A Dynamic Rate Adaptation Scheme for M2M Communications

Yalong Wu*, Wei Yu*, David Griffith†, and Nada Golmie†
*Towson University, MD, USA. Emails: ywu11@students.towson.edu, wyu@towson.edu
†National Institute of Standards and Technology. Emails:{david.griffith,nada.golmie}@nist.gov

***Abstract*—The number of Machine-to-Machine (M2M) devices has continued to grow at an accelerated rate. Without thoughtful and efficient resource management, M2M communications will be asymmetrically handicapped by service rate scarcity as more devices are continually added. To address these issues, in this paper, we propose a dynamic rate adaptation (DRA) scheme to obtain an optimized service rate distribution among a mixture of time-driven and event-driven M2M applications. DRA introduces real time monitoring of M2M traffic arrival rate, building on which service rate distribution between M2M applications can be adjusted momentarily, by using the mean value theorem of integrals (MVTI) and generalized processor sharing (GPS). We have validated the effectiveness of our proposed DRA scheme and our experimental results demonstrate that DRA can significantly improve M2M communications performance with respect to throughput and delay.**

***Keywords*—Internet of Things, M2M communications, Dynamic rate adaptation, Resource allocation**

I. Introduction

Machine-to-Machine (M2M) communication, also referred to as Machine Type Communication (MTC), enables ubiquitous connectivity between massively deployed devices (sensors, meters, etc.) with little or no human intervention [1]. With the development of the 3rd Generation Partnership Project (3GPP) Long-Term Evolution (LTE) and Long-Term Evolution Advanced (LTE-A) [6], M2M communication is the backbone of the Internet of Things (IoT) communication, supporting a diverse array of applications, including e-Health, smart grid, smart environment, intelligent transportation, public safety, etc. [9], [16], [17], [8].

Nonetheless, network infrastructures in general are facing an unprecedented challenge in exploding number of M2M devices. Thus, resource allocation is a critical issue to be solved. Our proposed scheme in this paper, designated DRA, seeks to improve M2M communications performance, based on a realistic and effective service rate adaptation for diverse M2M applications. Service rate adaptation for applications with diverse weights is a critical issue in M2M communications, especially as resources become strained. In the interest of developing an appropriate solution under the restriction of constrained network resources, a number of research efforts have been conducted. The results of these investigations show that more efficient service rate adaptation among applications can be achieved, especially with the technology advancement on communications and sensors [7], [12].

Nonetheless, the majority of these efforts do not take into account the uniqueness of M2M traffic arrival rate, which varies significantly moment to moment. Instead, they rely upon selected constant weights associated with distinct applications, resulting in fixed allocation of network resources to enable a stable service rate distribution among different applications. In reality, some applications might suffer from a service rate starvation while others remain a service rate surplus, due to the different traffic arrival times and the ever-changing traffic arrival rates of M2M applications. Thus, it is urgent to develop some techniques, which consider both traffic arrival time and traffic arrival rate variation, such that the service rate sharing between different M2M applications can be largely improved.

In this paper we propose a dynamic rate adaptation (DRA) scheme for M2M communications, which can effectively enhance the overall network performance with regard to throughput and delay. Dynamic rate adaptation aims to more efficiently allocating constrained network service rate to diverse M2M applications. In this scheme, DRA constantly monitors the arrival traffic of different M2M applications and distinguishes applications which are in active mode from those that are passive. Based on MVTI (the mean value theorem of integrals), DRA then determines a constant traffic arrival rate, for only the active M2M applications. The constant rate can generate the same amount of traffic data as real M2M arrival traffic, which follows *Beta distribution* [14]. Finally, by applying GPS (generalized processor sharing) to the generated constant traffic arrival rates, DRA can efficiently and fairly distribute service rate between multiple M2M applications, considering not only the application weights, but also different traffic arrival times and ever-changing traffic arrival rates. With the introduction of real-time monitoring and service rate adjusting, dynamic rate adaptation can significantly achieve higher throughput and lower delay.

Through a combination of detailed theoretical analysis and extensive simulation, we assess the effectiveness of our proposed DRA scheme against the conventional weight-based scheme in terms of throughput and delay. To be specific, we set up three resource scarcity scenarios, denoted as mild, moderate, and severe resource scarcity. The mild resource scarcity scenario suffers a service rate shortage that is $1/10$ of the total for all M2M applications to be fully satisfied, while moderate and severe resource scarcities with $1/50$ and $1/100$ of the total, respectively. Our experimental results show that DRA provides superior network performance over the conventional weight-based scheme for every scarcity scenario.

