This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

Optimizing M2M Communications and Quality of Services in the IoT for Sustainable Smart Cities

Jun Huang, *Senior Member, IEEE,* Cong-cong Xing, Sung Shin, Fen Hou, *Member, IEEE,* Ching-Hsien Hsu *Senior Member, IEEE,*

Abstract—Machine-to-machine (M2M) communications and applications are expected to be a significant part of the Internet of Things (IoT). However, conventional network gateways reported in the literature are unable to provide sustainable solutions to the challenges posted by the massive amounts of M2M communications requests, especially in the context of the IoT for smart cities. In this paper, we present an admission control model for M2M communications. The model differentiates all M2M requests into delay-sensitive and delay-tolerant first, and then aggregates all delay-tolerant requests by routing them into one low-priority queue, aiming to reduce the number of requests from various devices to the access point in the IoT for smart cities. Also, an admission control algorithm is devised on the basis of this model to prevent access collision and to improve the quality of service. Performance evaluations by network calculus, numerical experiments, and simulations show that the proposed model is feasible and effective.

Index Terms—Machine-to-machine (M2M) communications, Internet of Things (IoT), sustainable smart city, admission control, performance analysis

1 Introduction

The development of the Internet of Things (IoT) together with smart cities bring us not only new opportunities but new challenges as well. As part of the IoT, machine-to-machine (M2M) communications either broadly refer to the entire communications among man, machine, and system, or narrowly refer to the communications among machines (devices) only.

M2M communications can be understood as an automated process which requires minimum human interventions [1]. Tasks such as remote surveillance [2], health and environment monitoring, smart grids, smart cities [3], home and traffic security, and intelligent transportation [4], [5] are well-known instances of M2M communications. Personal navigation, e-pay, and industry automation are also expected to be benefited from M2M communications. Recent data collection suggests that the percentage of M2M connections in all Internet connections will grow from 24% to almost 43% by the year 2019 [6]. However, it has been observed that massive and concurrent machine accesses and radio signals generated by M2M devices (machine type communication devices (MTCDs)) in smart cities may cause unacceptable communication delays, data packet losses, and even service interruptions in human-to-human (H2H) communications [7]. Also, the sporadic and diverse nature of M2M traffic calls for a new design of wireless networks to deal with it. Notably, the emerging 5G wireless network design aims to provide a native support for M2M communications. The fact that the 5G technology promises to provide a widespread coverage, more mobility support, a lower latency, a prolonged battery life span, and a platform for a large number of devices makes 5G technology the key enabler for successful M2M communications [8]–[11].

One of the major challenges in the IoT for building sustainable smart cities resides in how to effectively handle the dramatic explosion of the number of connected devices. For example, the total number of mobile devices in 2014 was around 7.4 billion [6], nearly the same as the world population. According to data predictions, by 2019 there will be almost 11.4 billion mobile devices in the world. M2M connections are also expected to increase from 495 million in 2014 to almost 3 billion by 2019. Furthermore, the portion of M2M devices in cellular networks is expected to increase from 1% in 2014 to over 20% by 2019 [12]. As such, building sustainable smart cities needs to consciously consider not only the aggressive growth of mobile devices (including MTCDs), but also the consistently increasing traffic demand per device. It has been anticipated that the next generation 5G wireless network needs to deliver both a high data transmission rate [13] and a widespread connectivity covering about 300,000 devices within one cell.

With the recent activities of 3GPP, ETSI and IEEE standardization bodies intending to provide protocols and standards for M2M applications [14], note that most existing

- *This work was supported by NSFC under Grant Number 61671093.*

- *J. Huang is with the Institute of Electronic Information and Networking, Chongqing University of Posts and Telecommunications, Chongqing, China, 400065.*
  E-mail: xiaoniuadmin@gmail.com
- *C. Xing is with the Department of Mathematics and Computer Science, Nicholls State University, Thibodaux, LA 70310.*
  E-mail: cong-cong.xing@nicholls.edu
- *S Shin is with the Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD 57006.*
  E-mail: sung.shin@sdstate.edu
- *F Hou is with the Department of Electrical and Computer Engineering, University of Macau, Macau, China 999078.*
  E-mail: fenhou@umac.mo
- *C-H Hsu is with the Department of Computer Science and Information Engineering, Chung Hua University, Taiwan 300.*
  E-mail: chh@chu.edu.tw

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

4G base stations are designed to provide broadband services to regular H2H subscribers, and that M2M communications typically transmit small-sized packets in a frequent manner by using sporadic radio resources. So M2M communications are unable to effectively take advantage of the 4G communication channels. Although the idea of random accesses from MTCDs to channels may mitigate this problem, the huge number of MTCDs congested on channels will lead to a significant rise of collisions, a higher packet loss rate, and a performance degradation for both M2M and H2H services [15]. Incidentally, the co-existence of M2M and H2H communications is essential for both service providers and users. As such, to enhance the network performance, the spectrum utilization efficiency needs to be maximized while the M2M random accesses needs to be minimized.

Although extensive studies have been conducted for M2M communications in various aspects [16]–[18], the primary challenge in M2M communications lies in how to deal with the frequent and massive amounts of access requests sent from the exponentially increasing number of M2M devices in smart cities. Given the huge amounts of access requests raised by MTCDs in smart cities, traditional network gateway is no longer able to handle these requests satisfactorily. An effective admission model designed for M2M communications can not only reduce the number of collisions caused by MTCDs' random accesses, but also ensure an effective exploitation of the wireless resources.

It is well known that network calculus is an important and effective mathematical tool for the quantitative study of network system performances, and has been widely used in the modeling and analysis of quality of service (QoS) of networks. In particular, deterministic network calculus (DNC) calculates the delay bound, backlog bound, and other service quality parameters by using arrival curves and service curves. Compared with the traditional queueing theory, DNC is able to provide a determined boundary analysis for system performance, and offer a strict service guarantee by computing the worst-case scenarios. Taking the advantage of network calculus being a systematically structured theory, we, in this paper, propose a priority-based admission control model for M2M communications in smart cities, and analyze its performance by leveraging network calculus.

The main contributions of this paper are as follows.

- We present a new IoT architecture for smart cities, through which the network control and data transmission are separated. This architecture follows the same design philosophy of SDN, and thus enhances the manageability and the controllability of the entire network. Under this architecture, we propose an admission control model which first differentiates M2M access requests as delay-sensitive and delay-tolerant, and then aggregates all delay-tolerant requests into a low-priority queue. The proposed admission control model can effectively reduce the number of needed connections from MTCDs to base stations, and mitigate the possibility of collisions on channels generated by MTCDs' random requests.
- We present an admission control algorithm for massive M2M requests in smart cities. The algorithm can effectively prevent access request congestions, thereby improving the quality of M2M access connections. We also apply the network calculus to analyzing the performance of the proposed algorithm. Performance bounds including the worst-case delay and backlog bounds, which provide design guidelines for building sustainable smart cities, are derived.
- We evaluate the proposed model and examine the theoretical results by conducting extensive experiments. The validness and effectiveness of the developed theory are further confirmed. The idea of aggregating access flows of a massive number of devices/machines in the context of smart cities can effectively enhance their sustainability.

The reminder of this paper is structured as follows. Section 2 reviews related work. Section 3 introduces the M2M system model, the admission control model, and some network calculus basics. Section 4 describes the admission control algorithm, which is followed by, in Section 5, the performance analysis of the system. Experimental results are presented in Section 6, and Section 7 concludes the paper.

2 Related Work

Many researchers and standardization bodies, such as 3GPP and ETSI, are dedicated to reducing the wireless network load and using radio resources more efficiently. Standardization of M2M communications, together with related requirements and architectures, have already been proposed by 3GPP [19]. Chen et al. [20] presented a survey of recent developments in home M2M networks, summarizing the architecture M2M communications and positing some related challenges. These challenges were about the large-scale maintenance of devices and remote management, and these issues were addressed in [21]. Zheng et al. [15] explicated the M2M communication architecture and performance in LTE-advanced networks.

