QoS Constrained Semi-Persistent Scheduling of Machine Type Communications in Cellular Networks

by

Recep Gül

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ
UNIVERSITY

2016

QoS Constrained Semi-Persistent Scheduling of Machine Type Communications in Cellular Networks

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Recep Gül

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Assoc. Prof. Sinem Çöleri Ergen (Advisor)

Assoc. Prof. Öznur Özkasap

Asst. Prof. Yalçın Şadi

Date: ______________________

To Kloten

Abstract

The dramatic growth of machine-to-machine (M2M) communication in cellular networks brings the challenge of satisfying the Quality of Service (QoS) requirements of a large number of machine-type communication (MTC) devices with limited radio resources. In this thesis, an optimization framework for semi-persistent scheduling of M2M transmissions based on the exploitation of the periodicity of their transmissions with the goal of reducing the overhead of connection and scheduling is proposed. Semi-persistent scheduling allows decreasing the overhead of the signaling required for connection initiation and scheduling.The goal of the optimization problem is to minimize the number of frequency bands used by MTC devices to allow fair resource allocation of newly joining machines and human-to-human communications. The constraints of the problem are jitter and periodicity requirements of MTC devices. The optimization problem is proved to be NP-hard. We first propose an efficient fathoming-based smart enumeration algorithm to find the optimal solution. Then, we propose a polynomial-time heuristic algorithm employing a fixed priority assignment according to QoS characteristics of devices. We prove that this heuristic algorithm provide an asymptotic approximation ratio of 2.33 to the optimal solution for the case where the jitter tolerance of devices are equal to their periods. We also propose a call admission control scheme to dynamically manage the scheduling of newly joining devices. Through extensive simulations, we demonstrate that the proposed algorithm performs better than the existing algorithms by providing a greater efficiency in frequency band usage and satisfying QoS requirements of MTC devices.

ÖZETÇE

Makineler arası iletişimin hücresel ağlarda hızlı bir şekilde büyümesi, kısıtlı spektrum kaynaklarıyla bu makinelerin servis kalitesinin sağlanmasını önemli ölçüde zorlaştırmaktadır. Bu tezde, makineler arası iletişimin bağlanma ve çizelgeleme ek yükünün azaltılması amacıyla, makinelerin veri iletiminin periyodikliğini kullanarak, yarı kalıcı çizelgeleme için bir eniyileme çerçevesi önerilmiştir. Yarı kalıcı çizelgeleme, makinelerin ağa bağlanması ve cizelgelenmesi için gereken işaretleşme ek yükünün azaltılmasını sağlar. Eniyileme problemi, frekans kaynaklarının ağa yeni katılan makineler arası iletişim cihazları ve insandan insana iletişim cihazları arasında adil tahsis edilmesi için, makinelerin kullandığı frekans band sayısını minimize etmeyi amaçlar. Eniyileme probleminin kısıtlamaları makine arası iletişim cihazlarının periyodiklik ve zamanda sapma gereksinimlerinden oluşmaktadır. Problemin NP-Zor olduğu kanıtlanmıştır. Öncelikle, problemin en iyi çözümünü elde eden budamaya dayalı akıllı bir sayım metodu önerilmiştir. Daha sonra problemi çok terimli zamanda yaklaşık olarak çözen sezgisel bir algoritma önerilmiştir. Makinelerin zamanda sapma toleranslarının periyotlarına eşit olduğu durumda, önerilen sezgisel algoritmanın en iyi çözüme asimptotik yaklaşım oranının 2.33 olduğu kanıtlanmıştır. Ayrıca, ağa yeni katılan makinelerin çizelgelerinin dinamik olarak yönetilmesi için bir çağrı kabul denetimi mekanizması önerilmiştir. Tezde önerilen algoritmaların, daha önce ileri sürülen algoritmalara kıyasla, frekans bandını daha verimli kullandığı ve makinelerin servis kalitesi gerekliliklerini sağladığı kapsamlı simülasyonlarla gösterilmiştir.