The remainder of this paper is organized as follows: In

978-1-5386-3180-5/18/$31.00 ©2018 IEEE

Section II, we present our approach in detail. In Section III, we conduct the effectiveness analysis of our approach. In Section IV, we show experimental results to validate the effectiveness of our scheme. Finally, we conclude the paper in Section V.

II. Our Approach

In this section, we first present the basic concept of our approach, and then introduce the key components.

A. Basic Idea

In DRA, the typical M2M traffic characteristics of diverse M2M applications is estimated, such that a more efficient and fair service rate allocation can be achieved. To be specific, DRA proceeds to carry out service rate distribution between M2M applications with an additional consideration of the unique characteristics of M2M communications, whose traffic arrives at a very different regularity for multiple M2M applications. Within a given time duration, there might be several active M2M applications generating traffic.

Based on the huge variation of instant traffic arrival rates of different active M2M applications, DRA first cuts the duration into small identical time slices. Then, in each time slice, DRA works out a corresponding constant traffic arrival rate for each active application on the basis of the mean value theorem of integrals (MVTI). At last, a fair service rate is assigned to each M2M application through applying generalized processor sharing (GPS) to those obtained constant traffic arrival rates.

DRA considers two metrics (i.e., throughput and delay) to evaluate the effectiveness of our approach. With the emphasis on specific M2M communications traffic characteristics and M2M communications traffic arrival regularities, DRA seeks to realize more efficient service rate distribution across varying M2M applications.

Our proposed approach consists of two key components: (i) *M2M traffic modeling* highlights the real time traffic characteristics for both time-driven and event-driven M2M applications, (ii) *Service rate distribution* partitions service rate for distinct M2M applications with the adoption of MVTI and GPS. In the following, we describe these components in detail.

B. M2M Traffic Modeling

It has been showing that the traffic arrival rate of M2M communications follows the *Poisson distribution*, which can be modulated as *Beta distribution* when massive M2M devices generating data simultaneously within an undefined communication period [5], [4], [11]. In the context of M2M communications, M2M devices and servers are all initialized with a preset configuration to generate either time-driven or event-driven traffic sessions [15], [4]. Time-driven traffic sessions are triggered periodically with fixed time cycles, while event-driven traffic sessions are activated randomly with specific events.

In 2011, 3GPP proposed the *Beta distribution* to capture the unique characteristic of M2M traffic arrival pattern more precisely [13]. Within a finite time range $0 < t < \tau$ under parameters α, β $(\alpha > 0, \beta > 0)$, the *Probability Density Function* (PDF) $f(t)$ of the *Beta distribution* can be written as follows:

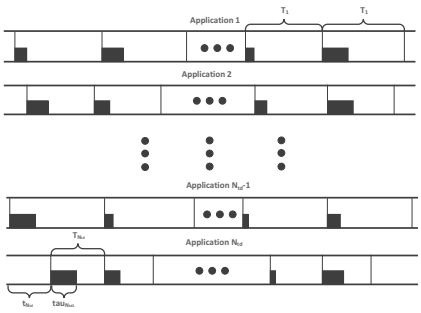

Fig. 1. Time-driven M2M Applications

$$f(t) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{t^{\alpha-1}(\tau-t)^{\beta-1}}{\tau^{\alpha+\beta-1}}, \tag{1}$$

where Γ is a *Gamma function*. As distinct M2M applications have different pairs of (α, β), the specification of those parameters can be finalized with the consideration of *Karl Pearson's estimation* based on fields test [4]. 3GPP recommended two integer pairs of (α, β), which are $(1, 1)$ and $(3, 4)$, respectively. With the assumption of α, β as integers, Jian *et al.* in [5] showed that it would be more appropriate to choose a reasonable gap between α and β with the constraint of $\alpha < \beta$ for different M2M applications. In this research, we assume α, β as integers for specific M2M applications.

Recall that DRA constantly monitors the real-time traffic of diverse M2M applications with the consideration of different traffic arrival times and the uniqueness of M2M traffic arrival characteristics. It then adjusts service rate distribution among multiple M2M applications based on MVTI and GPS, which is discussed in Section II-C. In the following, we first introduce the modeling of time-driven and event-driven M2M traffic, respectively. We then apply dynamic rate adaptation to efficiently allocate service rate to a mixture of time-driven and event-driven M2M applications.