Solutions to M2M support and LTE resource management were proposed in [22], [23]. The existing M2M solutions can be divided into two categories: (i) radio resource optimization and (ii) co-operation among devices. Viewed from the networking perspective, radio resource management is of the utmost importance in terms of maintaining a certain level of QoS. Massive access management [22] and reliable resource pooling schemes [24] were proposed to ensure the QoS and the reliability of M2M operations. In [25], Liu et al. proposed a scalable hybrid MAC protocol for machine type communications within heterogeneous networks. A batch data model was suggested by [26] and [27] to reduce the updating frequency of M2M core networks. A self-adaptive access barring parameter [27] was used to optimize the system performance by changing resource blocks. IEEE 802.11 ah MAC for M2M communications was enhanced in [28] with the mechanism of self-adaptive Restricted Access Windows (RAW). Various MAC protocols for M2M communications were surveyed in [23]. To support more machine accesses, MTCDs, just like the base stations, can be grouped together to collaborate with one another toward load balancing or resource sharing. A co-operative access class barring protocol to balance the number of MTC requests in overlapping macro- and micro-cell coverage

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

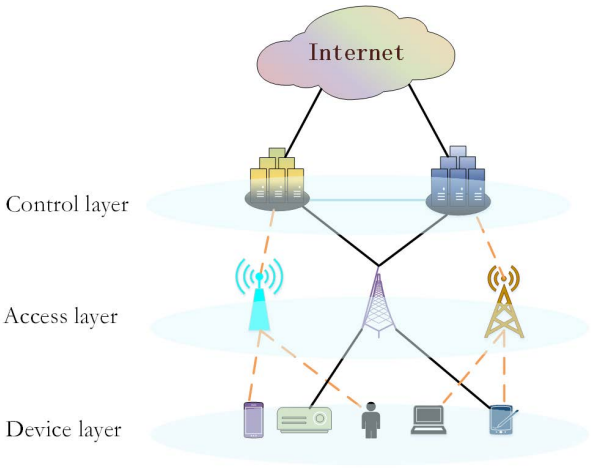


Fig. 1: The architecture of the M2M-supporting IoT system for smart cities.

area was proposed in [29]. In [30], Wang et al. described a clustered M2M network and studied the spatial reuse of random access resources in LTE-Advanced environment to support a larger number of MTCDs and to reserve more random access resources for H2H communications. The use of capillary network and gateway was studied by many researchers to reduce network congestions [20], [26], [31]. Also, the need for cognitive network gateways was emphasized in [20].

Most M2M research work in the literature, up to this point, focuses on radio resource management, base station load balancing, and M2M devices grouping. Few studies have been conducted on the optimization of networks through the access control of M2M requests. Data aggregation for M2M gateways, as described in [23], determines the optimal durations for data transmissions from MTC devices to gateway and from gateway to server. In wireless sensor networks (WSNs) and other wireless networks, there are some notable work to reduce the network load caused by data transmissions. Although the technique of self-adaptive data compression has been used in WSNs for bandwidth management and location updating, it has not been addressed in the context of M2M communications, which consists of enormous concurrent transmission requests from a myriad of devices. This observation motivates us to explore new architectures and techniques in designing novel and more efficient strategies to handle the admission control in M2M gateways.

3 M2M Admission Control Model

3.1 System Model

The successful design and implementation of M2M communications in the IoT for sustainable smart cities must be supported by an existing architectural framework. Only under such an architectural framework support, can ubiquitous MTCDs be effectively allowed to access base stations. There is an urgent need for the IoT to become an open, complete, standardized, and universal architectural framework, which facilitates various newly developed techniques, including

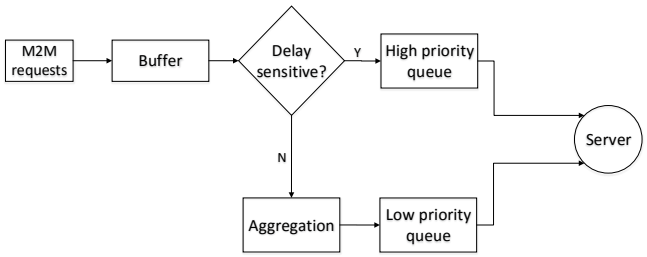


Fig. 2: M2M admission control model.

M2M communications, to be included in a consistent and effective manner.

Unfortunately, such an expectation is not met by the current IoT. Much like its concept, the current IoT lacks a widely-agreed, uniformed, and normalized architecture to support most conceivable functions in the world. In this section, we sketch a new IoT management and control architecture that intends to support intelligent M2M communications in smart cities. As shown in Fig. 1, the proposed architecture comprises 3 layers: a device layer, an access layer, and a control layer. The device layer is located at the bottom of the system and consists of a variety of wireless terminals. The access layer is located at the middle of the system and consists of cellular base stations and/or WiFi access points, which allows the bottom layer devices to access networks through cellular and/or WiFi technologies. The top layer is the control layer which provides administration and control over the access requests and data transmission activities at the lower layers, and is connected to the Internet for real time analyses and feedbacks.

It is worth mentioning that the devices and infrastructures in the device and access layers only forward data traffic, while the control mechanisms located in the control layer are dedicated to network control or management. This design follows the same philosophy of the Software-Defined Networking (SDN) paradigm that decouples the network control and data forwarding functionalities in order to enhance the controllability and manageability. Also, the interactions between the control and access layers are consistent with operations specified in the SDN protocol, this interaction-formed protocol is assumed given in this work.

3.2 Admission Control Model

We now introduce a request aggregating admission control model for M2M communications, which is depicted in Fig. 2. In the access layer depicted in Fig. 1, the incoming M2M requests for each base station or access point are classified either as delay-sensitive or as delay-tolerant. In conformity with this classification, each base station or access point is equipped with two queues inside: one is of high priority and is used to queue delay-sensitive requests; the other is of low priority and is used to queue delay-tolerant requests. As indicated in Fig. 2, when an M2M request arrives at a base station or an access point, it will be buffered first, and then be routed to the high priority queue if it is delay-sensitive, and the low priority queue otherwise. In this case, *all* delay-tolerant M2M requests

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

will be aggregated into *one* low priority queue waiting to be batch-processed. This would not be the case without the request aggregation idea: different delay-tolerant M2M requests would be routed to different queues to be process separately, depending on the extent of their delay-tolerance.

It should be noticed that the starvation of the low priority flow would be an issue that cannot be avoided in the above model. However, as the main focus of this work is to develop a priority-based model for M2M request access in smart cities to guarantee the delay performance of M2M communications, the issues of how to avoid the starvation and to ensure the fairness of multiple flows (no guarantees regarding delays can be provided in this case), are not discussed here. Also, the above model is suitable for smart city scale applications. For example, consider a public safety scenario in a smart city where a severe flood warning is being issued. In this case, the data update on water levels of surrounding bayous, rivers, ditches, and culverts in various areas of the city is of critical importance, and forms a delay-sensitive request flow. Other requests, such as air pollution monitoring, vehicle parking, transportation scheduling, etc. are aggregated as the delay-tolerant flow. Note that the criteria for flow classification vary from one case to another [32]. For other scenarios, the criteria for determining delay-sensitive or delay-tolerant would be different. This issue is out of the scope of this paper but is worth of studying in future investigations.

Network calculus will be used to analyze the performance of this setting. We assume that all access request flows are regulated by the common technique of token bucket, so that the request flows can be processed smoothly. The basic working mechanism of token bucket is as follows. Let r be the rate of adding tokens to the bucket, i.e., one token will be added to the bucket per $1/r$ second, and b be the maximum number of tokens the bucket can hold. If a token arrives when the bucket is full, then that token will be dropped. When a flow containing n requests arrives, n tokens will be removed from the bucket. The flow will be sent to the network if the bucket has more than n tokens available; otherwise, the flow has to wait, until the bucket acquires sufficient number of tokens, to be transmitted further. As such, the input function of a flow, after being shaped by the token bucket, is $f(t) = rt + b$, where r is the rate and b is the initial burst traffic.

3.3 Network Calculus Basics

Network calculus is a theory of queuing systems, which offers a deep insight into data flow problems found in computer networks. It was initiated by Chang [33] and Cruz [34], and further developed by Agrawal [35], Le Boudel [36], and others. At present, network calculus has successfully found applications in many areas such as QoS control, software defined networks, traffic scheduling, and controls of queue lengths and delays. The basic notions and notations of network calculus that are used in this paper are briefly explained below.