Acknowledgments

I would like to thank my advisor Assoc. Prof. Sinem Çöleri Ergen for her guidance, support and relentless patience. I would also like to thank committe members Asst. Prof. Yalçın Şadi and Assoc. Prof. Öznur Özkasap for their service on the committee.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

M2M communications is one of the main drivers for evolving 5G technology [1]. M2M services have a great variety of applications such as smart grid, healthcare, home automation and vehicular communications [2] . By 2019, more than % 40 of all connected devices are projected to be MTC devices [3]. Thus, providing a native support in 5G for fast growing M2M services is of paramount importance. However, supporting a massive number of MTC devices is very challenging due to the problem of allocating radio resources to a large number of devices with diverse QoS requirements in the same network. Furthermore, M2M communications have distinct data transmission characteristics compared to human-to-human (H2H) communications, including more frequent access to network, small amount of data transmission, and mostly periodic traffic generation [4]. This renders radio resource allocation solutions proposed for H2H communications inefficient for M2M communications.

Radio resource allocation problem consists of scheduling signalling and data resource allocation for MTC devices. Signalling resource allocation problem refers to allocation of radio resources to initiate the connection between devices and the base station as well as to send control and scheduling signals from base station to devices. Since signalling resources are not used for sending data, excessive signalling decreases the system performance by consuming limited radio resources. Data resource allocation problem refers to allocation of radio resources for devices to send data. In order to solve signalling resource allocation problem for M2M communications, various solutions are proposed in LTE. Such solutions involve backoff methods that spread access attempts of devices in time [5] [6] [7], various Access Class Barring (ACB) methods in which the access probability of a device is determined by a network parameter [8] [9] [10]. Furthermore, piggybacking method in which devices send data during random access procedure are evaluated [11] [12]. However, these methods have major limitations: they do not provide QoS guarantees as they employ a random access procedure and they do not exploit the periodicity of M2M communications.

For data resource allocation, a fully dynamic, QoS aware resource allocation at each Transmission Time Interval (TTI) would be the optimal resource allocation method [13]. In [14] and [15], dynamic joint resource allocation schemes for signalling and data communications are proposed. However, such dynamic resource allocation schemes per TTI causes a large amount of signaling overhead [16], given a very large number of MTC devices with frequent access to the network. Furthermore, they do not exploit periodicity of MTC devices. In LTE, this signalling overhead problem has been intensively studied for Voice over Internet Protocol (VoIP) which has similar characteristics with certain M2M applications: frequent, periodic data and small packet size. Various persistent scheduling schemes that exploit periodicity of devices [17] has been proposed for VoIP communications in LTE [18], [19], [20]. In persistent scheduling schemes, radio resources are allocated to devices for a longer period, instead of each Transmission Time Interval (TTI). Hence, signalling resources are employed more efficiently. Thus, persistent scheduling schemes are particularly suited to periodic M2M communications [13]. However, these persistent scheduling schemes developed for VoIP cannot be used for M2M communications. That is because M2M applications have very diverse traffic characteristics (period, jitter tolerance etc.) whereas VoIP is an application with very limited diversity. Thus, we need persistent scheduling schemes that can address QoS requirements of diverse M2M applications.

The persistent resource allocation for data transmission of MTC devices on the same cellular network are first proposed in [21] and [22] where the algorithms allocating fixed access grants over periodic time-intervals to MTC clusters considering a deterministic bound on the experienced jitter are developed. Here, an MTC cluster is defined as the set of devices with the same QoS characteristics with the goal of alleviating the complexity of managing a large number of devices by classifying them into groups [22]. These studies are followed by [23] and [24] where a statistical QoS metric, namely the delay threshold violation probability is deployed to tune the access grant time interval (AGTI) period for the clusters achieving less resource usage for M2M traffic. However, there are major limitations of these works. They use the entire bandwidth while reserving an AGTI to a cluster without considering its adverse effect on H2H communications. Furthermore, allocation of the entire bandwidth to a cluster implies a uniform schedule over all frequency bands since each cluster is present at each frequency band. However, such uniform scheduling imposes the same QoS requirements at each frequency band, thus preventing MTC devices with stringent QoS requirements to be admitted to the network.