1) Time-driven Modeling: Time-driven M2M applications generate traffic periodically with unchanging predefined time cycles. Particularly, there are always multiple traffic sessions of diverse time length τ, and a particular traffic inter-arrival time length T for each time-driven M2M application, where τ is typically much smaller than T, as shown in Fig. 1. If we assume that there are N_{td} time-driven applications, and suppose the traffic session parameters for time-driven application App_u^{td} $(u = 1, 2, \ldots, N_{td})$ in κ^{th} time cycle as α_u, β_u and τ_{u_κ}, the traffic starting time as t_u, and the inter-arrival time length as T_u. Based on Equation (1), we can express the traffic arrival rate $\lambda^a_{u_\kappa,td}$ of App_u^{td} in κ^{th} time cycle as follows:

$$\lambda^a_{u_\kappa,td} = \begin{cases} \dfrac{\Gamma(\alpha_u+\beta_u)}{\Gamma(\alpha_u)\Gamma(\beta_u)} \dfrac{[t-(t_u+\kappa T_u)]^{\alpha_u-1}(\tau_{u_\kappa}+t_u+\kappa T_u-t)^{\beta_u-1}}{\tau_{u_\kappa}^{\alpha_u+\beta_u-1}} \\ \qquad\qquad t_u+\kappa T_u \le t \le \tau_{u_\kappa}+t_u+\kappa T_u, \\ 0 \qquad\qquad otherwise, \end{cases} \tag{2}$$

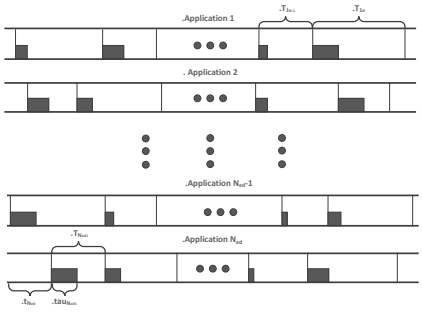

Fig. 2. Event-driven M2M Applications

where $\kappa = 0, 1, 2, \ldots, N$ and N is a large positive integer. As implied by Equation (2), the traffic arrival rate of each time-driven M2M application is a periodic function with a certain time cycle starting from some point in time. Thus, the generated traffic flows in all time cycles fit this description, making M2M communications less random. Regarding event-driven M2M applications, despite the likelihood of traffic flows in all time cycles, there is also something different with the length of each traffic inter arrival time, which will be discussed in the following.

2) Event-driven Modeling: Unlike time-driven M2M applications, which are activated at predefined cyclical time points, event-driven M2M applications are triggered randomly by specific events (earthquakes, hurricanes, fires, etc.), as shown in Fig. 2. Consequently, the inter-arrival time length of each event-driven M2M application varies significantly. According to Section II-B1, we assume the number of event-driven M2M applications as N_{ed}. Henceforth, we refer to the traffic session parameters of event-driven M2M application App_v^{ed} $(v = 1, 2, \ldots, N_{ed})$ in ζ^{th} time cycle as $\hat{\alpha}_v, \hat{\beta}_v$ and $\hat{\tau}_{v_\zeta}$, its traffic starting time as $\hat{t}_v$, and its ζ^{th} inter-arrival time length as T_{v_ζ}. With the same procedure as the derivation of Equation (2) on the basis of Equation (1), we have the traffic arrival rate $\lambda^a_{v_\zeta,ed}$ of App_v^{ed} in ζ^{th} time cycle as follows:

$$\lambda^a_{v_\zeta,ed} = \begin{cases} \hat{\Gamma}\dfrac{[t-(\hat{t}_v+\sum_{p=0}^{\zeta}T_{v_p})]^{\hat{\alpha}_v-1}(\hat{\tau}_{v_\zeta}+\hat{t}_v+\sum_{p=0}^{\zeta}T_{v_p}-t)^{\hat{\beta}_v-1}}{\hat{\tau}_{v_\zeta}^{\hat{\alpha}_v+\hat{\beta}_v-1}} \\ \qquad\qquad \hat{t}_v+\sum_{p=0}^{\zeta}T_{v_p} \le t \le \hat{\tau}_{v_\zeta}+\hat{t}_v+\sum_{p=0}^{\zeta}T_{v_p}, \\ 0 \qquad\qquad otherwise, \end{cases} \tag{3}$$

where $\zeta = 0, 1, 2, \ldots, N$ (N is also a large positive integer) and $\hat{\Gamma} = \frac{\Gamma(\hat{\alpha}_v+\hat{\beta}_v)}{\Gamma(\hat{\alpha}_v)\Gamma(\hat{\beta}_v)}$. With the traffic characteristic formulations of time-driven and event-driven M2M applications, we discuss efficient service rate distribution based on those characteristics in the next section.