Definition 1 (Wide-Sense Increasing Functions). Given a function f defined over $R \cup \{+\infty\}$. If $\forall s, t \in (R \cup \{+\infty\}), s \leq t$ implies $f(s) \leq f(t)$, we then call f a wide-sense increasing function.

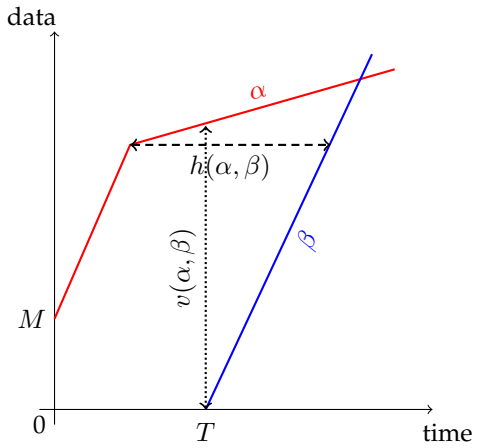


Fig. 3: Calculations of backlog bound and delay bound.

Definition 2 (Set of Wide-Sense Increasing Functions). The set F of wide-sense increasing functions is defined as follows.

$$F = \{f(t) | f(t) = 0, \forall t < 0; \quad f(0) \geq 0; \quad f(u) \leq f(t), \forall u \leq t, u, t \in [0, +\infty]\}.$$

Definition 3 (Arrival Curve). Given a wide-sense increasing function α defined for $t \geq 0$ (i.e. $\alpha \in F$), we say that a flow with (accumulative) input function R is constrained by α if and only if for all $s \leq t$, we have $R(t) - R(s) \leq \alpha(t-s)$ and in this case we say that R has α as an arrival curve, or that R is α-smooth.

Definition 4 (Service Curve). Consider a system S and a flow going through S with R and R^* as its (accumulative) input and output functions. We say that β is the service curve offered by the system S if and only if β is wide-sense increasing, $\beta(0) = 0$, and $R^* \geq R \otimes \beta$. Typically, β is expressed as $\beta(t) = r(t - T)^+$, where r is the service rate and T is the time delay.

Theorem 1 (Backlog Bound). Suppose a data flow with input function R and output function R^* is constrained by α and goes through a system whose service curve is β. Then at any time t, the backlog $R(t) - R^*(t)$ satisfies

$$R(t) - R^*(t) \leq v(\alpha, \beta) = \sup_{s \geq 0}\{\alpha(s) - \beta(s)\}$$

where $v(\alpha, \beta)$ is the vertical deviation between α and β.

Theorem 2 (Delay Bound). Suppose a data flow is constrained by α and goes through a system whose service curve is β. Then at any time t, the virtual delay $d(t)$ satisfies

$$d(t) \leq h(\alpha, \beta) = \sup_{s \geq 0}\{\inf\{T \geq 0 : \alpha(s) \leq \beta(s + T)\}\}$$

where $h(\alpha, \beta)$ is the horizontal deviation between α and β.

The calculations of backlog bound and delay bound of flows are illustrated in Fig. 3, where the solid lines are arrival curve α (upper red one) and service curve β (lower blue one), respectively, and the dashed line and dotted

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

Algorithm 1 Admission Control Algorithm

Input:
$r_i, r_h, r_l, r; \quad \forall i \in \{1, 2, ..., I\}$
Output:
accept/reject request$_i$ $\quad \forall i \in \{1, 2, ..., I\}$
1: For each current request$_i$
2: **if** $r_i \leq r - r_h - r_l$ **then**
3: accept request$_i$;
4: **else**
5: reject request$_i$;
6: **end if**
7: update r_h, r_l;
8: Repeat the process for the next due request;

line are delay bound and backlog bound, respectively. As shown in this figure, the delay bound $h(\alpha, \beta)$ is essentially the maximum horizontal deviation between arrival curve and service curve, and the backlog bound $v(\alpha, \beta)$ is the maximum vertical deviation between those two curves.

4 The Admission Control Algorithm

Due to the enormous amounts of M2M communication requests in smart cities, an acceptance and/or rejection algorithm with regard to these requests must be in place to avoid traffic congestion in the network. We thus devise such an algorithm to handle the large-scale M2M access requests on the basis of their arrival rate and the service capability of the service node.

One of the critical conditions to ensure a regular trouble-free running of a network is that the arrival rate of the data flow cannot be larger than the service rate (or capability) of the network. Otherwise, the flow will tend to encounter an infinite delay, causing a malfunction of the network. If the current service capability of the network is able to handle the incoming access request, then the request will be accepted; otherwise, it will be rejected. The notations used in the algorithm and the algorithm itself are given in and Table 1 and Algorithm 1, respectively.

TABLE 1: Notations of the Algorithm

Notation	Meaning
r	The total service rate (capability) of the system
r_i	The arrival rate of the i-th M2M device
r_h	The arrival rate of the unfinished high-priority flow
r_l	The arrival rate of the unfinished low-priority flow
I	The total number of M2M devices

5 Performance Analysis

By the model in Fig. 2, all M2M communication requests will be routed into a delay-sensitive high-priority queue or a delay-tolerant low-priority queue. In this section, we use f_h and f_l to denote the flows generated by the high-priority queue and the low-priority queue, respectively; R_h and R_h^* to denote the input and output functions of f_h , and R_l and R_l^* to denote the input and output functions of f_l.

We assume that the order of arrivals of f_h and f_l is completely arbitrary without any timing constraints, that the service curve β offered by the system follows the rate-latency function [37], i.e., $\beta_{r,T}(t) = r(t-0)^+$, and that f_l and f_h have $\alpha_l(t) = r_l(t) + b_l$ and $\alpha_h(t) = r_h(t) + b_h$ as their arrival curves, respectively. (Note that the reason of having these linear arrival curves was explained at the end of Section 3.2.) In the sequel, we analyze the performance of the system by the order of arrivals of f_l and f_h and by considering cases of preemptive scheduling and non-preemptive scheduling.

5.1 f_h Arrives Earlier than or Simultaneously with f_l

In this case, preemptive scheduling or non-preemptive scheduling will make no difference. We assume that some requests exist in the system before the M2M requests arrive and are pushed into the queues. These pre-existing requests in the system are called unfinished requests. The system will first serve those unfinished requests at hand, and then start serving requests from f_h and f_l in order. Let $l_{\max}$ be the amount of unfinished requests in the system, r be the total service rate of the system. Then in the time interval $(s, t]$, the amount of requests formed in f_h is

$$R_h^*(t) - R_h^*(s) \geq r(t-s) - l_{\max} \tag{1}$$

with

$$\beta_{r_h,T} = r(t - \frac{l_{\max}}{r}). \tag{2}$$

By Theorems 1 and 2, the delay bound and backlog bound can be obtained as follows:

$$d_h \leq \frac{b_h + l_{\max}}{r}, \tag{3}$$

$$q_h \leq b_h + r_h \frac{l_{\max}}{r}. \tag{4}$$

Similarly, the amount of requests formed in f_l is

$$R_l^*(t) - R_l^*(s) \geq r(t-s) - [R_h^*(t) - R_h^*(s)] - l_{\max} \tag{5}$$

with

$$\beta_{r_l,T} = (r - r_h)(t - \frac{b_h + l_{\max}}{r - r_h}), \tag{6}$$

and its delay bound and backlog bound would be

$$d_l \leq \frac{b_l + b_h + l_{\max}}{r - r_h} \tag{7}$$

and

$$q_l \leq b_l + r_l \frac{b_h + l_{\max}}{r - r_h}, \tag{8}$$

respectively.