Radio resource allocation schemes should address the effective partition of resources between M2M and H2H communications so that QoS requirements of both can be satisfied. One approach is to allocate resources for H2H communications first and use the remaining resources for MTC devices [25]. However, this leads to resource starvation for MTC devices whenever H2H communications is allocated excessive radio resources due to heavy H2H traffic [26]. On the other hand, prioritizing MTC devices will degrade H2H communications due to resource starvation and excessive delays caused by a massive number of MTC devices connected to the network under heavy traffic load. A possible solution for this problem is to allocate a constant part of bandwidth for H2H communications and the rest for MTC devices [23], thus separating the M2M and H2H resources. Such a separation alleviates both resource starvation problem and excessive delays that MTC and H2H devices would cause to each other, should they were allocated on the same frequency bands. Following this approach, we assume MTC and H2H devices are to be allocated on different frequency bands.

We propose a new approach that minimizes the frequency spectrum used by M2M traffic and allows admission of MTC devices with more stringent QoS requirements. Frequency band minimization for M2M is introduced for the first time in the literature. This minimization is based on the observation that MTC and H2H devices must be allocated on separate frequency bands, and thus an efficient radio resource usage of MTC devices is also crucial for the performance of H2H communications. In addition to solving resource starvation problem, our proposed radio resource allocation scheme also exploits the traffic characteristics of MTC devices to satisfy QoS requirements of MTC devices and reduces the signalling cost of M2M communications. The original contributions of the thesis are as follows:

- We formulate the radio resource allocation problem with the objective of minimizing the number of frequency bands used by MTC devices while stringent timing requirements of MTC devices are satisfied. We prove that the optimization problem is NP-hard. The solution of the NP-hard problem requires exponential runtime in the number of MTC devices.

- We first propose a fathoming-based smart enumeration algorithm to obtain the optimal solution. The algorithm is based on the depth-first search method for branch and bound technique. Even though the algorithm provides an efficient method to compute the optimal solution, it still requires an exponential runtime in the number of MTC devices. Therefore, a heuristic algorithm is needed for an efficient and fast solution of the problem.

- We propose a heuristic algorithm for radio resource allocation problem for MTC devices. The main feature of the proposed algorithm is to employ a priority assignment based on periods of devices and allocate as many devices as possible to a frequency band while the jitter requirements of each device is satisfied.

- We provide the worst case performance of the proposed algorithm with respect to optimal solution under certain conditions. Thus, a performance guarantee for the proposed the algorithm is provided, showing the robustness of the algorithm with respect to changing MTC device combinations.

- We further propose a call admission mechanism to effectively manage the admission of new devices. Our call admission mechanism is based on the principle that the frequency band should be used efficiently to serve as many devices as possible.

- Through extensive simulations we show that our proposed algorithm uses frequency band much more efficiently with superior schedulability performance than previous algorithms.

The rest of the thesis is organized as follows. Section 2 describes the system model and assumptions. Section 3 describes the scheduling framework. Section 4 describes the optimization problem and the fathoming-based smart enumeration algorithm. Section 5 presents the proposed radio resource allocation algorithm. Section 6 provides the call admission control scheme. In Section 7, simulation results are provided. Finally, Section 8 concludes the thesis.

Chapter 2

System Model and Assumptions

The system model and assumptions are detailed as follows.

1. We consider a cellular network with a base station which serves a large number of MTC devices with diverse traffic characteristics in addition to H2H devices.

2. Each device is allocated time-frequency resource elements called *Resource Blocks* (RBs). In LTE, a resource block is a time-frequency unit with 0.5 ms length in time and 180 kHz length in frequency. The groups of resource blocks determines the length of minimum scheduling unit [27], thus providing a time granularity for scheduling. RB-based granularity is expected to be preserved in 5G cellular networks, even though the size of an RB may change [1]. For periodic data generating devices, multiple RBs may be allocated in a period but these RBs do not have to be allocated consecutively.

3. Most M2M applications involve time-triggered devices generating periodic data [4] [13]; in such applications as smart grid, e-health applications [28], intelligent transportation and industrial supply systems [4]. Time-triggered MTC devices will be allocated resources with a persistent scheduling algorithm. There may be further event-triggered machines and H2H devices generating data at random intervals. For these devices, base station employs dynamic scheduling to allocate radio resources.

4. We define Unit Frequency Band (UFB) as the frequency band of 1 RB width. Each

device is allocated to one UFB; i.e. once a device is allocated to a UFB, all packets of that device will be transmitted on that particular UFB. This partitioned scheduling is particularly useful as it provides a lower scheduling overhead since it does not allow packets of the same device to migrate to the different bands [29].

