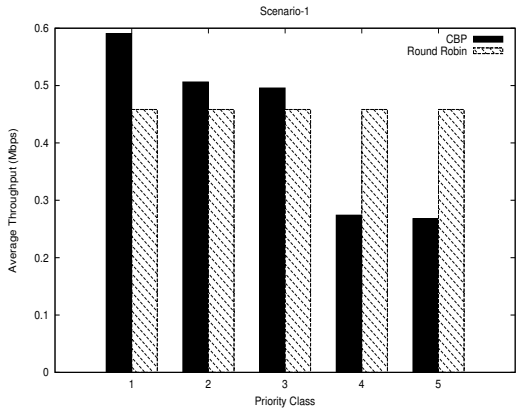

Fig. 8. Comparison of Average Throughput of RR and CBP schedulers.

VII. Conclusions

M2M communications is an emerging technology with more and more M2M services being deployed in the market. Enormous number of M2M devices brings great traffic pressure to the cellular network systems like LTE. Introduction of M2M communication effects the existing H2H/H2M communications as well. If proper radio resource allocation schemes are not followed, this effect could drastically reduce the performance of H2H/H2M communications. Proposed class based scheduling algorithm ensures uninterrupted H2H/H2M communications but at the same time, it facilitates M2M communications also in LTE systems. Class based scheduling of the resources gives importance to the priority of traffic and guarantees the user experience.

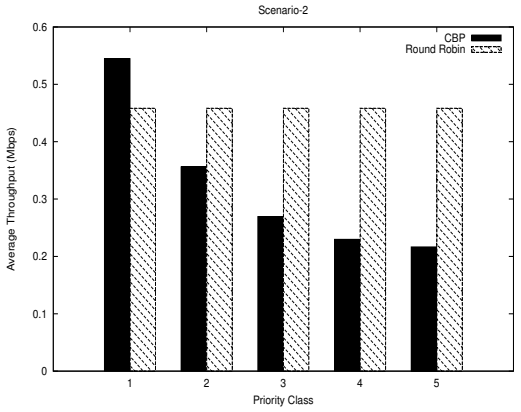

Fig. 9. Comparison of Average Throughput of RR and CBP schedulers.

We have implemented the proposed algorithm and evaluated the performance of the scheduler using various metrics. We found that value of λ can be taken in between 0.4 and 0.5 so that M2M users can be supported without interrupting H2H users. We also found that in case of TCP traffic, RR scheduler shows very weak performance while CBP scheduler shows good performance in both TCP and UDP traffic. We estimated the average per class utility and average per class throughput and compared with RR.

Acknowledgments

This work was supported by the Deity, Govt of India (Grant No. 13(6)/2010CC&BT).

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] V. Galeti, I. Boji, M. Kuek, G. Jei, S. Dei and D. Huljeni , "Basic principles of Machine-to-Machine communication and its impact on telecommunications industry", *in Proc. of IEEE MIPRO*, May 2011.

[3] M. Zubair Shafiq, Lusheng Ji, Alex X. Liu, Jeffrey Pang and Jia Wang, "A First Look at Cellular Machine-to-Machine Traffic Large Scale Measurement and Characterization", *ACM SIGMETRICS*, vol. 40, no. 1, pp. 65-76, June 2012.

[4] Rongduo Liu, Wei Wu, Hao Zhu and Dacheng Yang, "M2M-Oriented QoS Categorization in Cellular Network", *in Proc. of WiCOM 2011*, September 2011.

[5] 3GPP TS 36.211 V8.7.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical Channels and Modulation (Release 8)", June 2009.

[6] 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.

[7] 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.

[8] 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.

[9] Network Simulator. Available online at:*"http://www.nsnam.org"*.