5.2 f_h Arrives Later than f_l

In this case, preemptive scheduling or non-preemptive scheduling yields different results, and needs to be considered separately.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

5.2.1 Non-Preemptive Scheduling

By virtue of the nature of the non-preemptive scheduling algorithm, the system will serve its unfinished requests first, and then start processing requests from f_l and f_h in order. As such, the amount of requests formed in f_l in the time interval $(s,t]$ is

$$R_l^*(t) - R_l^*(s) \geq r(t-s) - l_{\max} \tag{9}$$

with

$$\beta_{r_l,T} = r(t - \frac{l_{\max}}{r}). \tag{10}$$

The delay bound and backlog bound of f_l can be obtained as follows:

$$d_l \leq \frac{b_l + l_{\max}}{r}, \tag{11}$$

$$q_l \leq b_l + r_l \frac{l_{\max}}{r}. \tag{12}$$

In a similar fashion, the amount of requests formed in f_h in the time interval $(s,t]$ is

$$R_h^*(t) - R_h^*(s) \geq r(t-s) - [R_l^*(t) - R_l^*(s)] - l_{\max} \tag{13}$$

with

$$\beta_{r_h,T} = (r - r_l)(t - \frac{b_l + l_{\max}}{r - r_l}), \tag{14}$$

and f_h's delay bound and backlog bound are

$$d_h \leq \frac{b_l + b_h + l_{\max}}{r - r_l} \tag{15}$$

and

$$q_h \leq b_h + r_h \frac{b_l + l_{\max}}{r - r_l}, \tag{16}$$

respectively.

5.2.2 Preemptive Scheduling

In this case, with the assumption that the current f_h is empty, the system will serve requests from f_l until the request from f_h arrives. At that time, the system will stop processing requests from f_l and start processing requests from f_h until all requests from f_h have been processed, and then resume the processing of requests from f_l. As such, the amount of packets transmitted by f_h in the time interval $(s,t]$ is

$$R_h^*(t) - R_h^*(s) \geq r(t-s) \tag{17}$$

with

$$\beta_{r_h,T} = r(t-0)^+. \tag{18}$$

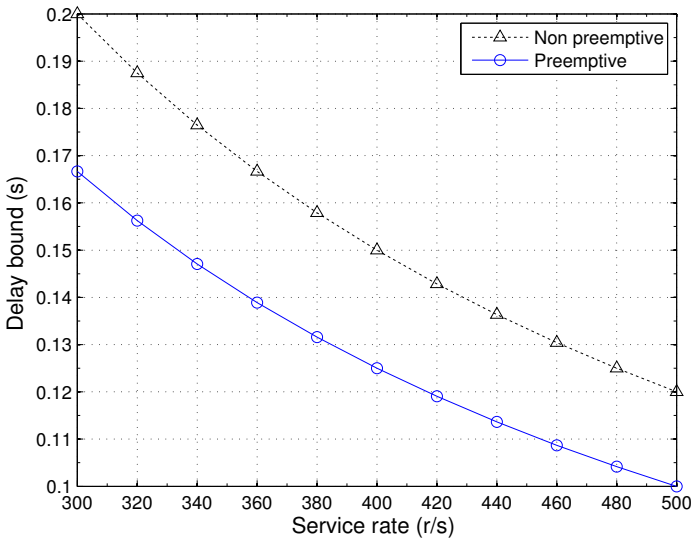
By Theorems 1 and 2, we can find its delay bound and backlog bound as follows:

$$d_h \leq \frac{b_h}{r}, \tag{19}$$

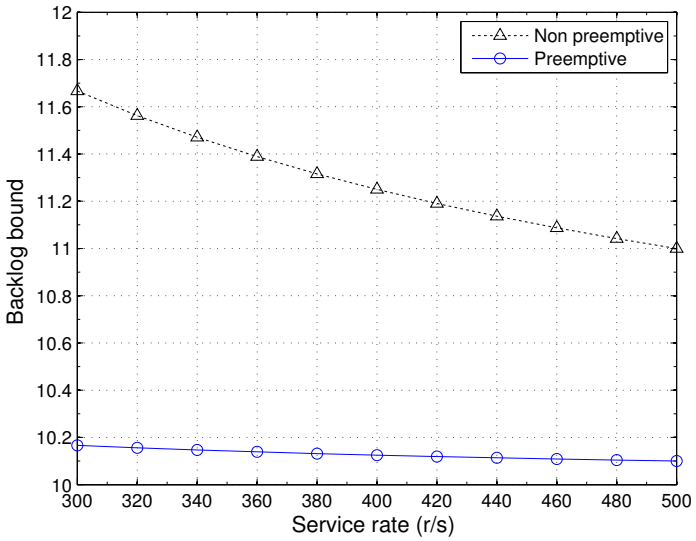
$$q_h \leq b_h + \frac{r_h}{r}. \tag{20}$$

Similarly, the amount of requests formed in f_l in the time interval $(s,t]$ is

$$R_l^*(t) - R_l^*(s) \geq r(t-s) - [R_h^*(t) - R_h^*(s)] \tag{21}$$



(a) Delay bound



(b) Backlog bound

Fig. 4: Delay bound and backlog bound of f_h with respect to preemptive scheduling and non-preemptive scheduling.

with

$$\beta_{r_l,T} = (r - r_h)(t - \frac{b_h}{r - r_h}), \tag{22}$$

and f_l's delay bound and backlog bound are

$$d_l \leq \frac{b_l + b_h}{r - r_h} \tag{23}$$

and

$$q_l \leq b_l + r_l \frac{b_h}{r - r_h}, \tag{24}$$

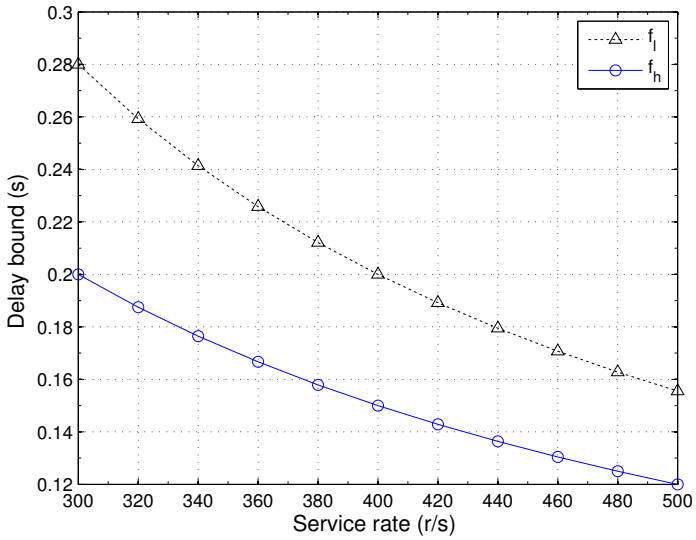
respectively.

6 Experimental Results

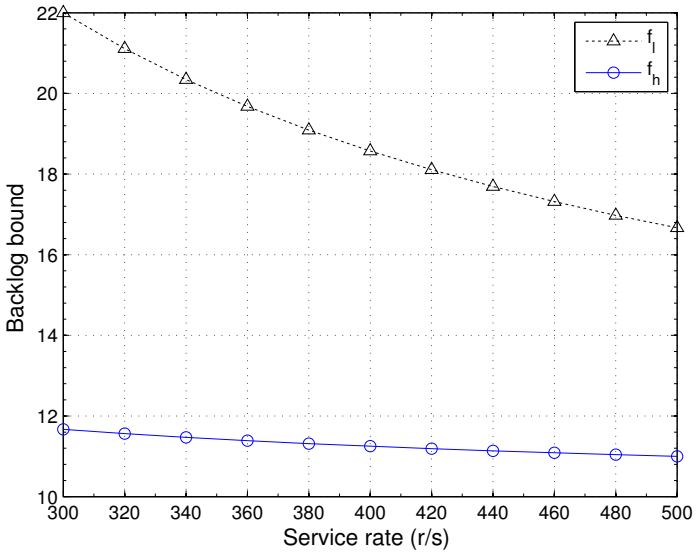
6.1 Numerical Experiments

In this section, we continue the theoretical performance analysis carried out in the previous section together with