5. Many M2M application require deterministic QoS requirements to be satisfied. For event-triggered machines, QoS requirement is captured by delay, whereas for time-triggered periodic data generating machines the QoS is captured by jitter. Jitter is defined as the time difference between two successive packet arrivals and departures [21]. Periodic MTC devices have a maximum allowable jitter that we call *jitter tolerance*. Satisfying deterministic QoS requirements is critical in many applications, especially in safety critical operations such as navigational data communications [30] or health care applications[31]. Furthermore, many real-time control systems also require hard QoS requirements to be satisfied [32].

6. Time-triggered MTC devices are assigned priorities in the decreasing order of their periods; i.e. a lower period implies a higher priority. In the case of equality of periods, devices with lower jitter values are prioritized. If jitter values are also equal, then devices with higher transmission times are prioritized. If transmission times are also equal, they are randomly assigned priorities.

7. Time-triggered MTC devices are given priority within a certain number of UFBs, denoted by k_{max}. k_{max} may be determined according to traffic density, number of devices or channel condition. Once time-triggered MTC devices are allocated within k_{max} bands, event-triggered MTC and H2H devices can be scheduled if resources are available. The scheduling outside k_{max} bands where time-triggered MTC devices are not allocated is out of scope of this thesis.

8. The semi-persistent schedule is updated regularly with period P. The period P may be determined according to traffic density of time-triggered MTC devices and ratio of UFBs used. With low traffic density and low bandwidth usage, the schedule can

be updated over longer intervals; i.e. P can be larger. On the other hand, if traffic density for time-triggered M2M is high and a large number of UFBs are used, then the schedule can be updated on shorter intervals.

Chapter 3

Scheduling Framework

This section provides an overview of our proposed scheduling framework; as summarized in Fig 3.1. The base station uses dynamic scheduling to allocate event-based MTC and H2H devices; whereas it employs a semi-persistent scheduling scheme to allocate time-triggered MTC devices. Semi-persistent scheduling of the transmission of time-triggered MTC machines allows the reduction of both signalling and scheduling overhead. Since the data transmission times are preknown, the signalling required to initiate connection and request resources for the transmission of data is eliminated. Semi-persistent scheduling also cancels the signalling of the announcement of the schedules in the downlink. While the base station needs to update the dynamic schedule at each TTI, semi-persistent schedule persists for a longer time interval.

We construct a semi persistent schedule that minimizes the number of UFBs used by the time-triggered machines generating data periodically at predetermined rates while satisfying strict jitter tolerances of these devices. Minimization of the number of UFBs is motivated by the spectrum scarcity, separation of resources allocated to machines and humans, and presence of newly joining time-triggered machines, event-driven devices and human-to-human communication with various QoS requirements as follows:

- Massive number of MTC devices consume scarce radio resources that are already strained by H2H communications [33]. This scarcity of available spectrum is a major problem for M2M communications [34], and it brings the challenge of sustaining acceptable QoS with scarce resources [35]. Thus, M2M communications must be