C. Service Rate Distribution

Through the analysis of traffic arrival rates for time-driven and event-driven M2M applications presented in Sections II-B1 and II-B2, there is always a time range (much longer than traffic session length), in which an application transitions into sleep mode during each time cycle. It would not make any sense if some M2M applications are struggling with service rate acquisition while others still hold idle service rate while in their sleep mode. To this end, DRA adopts *Service Rate Distribution* to constantly monitor arrival traffics of M2M applications, and recognize those applications in sleep mode, redistributing service rates away from them. Our service rate distribution mechanism consists of the following components: rate equalization, time division, and rate relocation, which will be detailed next.

1) Rate Equalization: As implied by Equation (1), the traffic arrival rate of M2M applications changes over time, making it impossible for rate relocation in Section II-C3 to adopt GPS, which requires a constant rate from each session [10]. To tackle this issue, we have devised a rate equalization strategy to determine an equally-beneficial constant traffic arrival rate for each M2M application, based on MVTI. It is known that the integrals of traffic arrival rate over time is the overall traffic data. With the adoption of MVTI, the corresponding constant rate, derived from real M2M traffic that follows the *Beta distribution*, can generate the same amount of traffic data within the same time duration.

Recall that the traffic flows of time-driven and event-driven M2M applications arrive periodically with a consistent shape in a given time duration, making it reasonable to utilize rate equalization to normalize time cycles into a duration beginning at time 0 to evaluate associated constant rates. Suppose that the corresponding constant rates of App_u^{td} and App_v^{ed} in κ^{th} and ζ^{th} time cycles are $\lambda^{a,c}_{u_\kappa,td}$ and $\lambda^{a,c}_{v_\zeta,ed}$, respectively. Then, we have

$$\begin{aligned} \lambda^{a,c}_{u_\kappa,td} &= \frac{1}{\tau_{u_\kappa}^{\alpha_u+\beta_u}} \sum_{q=\alpha_u}^{\alpha_u+\beta_u-1} \frac{\tau_{u_\kappa}^q(\alpha_u+\beta_u-1)!}{q!(\alpha_u+\beta_u-1-q)!}, \\ \lambda^{a,c}_{v_\zeta,ed} &= \frac{1}{\hat{\tau}_{v_\zeta}^{\hat{\alpha}_v+\hat{\beta}_v}} \sum_{q=\hat{\alpha}_v}^{\hat{\alpha}_v+\hat{\beta}_v-1} \frac{\hat{\tau}_{v_\zeta}^q(\hat{\alpha}_v+\hat{\beta}_v-1)!}{q!(\hat{\alpha}_v+\hat{\beta}_v-1-q)!}. \end{aligned} \tag{4}$$

2) Time Division: To identify the time periods when M2M applications are idle and preempt the service rates they hold for activated M2M applications to use, service rate distribution introduces time division to slice a long time duration into pieces, such that rate relocation can constantly monitor traffic patterns of M2M applications, timely recognize their idle time periods, and redistribute service rate more effectively. Additionally, the shorter the time slices are, the more promptly the idle time periods can be identified. Nonetheless, this also increases the burden on M2M infrastructures. As a compromise between promptly identifying idle time periods and lowering the cost of relocating service rate within each time slice, time division utilizes the largest M2M application session length as the length of the time slice, which can be denoted as

$$L_{ts} = max(\max_{u,\kappa} \tau_{u_\kappa}, \max_{v,\zeta} \hat{\tau}_{v_\zeta}), \tag{5}$$

where τ_{u_κ} and $\hat{\tau}_{v_\zeta}$ are predicted based on empirical values. L_{ts} can not only accommodate all traffic sessions of M2M applications, but also have a reasonable length for M2M applications to get served after the rearrangement of service rate at the beginning of each time slice.

3) Rate Relocation: Based on the preparation of the equations in Sections II-C1 and II-C2, rate relocation employs GPS [10] to efficiently and fairly reassign service rate only to activated M2M applications at the beginning of each time slice, such that idle M2M applications do not occupy resources. As GPS merely allocates service rate to diverse applications within the limitation of system capacity, it is worth mentioning that the generated traffic of M2M applications might not get completely served during a single time slice, which requires the corresponding M2M applications to participate in more time slices until all the traffic is wholly absorbed. Recall that the length of a time slice specified in II-C2 is L_{ts}, if we assume that the number of time slices is N_{ts}, then the following equation can be derived:

$$N_{ts} = \frac{max(\max_{u}(t_u + NT_u), \max_{v}(\hat{t}_v + \sum_{p=1}^{N} \hat{T}_{v_p}))}{L_{ts}}, \tag{6}$$

where T_u and $\hat{T}_{v_p}$ are also derived on the basis of empirical values and the number of time cycles for all M2M applications is set up as N.