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

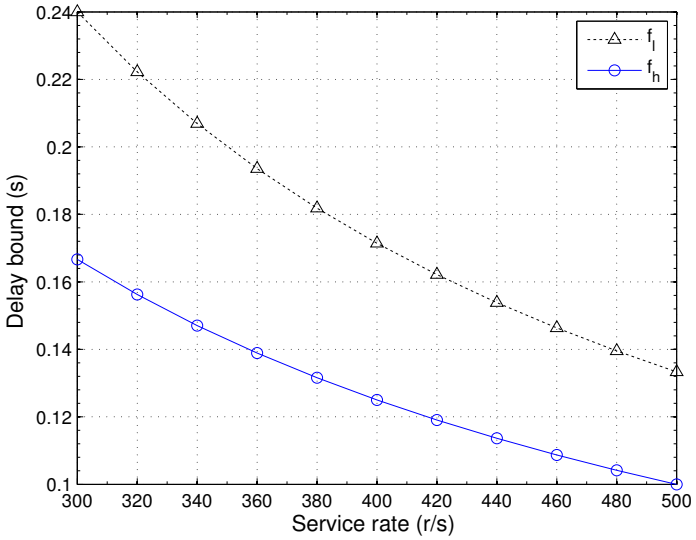


(a) Delay bound

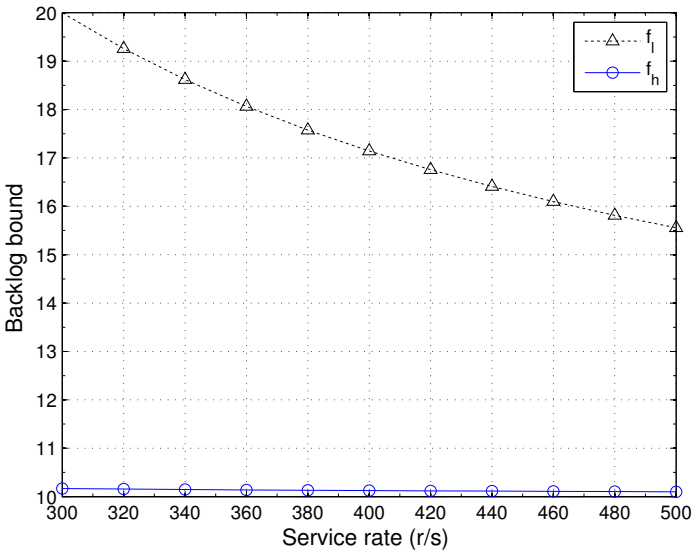


(b) Backlog bound

Fig. 5: Delay bounds and backlog bounds of f_h and f_l, with non-preemptive scheduling, when f_h arrives earlier than or simultaneously with f_l.



(a) Delay bound



(b) Backlog bound

Fig. 6: Delay bounds and backlog bounds of f_h and f_l, with preemptive scheduling, when f_h arrives earlier than or simultaneously with f_l.

the conduction of numerical experiments. For all figures in this section, except Fig. 7, the parameters are set as follows: $r \in [300, 500]$ requests/second (r/s), $r_l = r_h = 50$ r/s, $b_l = b_h = 10$, and $l_{\max} = 10$. Note that these parameters used here are intended to show the geographical features of theoretical results derived from previous section. Other parameters can also be configured to conduct the same analysis as shown in the following.

Fig. 4 shows the delay bound and backlog bound of f_h with respect to preemptive scheduling and non-preemptive scheduling. It can be seen clearly that preemptive scheduling delivers a superior performance than non-preemptive scheduling. This observed result can also be derived by purely analyzing the relevant bounds obtained in the previous section, as follows. By (19), (15), (20), and (16), we can see that

$$\frac{b_h}{r} < \frac{b_h}{r - r_l} < \frac{b_l + b_h + l_{max}}{r - r_l}$$

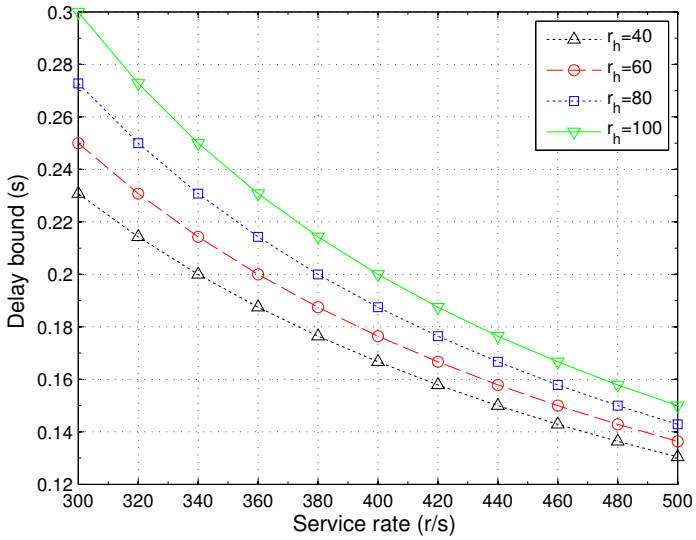
and

$$b_h + \frac{r_h}{r} < b_h + \frac{r_h}{r - r_l} < b_h + \frac{r_h(b_l + l_{max})}{r - r_l},$$

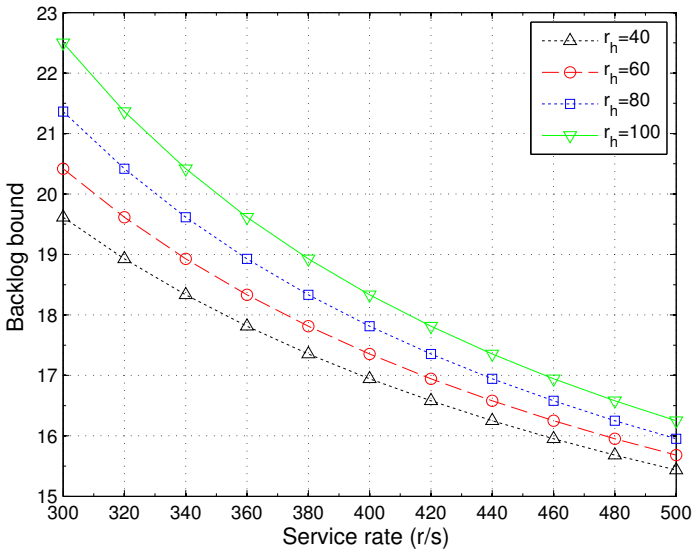
when r_l, b_l, and l_{max} are all positive, which show that the delay bound and backlog bound of f_h with preemptive scheduling are smaller than that with non-preemptive scheduling.

Fig. 5 depicts the delay bounds and backlog bounds of f_l and f_h when f_h arrives earlier than or simultaneously with f_l, and the non-preemptive scheduling scheme is used. We can see that f_h has lower backlog bound and delay

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing

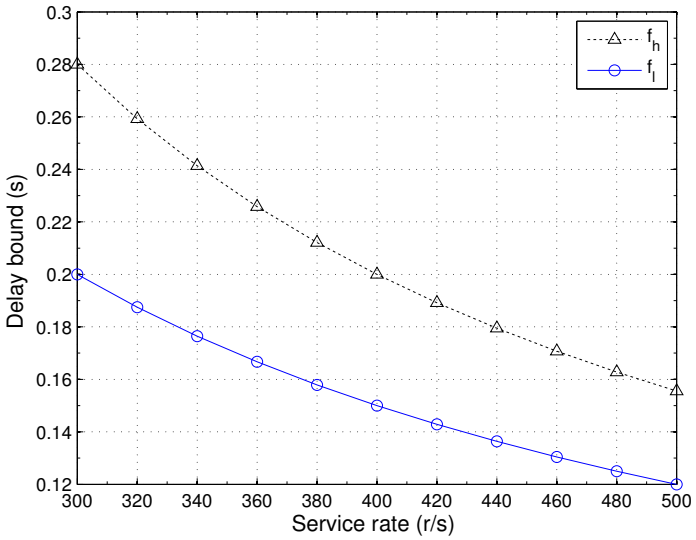


(a) Delay bound

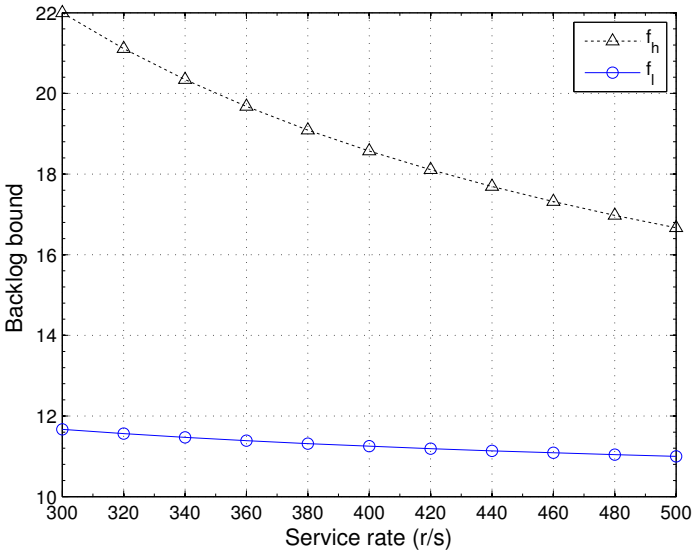


(b) Backlog bound

Fig. 7: With preemptive scheduling, the changes of delay bounds and backlog bounds of f_l with respect to the changes of arrival rate r_h of f_h.