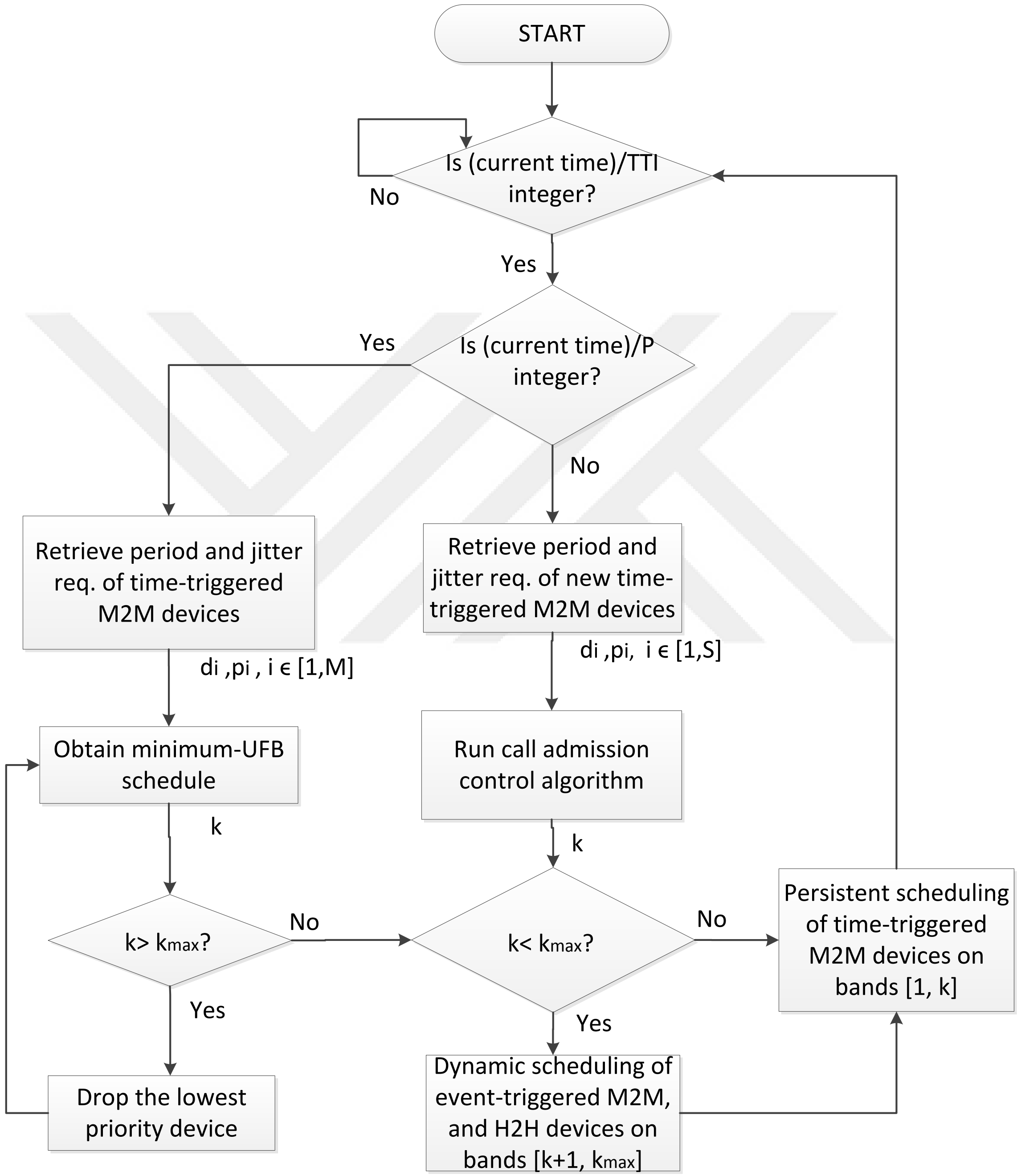

Figure 3.1: Scheduling Framework

designed to efficiently utilize available spectrum resources [36]. This, in turn, can be achieved by minimizing number of UFBs allocated for MTC devices.

- As mentioned earlier, frequency bands reserved for M2M and H2H communications are separated. Many M2M applications have strict timing requirements; such as medical applications, assisted living, industrial control and navigational data communications [37] [38] [30]. If H2H and M2M bands are not separated and H2H communications are given priority, then these delay sensitive M2M applications suffer excessive delays and resource starvation due to H2H devices. Similarly, many H2H applications such as online gaming, internet browsing, video streaming, and VoIP have strict latency requirements [39][19]. If M2M communications are given priority, these delay-sensitive H2H applications may experience performance degradation due to massive number of MTC devices allocated to the same bands. An efficient solution to this problem is to partition the spectrum between MTC and H2H devices. In that case, the number of frequency bands allocated to time-triggered MTC devices must be minimized in order to serve more MTC devices and provide more resources for H2H communications.

- We need to further consider that new devices with strict delay requirements may join the network. Due to strict delay requirements, there must be sufficient radio resources immediately available for such devices. This further requires an efficient usage of radio resources by MTC devices so that some idle resources are available for such delay sensitive devices.

This minimum-UFB semi-persistent schedule will need to be updated regularly as new time-triggered MTC devices join and leave the network. If the number of UFBs needed to schedule time-triggered MTC devices is more than k_{max}, then the lowest priority devices are dropped until the remaining devices can be scheduled on k_{max} bands.