Assume that the system capacity of service rate is $\mathcal{C}$. If we denote the service rates allocated to App_u^{td} and App_v^{ed} in δ^{th} $(\delta = 1, 2, \ldots, N_{ts})$ time slice as $\lambda_{u_\kappa,td}^{s,\delta}$ and $\lambda_{v_\zeta,ed}^{s,\delta}$, respectively, we can derive the following equation by applying GPS only to the activated M2M applications at the beginning of each time slice:

$$\sum_{u=1}^{N_{td}} min(\lambda_{u_\kappa,td}^{a,c}, r_\delta w_u) + \sum_{v=1}^{N_{ed}} min(\lambda_{v_\zeta,ed}^{a,c}, r_\delta \hat{w}_v) = \mathcal{C}. \tag{7}$$

Here, r_δ is the fair rate in δ^{th} time slice, and w_u and $\hat{w}_v$ are the priority weights of time-driven and event-driven M2M applications, respectively. Equation (7) guarantees that the summation of service rates distributed to all activated M2M applications is exactly the same as the system capacity of service rate. With the produced r_δ, the corresponding service rates assigned to all activated time-driven and event-driven M2M applications can be written as follows:

$$\lambda_{u_\kappa,td}^{s,\delta} = min(\lambda_{u_\kappa,td}^{a,c}, r_\delta w_u), \quad \lambda_{v_\zeta,ed}^{s,\delta} = min(\lambda_{v_\zeta,ed}^{a,c}, r_\delta \hat{w}_v), \tag{8}$$

where u, v, κ, ζ in Equations (7) and (8) must be subject to the following two cases:

Case 1: The κ^{th} time-driven M2M application periodically starts generating traffic in δ^{th} time slice, and the ζ^{th} event-driven M2M application is triggered into alive in δ^{th} time slice.

$$(\delta - 1)L_{ts} \leq t_u + \kappa T_u, \quad \hat{t}_v + \sum_{p=0}^{\zeta} T_{v_p} < \delta L_{ts}, \tag{9}$$

Case 2: The κ^{th} time-driven M2M application periodically started generating traffic before δ^{th} time slice, and there is still leftover traffic data not being absorbed. The ζ^{th} event-driven M2M application was triggered ahead of δ^{th} time slice, and also there is still generated data in transmission.

$$\begin{cases} 0 < \xi < \delta \\ (\xi - 1)L_{ts} \leq t_u + \kappa T_u < \xi L_{ts} \\ \lambda_{u_\kappa,td}^{s,\xi}[\xi L_{ts} - t_u - \kappa T_u] + \sum_{g=\xi+1}^{\delta-1} \lambda_{u_\kappa,td}^{s,g} L_{ts} < \tau_{u_\kappa} \lambda_{u_\kappa,td}^{a,c}, \\ 0 < \gamma < \delta \\ (\gamma - 1)L_{ts} \leq \hat{t}_v + \sum_{p=0}^{\zeta} T_{v_p} < \gamma L_{ts} \\ \lambda_{v_\zeta,ed}^{s,\gamma}[\gamma L_{ts} - \hat{t}_v - \sum_{p=0}^{\zeta} T_{v_p}] + \sum_{g=\gamma+1}^{\delta-1} \lambda_{v_\zeta,ed}^{s,g} L_{ts} < \hat{\tau}_{v_\zeta} \lambda_{v_\zeta,ed}^{a,c}, \end{cases} \tag{10}$$

where $\xi = \lceil \frac{t_u + \kappa T_u}{L_{ts}} \rceil$, $\gamma = \lceil \frac{\hat{t}_v + \sum_{p=0}^{\zeta} T_{v_p}}{L_{ts}} \rceil$, respectively. The combination of Equations (9) and (10) pledges that only activated time-driven and event-driven M2M applications take part in sharing system capacity of service rate in each time slice. To be specific, Equation (9) singles out M2M applications whose traffic arrives in δ^{th} time slice, while Equation (10) recognizes M2M applications whose traffic arrives before δ^{th} time slice, but still not getting completely served (therefore need to be involved in δ^{th} time slice to further process untreated traffic data). Assume that the activation statuses of App_u^{td} and App_v^{ed} in δ^{th} time slice are $\Lambda_{u_\kappa,td}^{s,\delta}$ and $\Lambda_{v_\zeta,ed}^{s,\delta}$, respectively, then we have

$$\Lambda_{u_\kappa,td}^{s,\delta} = \Lambda_{v_\zeta,ed}^{s,\delta} = \begin{cases} 1 & \text{if (9) and (10) are true} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

If M2M applications transition into sleep mode in any time slice (violating with Equations (9) and (10)), they will not be involved in service rate competition, which can be referred to as

$$\lambda_{u_\kappa,td}^{s,\delta} = \lambda_{v_{\zeta,ed}}^{s,\delta} = 0. \tag{12}$$

Recall that DRA not only models real time M2M traffic to capture its unique characteristics, but also introduces service rate distribution, based on MVTI and GPS, to timely monitor M2M application traffic and constantly rearrange service rate allocation. With the collaboration of rate equalization, time division, and rate relocation, service rate distribution flexibly shares service rate only among activated time-driven and event-driven M2M applications, leading to more efficient and fair service rate allocation among M2M applications.