(a) Delay bound



(b) Backlog bound

Fig. 8: Delay bound and backlog bound of f_l and f_h with non-preemptive scheduling when f_h arrives later than f_l.

bound than f_l, which is also consistent with the theoretical result obtained in the previous section, as detailed below. Referring to (3), (7), (4), and (8), when all parameters are positive and $b_h = b_l, r_h = r_l$, we have,

$$\frac{b_h + l_{max}}{r} < \frac{b_h + l_{max}}{r - r_h} < \frac{b_l + b_h + l_{max}}{r - r_h}$$

and

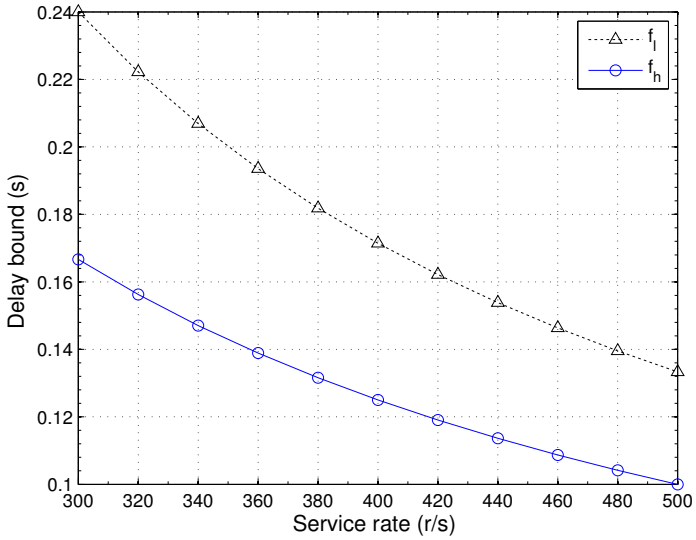
$$b_h + r_h\frac{l_{max}}{r} < b_h + r_h\frac{l_{max}}{r - r_h} < b_h + r_h\frac{b_h + l_{max}}{r - r_h} = b_l + r_l\frac{b_h + l_{max}}{r - r_h}.$$

Hence, both delay bound and backlog bound of f_h are smaller (lower) than that of f_l. Fig. 6 exhibits the situation of Fig. 5 with preemptive scheduling, instead of non-preemptive scheduling, being used. Similar results to that of Fig. 5 can be observed and mathematically derived as well.

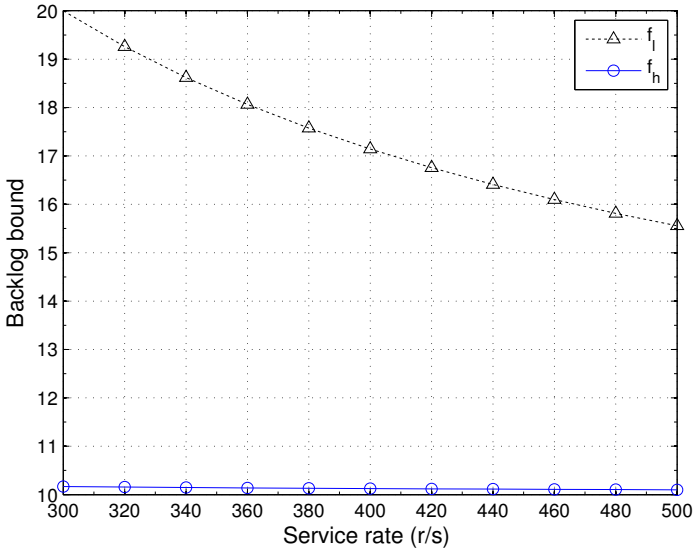
Fig. 7 shows the variations of the delay bound and the backlog bound of f_l with the arrival rate of f_h being 40, 60, 80, and 100, when the preemptive scheduling scheme is used. Clearly, both bounds increase as the r_h increases. This can be seen by the following two cases. Case 1: f_h arrives earlier than or at the same time with f_l. In this case, by (7) and (8), we can see that when r_h increases and other parameters remain unchanged, the values of $r - r_h$ will decrease resulting in a larger (higher) value of both

$$\frac{b_l + b_h + l_{max}}{r - r_h}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing



(a) Delay bound



(b) Backlog bound

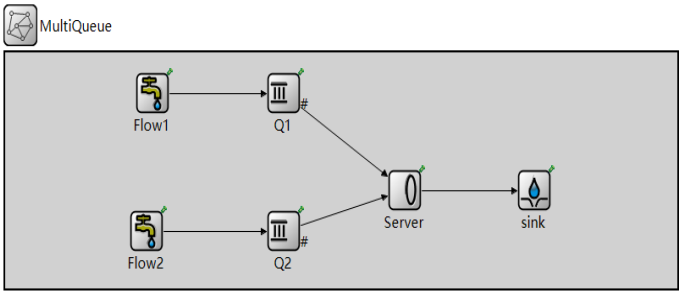
Fig. 9: Delay bound and backlog bound of f_l and f_h with preemptive scheduling when f_h arrives later than f_l.

and

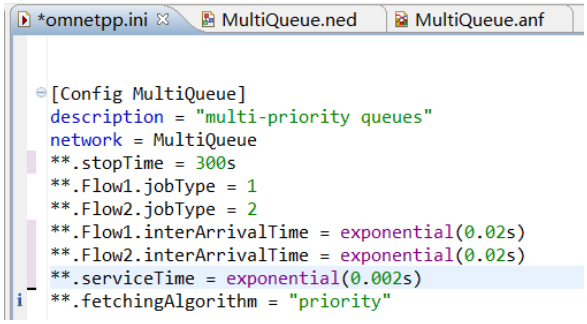
$$b_l + r_l \frac{b_h + l_{max}}{r - r_h}.$$

Case 2: f_h arrives later than f_l. The reasoning for this is similar to the first case, except that we need to use the formulas in (23) and (24). Either way, the analyses show that as r_h increases, both delay bound and backlog bound of f_l increase as well.

Fig. 8 and Fig. 9 depict, respectively, the situations when f_h arrives later than f_l and either a non-preemptive or a preemptive scheduling scheme is used. Clearly, Fig. 8 shows that the delay and backlog bounds of f_h are *higher* than that of f_l. This result should not be surprising as (15) gives and delay bound of f_h and (11) gives the delay bound of f_l, and



(a) Two flows/queues (Q1 = high-priority, Q2 = low-priority).



(b) Parameter settings.