At each TTI between two schedule update points, new H2H, time-triggered and event triggered MTC devices may arrive to the network. Newly joining time-triggered MTC devices are given priority within the frequency bands allocated for time-triggered MTC communications. Within these bands, we need to schedule newly joining MTC devices while using frequency band efficiently. For that purpose, we propose a call admission control mechanism. This mechanism is based on the idea that if any new device can be scheduled without violating QoS requirements of existing devices within k_{max}, then it must be admitted. Moreover, at any TTI, if there are idle bands within k_{max} after time-triggered MTC devices are scheduled; then, the base station may use these idle bands to dynamically schedule event-triggered MTC and H2H devices. In order to schedule these devices, the base station may use any previously proposed dynamic scheduling algorithm for cellular networks such as [40].

The minimum-UFB problem is formulated in Section 4. In section 5 an heuristic algorithm for minimum-UFB is proposed. The call admission control scheme is given in detail in Section 6.

Chapter 4

Description Of The Optimization Problem

In this section, we formulate the optimization problem. The objective of the problem is to minimize the number of UFBs occupied by time-triggered MTC devices. The constraints of the problem are period and jitter tolerances of devices. We prove that the optimization problem is NP-hard. Then, we propose a fathoming-based smart enumeration (FBSE) algorithm to obtain the optimal solution.

4.1 QoS Constraints

The period and jitter tolerances of time-triggered MTC devices are given. In order to give QoS guarantees, we must ensure that jitter tolerances are never violated. This, in turn, requires that jitter tolerances must be satisfied in the worst case for each device. For any device, the worst case occurs when the device wants to transmit a packet at the same time with all higher priority devices. In this case, the device has to wait for all higher priority devices to transmit their data. This worst case bound, called jitter bound, is proposed in [22]. The jitter bound provides a tight bound to experienced jitter, serving as a computationally simple sufficient condition for satisfying jitter tolerances.

In the following, we provide a simple extension of jitter bound provided in [21] for variable transmission times with the following variables: There are N time-triggered devices on the same band. δ_i^* is the jittter bound, and τ_i is the transmission time of device i. p_i is the period and d_i is jitter tolerance of device i. Devices are ordered according to the priority assignment; i.e. if device i is prior to device l, then $i < l$.

$$\delta_i^* = \tau_i + \sum_{l=1}^{i-1} \lceil \frac{pi}{p_l} \rceil \tau_l \le d_i \qquad , \tag{4.1}$$

for $i = 1, ..N.$

4.2 Formulation of Optimization Problem

The optimization problem is formulated as follows:

minimize

$$\sum_{k=1}^{K} y_k \tag{4.2}$$

subject to

$$\sum_{k=1}^{K} x_{ik} = 1, \quad \text{for } i = 1, ..N \tag{4.3}$$

$$\sum_{i=1}^{N} x_{ik} \le N y_k, \quad \text{for } k = 1, ..K \tag{4.4}$$

$$\tau i + \sum_{l=1}^{i-1} \lceil \frac{p_i}{p_l} \rceil \tau_l x_{lk} \le d_i + (1 - x_{ik}) T_i \tag{4.5}$$

for $i = 1, ..N$ for $k = 1, ..K$

variables

$$y_k \in \{0, 1\}, \; x_{ik} \in \{0, 1\}, \quad \text{for } i = 1, ..N, \quad k = 1, ..K \tag{4.6}$$

where $T_i = \tau_i + \sum_{l=1}^{M} \lceil \frac{p_i}{p_l} \rceil \tau_l - d_i$; i.e. T_i is a large integer to ensure that (4.5) is always satisfied when $x_{ik} = 0$. y_k is a binary variable taking the value 1 if any device is allocated to UFB k, and 0 otherwise. Similarly, x_{ik} is a binary variable taking the value 1 if the device i is allocated to band k, and 0 otherwise. The objective is to minimize the number of UFB's used. The equation (4.3) states that each device must be scheduled to one UFB. Equation (4.4) represents the relation between two variables: if a band is not used, there can be no device on this band. Similarly, if a device is on a band, then this band must be used. Finally, the equation (4.5) provides jitter bound constraint. Note the tradeoff between jitter tolerance of devices and the number of frequency bands used for allocating devices. As the jitter tolerance decreases, fewer x_{lk} terms on the left-hand side of (4.5) can be nonzero. Therefore, we can allocate fewer devices on the same band (band k), implying a higher number of bands needed for all devices. The problem is a binary linear programming (BLP) problem.