III. Analysis

We now analyze the effectiveness of our proposed scheme (DRA) with respect to average performance of diverse M2M applications. The metrics consist of: (i) *Throughput* is defined as the average of service rates assigned to each M2M application during its activated time slices, and (ii) *Delay* is referred

to as the average consuming time for the generated traffic of each M2M application in its time cycles to get completely absorbed.

A. Throughput Analysis

DRA is intended to effectively distribute service rate within the system capacity by constantly monitoring real time traffic, characterized by finite time slices. It distinguishes activated M2M applications from those that are in sleep mode, and partitions service rate only between those activated in each time slice. Based on the description of rate relocation in Section II-C3, the service rates allocated to time-driven M2M application App_u^{td} and event-driven M2M application App_v^{ed} in δ^{th} time slice are $\lambda_{u_\kappa,td}^{s,\delta}$ and $\lambda_{v_\varsigma,ed}^{s,\delta}$, respectively (i.e., conforming to Equations (8) and (12)). To properly assess the general performance of our proposed scheme in network throughput, we obtain the average service rate in N_{ts} time slices for each M2M application. Assume that the average throughput of App_u^{td} and App_v^{ed} by using DRA is Thr_u^{td} and Thr_v^{ed}, respectively, we then have

$$Thr_u^{td} = \frac{\sum_{\kappa=1}^{N}\sum_{\delta=1}^{N_{ts}} \lambda_{u_\kappa,td}^{s,\delta}}{\sum_{\kappa=1}^{N}\sum_{\delta=1}^{N_{ts}} \Lambda_{u_\kappa,td}^{s,\delta}}, \quad Thr_v^{ed} = \frac{\sum_{\zeta=1}^{N}\sum_{\delta=1}^{N_{ts}} \lambda_{v_\varsigma,ed}^{s,\delta}}{\sum_{\zeta=1}^{N}\sum_{\delta=1}^{N_{ts}} \Lambda_{v_\varsigma,ed}^{s,\delta}}, \quad (13)$$

where the larger Thr_u^{td} and Thr_v^{ed}, the better efficiency DRA has.

B. Delay Analysis

As long as the generated traffic of either time-driven or event driven M2M applications fall into a certain time slice, the corresponding M2M applications will participate in service rate sharing. During each time slice, DRA fairly distributes service rate among activated M2M applications on the basis of their traffic arrival rates and priority weights, within the system capacity. Thus, the service rate demanded from some applications might not be completely satisfied if too many M2M applications are activated simultaneously in that time slice, leading to some amount of service delay.

To simplify the formulation process, we figure out the generated traffic data for each M2M application during N_{ts} time slices and then use Thr_u^{td} or Thr_v^{ed} from Section III-A to divide it. If we represent the average finishing time of App_u^{td} and App_v^{ed} during N_{ts} time slices, with applying DRA, as Del_u^{td} and Del_v^{ed}, respectively, we will have

$$Del_u^{td} = \frac{\sum_{\kappa=1}^{N} \tau_{u_\kappa} \lambda_{u_\kappa,td}^{a,c}}{Thr_u^{td}}, \quad Del_v^{ed} = \frac{\sum_{\zeta=1}^{N} \hat{\tau}_{v_\varsigma} \lambda_{v_\varsigma,ed}^{a,c}}{Thr_v^{ed}}. \quad (14)$$

IV. Performance Evaluation

In this section, we present the simulation results of our proposed DRA scheme in terms of throughput and delay, which are discussed in Section III. Notice that DRA fully exploits the unique periodic property of M2M traffic, dynamically allocates service rate to M2M applications, which are in activation mode, and specifies service rate partition in each time slice based on MVTI and GPS. In conventional weight-based scheme, priority weight is the only key factor leveraged for allocating service rate of individual M2M applications [3]. We conduct our performance evaluation, in the context of crowded coexistence with fierce competition, among a number of M2M applications.