Fig. 10: Setup of the simulation experiment using the OMNeT++ software.

we have

$$\frac{b_l + b_h + l_{max}}{r - r_l} > \frac{b_l + b_h + l_{max}}{r} > \frac{b_l + l_{max}}{r}.$$

The situation of backlog bounds is similar. By (16) and (12), we see

$$b_h + r_h \frac{b_l + l_{max}}{r - r_l} > b_h + r_h \frac{b_l + l_{max}}{r} > b_h + r_h \frac{l_{max}}{r} = b_l + r_l \frac{l_{max}}{r}$$

provided that $b_h = b_l$ and $r_h = r_l$. Fig. 9 shows an opposite situation. That is, the delay and backlog bounds of f_h are *lower* than that of f_l, which can also be seen by pure mathematical reasoning. Specifically, by (19), (23), (20), and (24), and the assumption that $b_h = b_l$ and $r_h = r_l$, the fact that

$$\frac{b_h}{r} < \frac{b_h}{r - r_h} < \frac{b_l + b_h}{r - r_h}$$

and

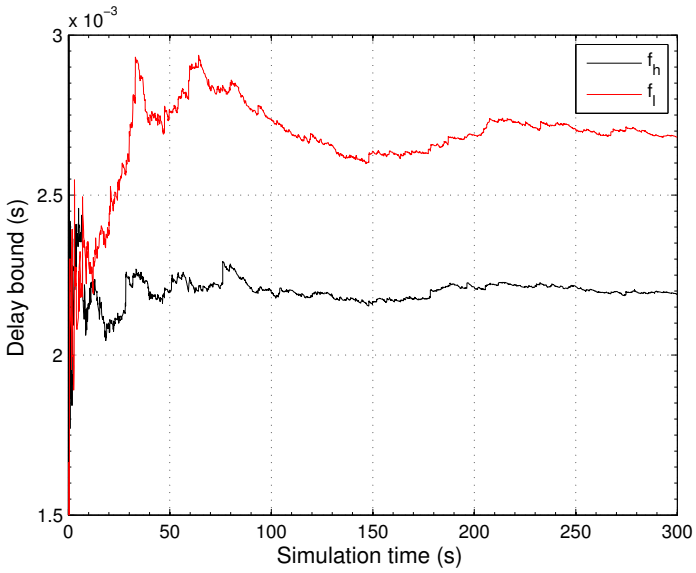
$$b_h + \frac{r_h}{r} < b_h + \frac{r_h}{r - r_h} < b_h + \frac{r_h b_h}{r - r_h} = b_l + \frac{r_l b_h}{r - r_h}$$

shows exactly what exhibited in Fig. 9.

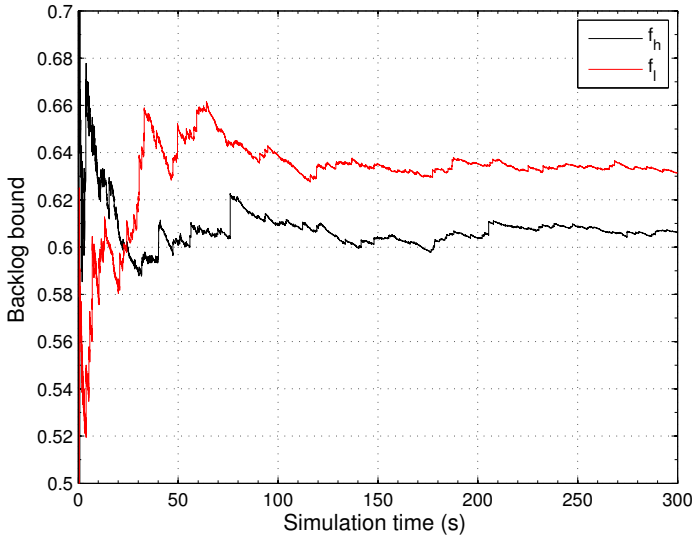
6.2 Simulations

We now demonstrate the result of simulating our proposed model by using the OMNeT++ [38] simulation software. Two screenshots associated with the simulation setup using this software are shown in Fig. 10. Fig. 10(a) is a simulation of the model depicted in Fig. 2, in which

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing



(a) Delay bound



(b) Backlog bound

Fig. 11: Delay bounds and backlog bounds of f_h and f_l with preemptive scheduling.

Flow1 is time-sensitive (i.e., Flow1 = f_h, or Q1 is of high-priority) and Flow2 is time-tolerant (i.e., Flow2 = f_l, or Q2 is of low-priority). Fig. 10(b) shows the details of parameter settings. Specifically, the two lines that indicate the `jobType` of Flow1 and Flow2 specify that Flow1 has higher priority that Flow2; the next two lines that indicate the `interArrivalTime` specify that, for both Flow1 and Flow2, a request will arrive every 0.02 seconds, or 50 requests will arrive in 1 second (i.e., $r_h = r_l = 50$ r/s); and the line that indicates `serviceTime` specifies that the service rate r of the server is 500 r/s ($1/500 = 0.002$).

Note that the above two-flow simulation setup is valid and reasonable. To see this, we can still use the example of public safety scenario in smart cities. While the high-priority flow would comprise the data processing requests of water levels of surrounding bayous, rivers, ditches, and culverts in various areas of the city, the data processing requests for air pollution monitoring, vehicle parking, transportation management, etc. would constitute the low-priority flow. As our model enables the delay-tolerant requests to be aggregated, so only two flows are needed and thus configured here in the simulation setup. As such, this simulation setting can faithfully reflect the performance of the proposed model in a smart city scale.

Fig. 11 shows the delay bounds and backlog bounds of f_h and f_l under such a setting and the preemptive scheduling scheme is utilized. We can see that f_h excels f_l in terms of both delay bound and backlog bound. This is not surprising, and actually matches the theoretical analysis as expected, due to the nature of the preemptive scheduling. Fig. 12 depicts the variations of the delay bound and the backlog bound of f_l with respect to $r_h = 40, 60, 80, 100$ r/s when the service rate $r = 500$ r/s and the arrival rate r_l of f_l is 50 r/s. Again, we are able to see that both delay bound and backlog bound of f_l grow as the arrival rate r_h of f_h grows, and this is also consistent with the numerical experiments in the previous section and the theoretical results in Section 5.

In a nutshell, the purpose of the simulation in this section is to examine the properties of the high-priority flow and the low-priority flow, and compare them against the results obtained in Sections 5 and 6, attempting to establish a validation for those results. Based on the observations from the simulation, this validation is clearly achieved.

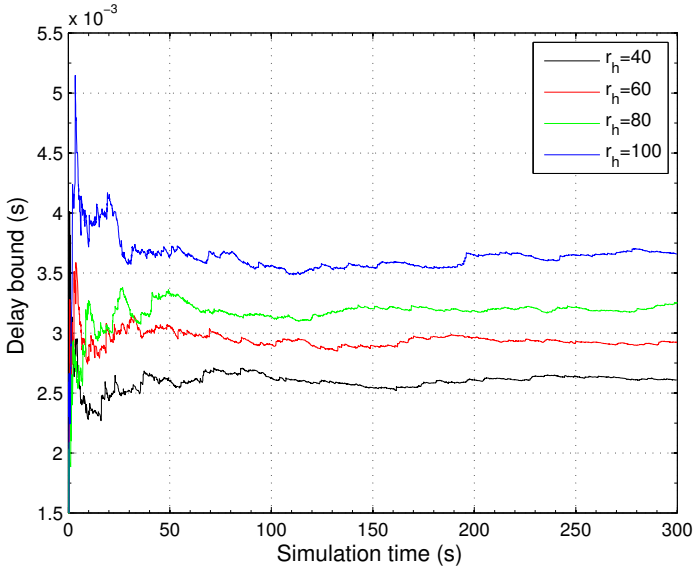
7 Conclusion

The exponential increase in the amount of M2M devices induced by the rapid development of the IoT in smart cities can no longer be adequately handled by the traditional network gateway techniques. How to effectively deal with such an enormous amount of M2M access requests and the ensuing data transmissions in IoT thus becomes the bottleneck hindering the development of sustainable smart cities. Considering that most M2M requests are delay-tolerant, we have presented an architecture for IoT control and management in smart cities, and proposed a priority-based model for M2M communications, which can reduce the collision possibility caused by random M2M accesses on wireless channels. The performance of this model is subsequently analyzed and evaluated by using network calculus, by numerical experiments, and by simulations using the OMNeT++. The consistency of results in network calculus, numerical experiments, and OMNeT++ simulation validates the effectiveness and correctness of the proposed model.