4.3 NP-Hardness of Optimization Problem

Theorem 1 *The optimization problem described in Section III-C is NP-Hard.*

Proof: *Consider a problem instance where $d_i = p_i$ and transmission times $\tau_i = x_i$ where $\frac{B}{4} < x_i < \frac{B}{2}$ for all $i \in [1, 3k]$ and $B \in \mathbb{Z}^+$. We will show that this special instance of our problem is equivalent to 3-Partition Problem. 3-Partition Problem can be described as follows: Consider a set of $3k$ positive integers, $S = \{x_1, x_2, .., x_{3k}\}$ and $B \in \mathbb{Z}^+$ where $\frac{B}{4} < x_i < \frac{B}{2}$ for all $i \in [1, 3k]$ and $\sum_{i=1}^{3k} x_i = kB$. Can set S be divided into k disjoint sets $S_1, .., S_k$ such that for each $m \in [1, k]$, $\sum_{x_i \in S_m} x_i = B$ is satisfied? Note that $\frac{B}{4} < x_i < \frac{B}{2}, i \in [1, 3k]$ constraint forces each disjoint set to have exactly 3 elements. With that description, we take the 3-Partition Problem, $S = \{x_1, ..., x_{3k}\}$ and $B \in \mathbb{Z}^+$, and construct the following system: there are $3k$ devices to be allocated into k bands, where $d_i = p_i = B$ and $\tau_i = x_i$ for all $i \in [1, 3k]$ (Note that RB's of a device does not have to be allocated consecutively). At each band, there are 3 devices and for these devices to be schedulable it is necessary and sufficient that the sum of their transmission times is not*

larger than the period; i.e. $\sum_{x_i \in S_m} x_i \leq B$ *for* $m \in [1, k]$*. Summing over all* $m \in [1, k]$*, we obtain* $\sum_{i=1}^{3k} x_i = kB \leq kB$*. Thus, our constructed system is schedulable if and only if the inequality described is satisfied, which in turn is satisfied only in equality case. Then, the constructed system is schedulable if and only if at each band* k*, the sum of transmisson times are equal to* B*; and this problem has a solution if and only if given instance of 3-Partition Problem has a solution. Since this construction is carried out in polynomial time, the problem of whether a set of devices* S *with integer transmission times is schedulable on* k *bands is NP-hard. Obviously, if the problem of finding minimum number of bands that can schedule the set* S *can be solved in polynomial time, then the problem whether the set* S *can be scheduled on* k *bands can be solved in polynomial time as well, for any* $k \in \mathbb{Z}^+$*. Thus,the problem of finding minimum number of bands to schedule set* S *is also NP-hard.*
□

4.4 Fathoming-Based Smart Enumeration Algorithm

In this section, we describe an algorithm to find the optimal solution based on depth-first search for branch and bound method. Since the optimization problem is NP-hard, the EPDS algorithm has exponential runtime with respect to number of devices. The computational complexity of the algorithm is $\mathcal{O}(N^N)$ where N is number of devices. The basic structure for branch-and-bound tree is as follows: At each level $j \in [1, N]$, each kth children node of the parent node (which is at level $j - 1$) represents the allocation of device j on band k. For example, each parent at level 5 branches into N children nodes where each children represent the allocation of device 6 onto a particular band. In Fig. 4.1, the algorithm is described in more detail with an example. We employ a depth-first search, which implies that we move down to descendants of a branch as deep as possible before doing any back-tracking to parent nodes. We abort the search prematurely on a node if and only if at least one of the following conditions are satisfied:

1. The allocation of the device on the particular band that the node represents violates the jitter bound for the device.

2. The allocation of the device on the particular band that the node represents results in a worse solution then the best feasible solution already obtained.

For both conditions, note that descendant nodes do not change allocation of devices represented by the parent node. Therefore, if the jitter bound is violated for a device on a particular band, the violation cannot be reverted on descendant nodes. Similarly, if the allocation of the device requires more bands than the best solution we obtained so far, then the search through descendant nodes will not decrease the number of required bands.