To ultimately fulfill this premise, the numbers of time-driven and event-driven M2M applications are both set at 1000. As 3GPP defines small data transmission feature for M2M communications, it is preferable to select low order parameters for M2M application arrival traffic, which follows *Beta distribution* [2], [4]. Thus, we randomly generate integers ranging from 1 to 4 as α, β, and τ (seconds) for all 2000 M2M applications. To emphasize the way less property of traffic session length under time cycle length, which is discussed in Section II-B1 and Section II-B2, we randomly generate integers between 36 and 48 in seconds for T_u and T_{v_ς} as the length of time cycles. As to each M2M application, the number of occurring time cycles N is set to 200, with the first session arrival time t randomly generated between 0 and 2000 seconds (10 times of the number of time cycles), in order to guarantee randomness. With respect to the priority weight w of each M2M application, it is randomly assigned from 1 to 3.

In our evaluation, the performance of DRA is estimated under scenarios of mild, moderate, and severe resource scarcity with respect to throughput and delay. All simulation in this paper is conducted in MATLAB[1].

Figs. 3, 4, and 5 demonstrate the average throughput of each M2M application under mild, moderate, and severe resource scarcities, respectively. From the figures, the performance of our proposed *Dynamic Rate Adaptation* (DRA) scheme performs better than the conventional weight-based scheme (denoted as normal in figures), the baseline for comparison, in every scenario. Though the improvement of DRA over the normal case decreases with increasing scarcity, DRA nonetheless maintains a much better throughput performance. Also, this implies that DRA can significantly improve the overall network performance in terms of throughput. For example, most M2M applications, running DRA under mild resource scarcity scenario in Fig. 3, maintain a throughput between 2 and 3 bytes/second, some even reaching beyond 4 or 5 bytes/second. In comparison, all M2M applications only have a throughput below 1 byte/second under the conventional weight-based scheme.

Figs. 6, 7 and 8 illustrate the comparison of DRA and the baseline weight-based scheme with respect to delay. As we can see from the figures, the average delay on each M2M application running DRA is much smaller than the delay on

[1]Certain commercial equipment, instruments, or materials are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement by the National Institute of Standards and Technology, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.

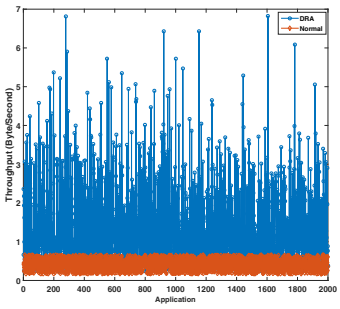

Fig. 3. Throughput with Mild Resource Scarcity

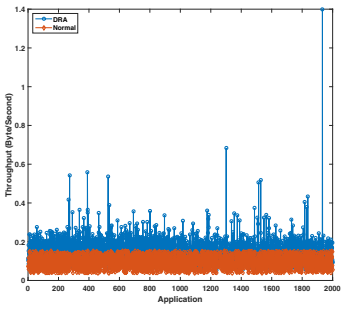

Fig. 4. Throughput with Moderate Resource Scarcity

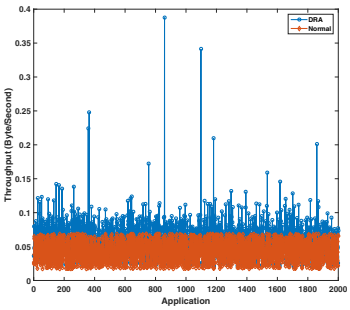

Fig. 5. Throughput with Severe Resource Scarcity

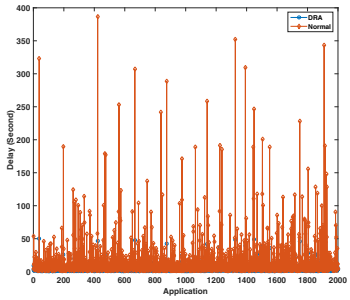

Fig. 6. Delay with Mild Resource Scarcity

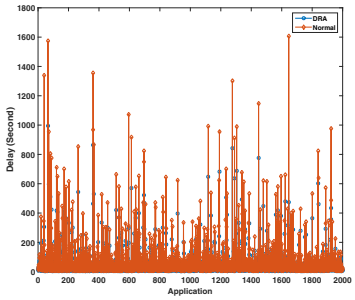

Fig. 7. Delay with Moderate Resource Scarcity

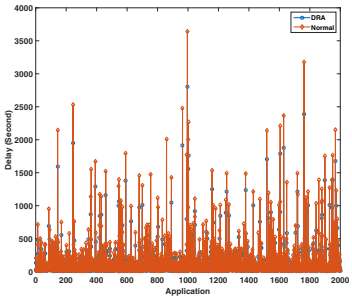

Fig. 8. Delay with Severe Resource Scarcity

M2M applications without DRA. Similar to throughput, the efficiency improvement over baseline decreases as resource scarcity increases, but is nevertheless, significant.