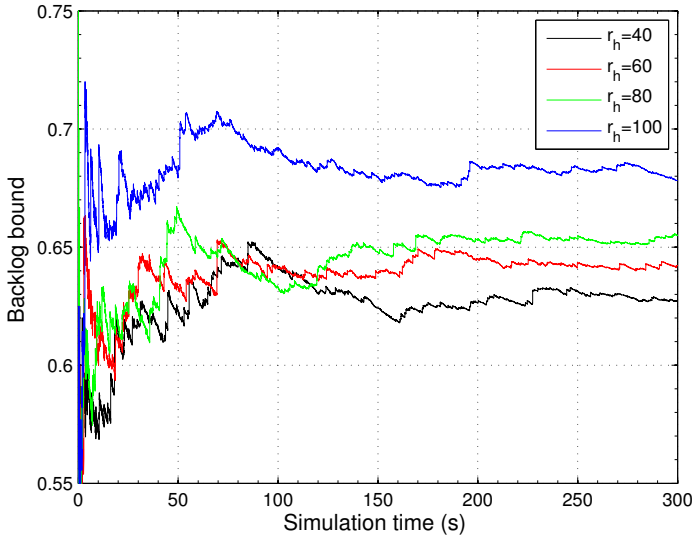
For future work, we plan to investigate the following problems based on the current work.

- The multi-priority model would cause the starvation of the low priority flow, although the model can guarantee the delay performance. How to resolve this starvation issue and assure the fairness of the flows are interesting topics worthy of further investigations.
- The network calculus used in this paper is deterministic. It would be interesting to extend the current work by using stochastic network calculus to analyze

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSUSC.2017.2702589, IEEE Transactions on Sustainable Computing



(a) Delay bound



(b) Backlog bound

Fig. 12: Changes of delay bound and backlog bound of f_l with respect to changes of r_h.

the proposed model as it offers a more generic treatment for real-world IoT for sustainable smart cities.

REFERENCES

[1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.

[2] S. He, J. Chen, X. Li, X. . Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 6, pp. 1268–1282, June 2014.

[3] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb 2014.

[4] S. He, D. H. Shin, J. Zhang, J. Chen, and Y. Sun, "Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 9, pp. 7448–7461, Sept 2016.

[5] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, "Energy-efficient probabilistic area coverage in wireless sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 367–377, Jan 2015.

[6] "Cisco Virual Networking," http://www.cisco.com/, 2016.

[7] S. C. Lin and K. C. Chen, "Cognitive and opportunistic relay for qos guarantees in machine-to-machine communications," *IEEE Transactions on Mobile Computing*, vol. 15, no. 3, pp. 599–609, March 2016.

[8] Y. Gao, Z. Qin, Z. Feng, Q. Zhang, O. Holland, and M. Dohler, "Scalable amp; reliable iot enabled by dynamic spectrum management for m2m in lte-a," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2016.

[9] M. Abu-Lebdeh, J. Sahoo, R. Glitho, and C. W. Tchouati, "Cloudifying the 3gpp ip multimedia subsystem for 4g and beyond: A survey," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 91–97, January 2016.

[10] "External integrity verification for outsourced big data in cloud and iot: A big picture," *Future Generation Computer Systems*, vol. 49, pp. 58 – 67, 2015.

[11] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile phone sensing systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, First 2013.

[12] N. Ericsson, "Ericsson mobility report," 2014.

[13] A. Osseiran, F. Boccardi, V. Braun, K. Kusume, P. Marsch, M. Maternia, O. Queseth, M. Schellmann, H. Schotten, H. Taoka, H. Tullberg, M. A. Uusitalo, B. Timus, and M. Fallgren, "Scenarios for 5g mobile and wireless communications: the vision of the metis project," *IEEE Communications Magazine*, vol. 52, no. 5, pp. 26–35, May 2014.

[14] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, "M2m scheduling over lte: Challenges and new perspectives," *IEEE Vehicular Technology Magazine*, vol. 7, no. 3, pp. 34–39, Sept 2012.

[15] K. Zheng, F. Hu, W. Wang, W. Xiang, and M. Dohler, "Radio resource allocation in lte-advanced cellular networks with m2m communications," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 184–192, July 2012.

[16] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, and M. Guizani, "Home m2m networks: Architectures, standards, and qos improvement," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 44–52, April 2011.

[17] C. Y. Ho and C. Y. Huang, "Energy-saving massive access control and resource allocation schemes for m2m communications in ofdma cellular networks," *IEEE Wireless Communications Letters*, vol. 1, no. 3, pp. 209–212, June 2012.

[18] G. Wang, X. Zhong, S. Mei, and J. Wang, "An adaptive medium access control mechanism for cellular based machine to machine (m2m) communication," in *2010 IEEE International Conference on Wireless Information Technology and Systems*, Aug 2010, pp. 1–4.

[19] "3GPP," http://www.3gpp.org/news-events/3gpp-news/1426-global-initiative-for-m2m-standardization, 2016.

[20] M. Chen, J. Wan, S. Gonzalez, X. Liao, and V. C. M. Leung, "A survey of recent developments in home m2m networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 98–114, First 2014.

[21] T. Taleb and A. Kunz, "Machine type communications in 3gpp networks: potential, challenges, and solutions," *IEEE Communications Magazine*, vol. 50, no. 3, pp. 178–184, March 2012.

[22] S. Y. Lien and K. C. Chen, "Massive access management for qos guarantees in 3gpp machine-to-machine communications," *IEEE Communications Letters*, vol. 15, no. 3, pp. 311–313, March 2011.

[23] J. Matamoros and C. Antn-Haro, "Data aggregation schemes for machine-to-machine gateways: Interplay with mac protocols," in *2012 Future Network Mobile Summit (FutureNetw)*, July 2012, pp. 1–8.

[24] G. C. Madueo, C. Stefanovic, and P. Popovski, "Reliable reporting for massive m2m communications with periodic resource pooling," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 429–432, Aug 2014.

[25] Y. Liu, C. Yuen, X. Cao, N. U. Hassan, and J. Chen, "Design of a scalable hybrid mac protocol for heterogeneous m2m networks," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 99–111, Feb 2014.

[26] S. I. Sou and S. M. Wang, "Performance improvements of batch data model for machine-to-machine communications," *IEEE Communications Letters*, vol. 18, no. 10, pp. 1775–1778, Oct 2014.

[27] D. T. Wiriaatmadja and K. W. Choi, "Hybrid random access and data transmission protocol for machine-to-machine communica-

tions in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 33–46, Jan 2015.

[28] A. Rajandekar and B. Sikdar, "A survey of mac layer issues and protocols for machine-to-machine communications," *IEEE Internet of Things Journal*, vol. 2, no. 2, pp. 175–186, April 2015.

[29] S. Y. Lien, T. H. Liau, C. Y. Kao, and K. C. Chen, "Cooperative access class barring for machine-to-machine communications," *IEEE Transactions on Wireless Communications*, vol. 11, no. 1, pp. 27–32, January 2012.

[30] S. H. Wang, H. J. Su, H. Y. Hsieh, S. p. Yeh, and M. Ho, "Random access design for clustered wireless machine to machine networks," in *2013 First International Black Sea Conference on Communications and Networking (BlackSeaCom)*, July 2013, pp. 107–111.

[31] I. Stojmenovic, "Machine-to-machine communications with in-network data aggregation, processing, and actuation for large-scale cyber-physical systems," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 122–128, April 2014.

[32] J. Zhang, Y. Xiang, Y. Wang, W. Zhou, Y. Xiang, and Y. Guan, "Network traffic classification using correlation information," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 104–117, Jan 2013.

[33] C.-S. Chang, "Stability, queue length, and delay of deterministic and stochastic queueing networks," *IEEE Transactions on Automatic Control*, vol. 39, no. 5, pp. 913–931, May 1994.

[34] R. L. Cruz, "A calculus for network delay. i. network elements in isolation," *IEEE Transactions on Information Theory*, vol. 37, no. 1, pp. 114–131, Jan 1991.

[35] R. Agrawal, R. L. Cruz, C. Okino, and R. Rajan, "Performance bounds for flow control protocols," *IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 310–323, Jun 1999.

[36] J.-Y. L. Boudec and P. Thiran, "Network calculus: A theory of deterministic queuing systems for the internet," in *tutorial*, ser. LNCS, G. Goos, J. Hartmanis, and J. van Leeuwen, Eds. Springer, 2001, vol. 2050.

[37] D. Stiliadis and A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms," *IEEE/ACM Transactions on Networking*, vol. 6, no. 5, pp. 611–624, Oct 1998.

[38] "OMNET++," http://omnetpp.org/, 2016.