V. Conclusion

In this paper, we proposed a dynamic rate adaptation scheme to efficiently allocate service rate for diverse M2M applications. Particularly, DRA adopts MVTI to equalize the real M2M traffic arrival rate, which follows the *Beta distribution*, into a constant value. The constant rate is exploited by GPS to efficiently assign service rate for M2M applications. Our proposed scheme is competent in taking advantage of the unique traffic property of M2M communications to effectively distribute service rate between M2M applications. The experimental results demonstrate that DRA achieves a superior network performance with respect to throughput and delay, in comparison with the conventional weight-based scheme.

References

[1] D. Boswarthick, O. Elloumi, and O. Hersent. *M2M communications: a systems approach*. John Wiley & Sons, 2012.

[2] F. Ghavimi and H.-H. Chen. M2m communications in 3gpp LTE/LTE-A networks: architectures, service requirements, challenges, and applications. *IEEE Communications Surveys & Tutorials*, 17(2):525–549, 2015.

[3] J. Huang, H. Wang, Y. Qian, and C. Wang. Priority-based traffic scheduling and utility optimization for cognitive radio communication infrastructure-based smart grid. *IEEE Transactions on Smart Grid*, 4(1):78–86, 2013.

[4] X. Jian, X. Zeng, J. Huang, Y. Jia, and Y. Zhou. Statistical description and analysis of the concurrent data transmission from massive mtc devices. *International Journal of Smart Home*, 8(4):139–150, 2014.

[5] X. Jian, X. Zeng, Y. Jia, L. Zhang, and Y. He. Beta/m/1 model for machine type communication. *IEEE Communications Letters*, 17(3):584–587, 2013.

[6] A. Laya, L. Alonso, and J. Alonso-Zarate. Is the random access channel of LTE and LTE-A suitable for M2M communications? a survey of alternatives. *IEEE Communications Surveys & Tutorials*, 16(1):4–16, 2014.

[7] S.-Y. Lien, K.-C. Chen, and Y. Lin. Toward ubiquitous massive accesses in 3GPP machine-to-machine communications. *IEEE Communications Magazine*, 49(4):66–74, 2011.

[8] J. Lin, W. Yu, X. Yang, Q. Yang, X. Fu, and W. Zhao. A real-time en-route route guidance decision scheme for transportation-based cyberphysical systems. *IEEE Transactions on Vehicular Technology*, 66(3):2551–2566, March 2017.

[9] 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 of Things Journal*, 4(5):1125–1142, Oct 2017.

[10] A. K. Parekh and R. G. Gallagher. A generalized processor sharing approach to flow control in integrated services networks: The multiple node case. *IEEE/ACM Transactions on Networking (ToN)*, 2(2):137–150, 1994.

[11] M. Z. Shafiq, L. Ji, A. X. Liu, J. Pang, and J. Wang. A first look at cellular machine-to-machine traffic: large scale measurement and characterization. *ACM SIGMETRICS Performance Evaluation Review*, 40(1):65–76, 2012.

[12] T. Taleb and A. Kunz. Machine type communications in 3GPP networks: potential, challenges, and solutions. *IEEE Communications Magazine*, 50(3):178–184, 2012.

[13] G. T. . V11.0.0. Study on RAN improvements for machine-type communications. http://www.qtc.jp/3GPP/Specs/37868-b00.pdf, 2011 (accessed February 6, 2017).

[14] Wikipedia. Beta function. https://en.wikipedia.org/wiki/Beta_function, 2016 (accessed February 12, 2017).

[15] H. Wu, C. Zhu, R. J. La, X. Liu, and Y. Zhang. FASA: Accelerated S-ALOHA using access history for event-driven M2M communications. *IEEE/ACM Transactions on Networking (ToN)*, 21(6):1904–1917, 2013.

[16] G. Xu, W. Yu, D. Griffith, N. Golmie, and P. Moulema. Toward integrating distributed energy resources and storage devices in smart grid. *IEEE Internet of Things Journal*, 4(1):192–204, Feb 2017.

[17] R. Yao, W. Wang, M. Farrokh-Baroughi, H. Wang, and Y. Qian. Quality-driven energy-neutralized power and relay selection for smart grid wireless multimedia sensor based IoTs. *IEEE Sensors Journal*, 13(10):3637–3644, 2013.