Algorithm 1 Fathoming-Based Smart Enumeration Algorithm

initialize variables: $bestFeasible$=N, F={all nodes}, CN= zeros(N,1), $optimalAllocation$=zeros(N,1);
while $F \neq \varnothing$ **do**
    if $R_{CN} = \varnothing$ **and** CN is not at level N **then**
        $F \leftarrow F \setminus CN$;
        $CN \leftarrow$ Parent of CN;
    else if CN satisfies any pruning condition stated above **then**
        $F \leftarrow F \setminus (R^*_{CN} \cup CN)$;
        $CN \leftarrow$ Parent of CN;
    else if CN is not at level N **then**
        $CN \leftarrow$ Child in R_{CN} with minimum band value;
    else if max $(CN) \leq bestFeasible$ **and** jitter bound is not violated **then**
        $bestFeasible$=max(CN);
        $optimalAllocation = CN$;
        $CN \leftarrow$ Parent of CN;
        $F \leftarrow F \setminus (R_{CN} \cup CN)$;
        $CN \leftarrow$ Parent of CN;
    else
        $F \leftarrow F \setminus CN$;
        $CN \leftarrow$ Parent of CN;
    end if
end while
$optimumNumberOfBands = bestFeasible$;

The FBSE algorithm starts the search from the 'root' node. The 'root' node is initial node from which we start branching, and it is the parent of the node that represents the allocation of the highest priority device. As we traverse down a branch, we store the the band allocated for each device in each Current Node (CN), which is initialized to a zero vector. We store the best feasible solution in $bestFeasible$ variable. $bestFeasible$ is initialized to

N since in the worst case all devices are allocated on a different band. The band allocation vector corresponding to optimal solution is stored in $optimalAllocation$ vector. Note that there may be multiple $optimalAllocation$ vectors satisfying the optimal band number. Our algorithm stores only one of the $optimalAllocation$ vectors corresponding to optimal band number. F is the set of unexplored nodes and is initialized to all nodes. R_{CN} refers to unexplored children of node CN. R^*_{CN} is all unexplored descendants of node CN. The algorithm will terminate when all nodes are explored (Line 2). If the current node does not have any unexplored children and it is not at the level N; i.e. if it is not at the bottom of the search tree, then we mark the node explored and return to parent node (Line 3-5). If current node has children to be explored but it satisfies the pruning conditions, then we mark the node and all its descendants explored and return to parent node (Line 6-8). If current node has children to be explored, and if it does not satisfy pruning conditions, then we move to the unexplored child node with minimum band value (Line 9-10). If the current node is at the level N, and if it provides a better feasible solution than $bestFeasible$, then we update $bestFeasible$ and store the CN in the $optimalAllocation$ vector (Line 11-13). Since from each parent node, we move to the unexplored child node with the minimum band value, all other unexplored nodes of the same parent will provide a worse solution. So, we do not need to visit the 'sibling' nodes. Thus, we first move to the parent node and then mark the parent and the all unexplored children explored (Line 14-15). Now that we are done with the parent and the children, we move to parent node (line 16). If we are at the level N, but the current node provides a worse solution than $bestFeasible$ or if it violates the jitter bound, then we mark the current node explored and move to the parent node (Line 17-19). Finally, the optimum number of bands is equal to the best feasible value obtained during the search (Line 22).

We explain the FBSE algorithm with an example in Fig. 4.1. Note that since the number of devices is 3 in this example, $bestFeasible$ is initialized to 3. Assume that the red nodes violate jitter bound, and that green and white nodes do not violate jitter bound. We start with the root node, where no device is allocated yet. We then move to the first node, which allocates the first device to band 1. Since this node does not satisfy pruning conditions, we will move to the child node with the minimum band number. This child node allocates the second device on band 1, whereas the other child allocates the second device on band 2. Since this node also does not satisfy any pruning condition, we move to the unexplored child node with minimum band number, (1,1,1). However, this node does not satisfy jitter conditions. Hence, we move back to parent node, (1,1,0). The parent node picks the unexplored child node with the smallest band number (1,1,2). However, this node also does not satisfy jitter conditions. Finally, the parent node picks the (1,1,3) node. This node provides a feasible solution and it is not worse than $bestFeasible$. Then, we update the $optimalAllocation$ algorithm to (1,1,3). Then, we move to parent node, (1,1,0). The parent node has still an unexplored child remaining, (1,2,0); thus we move to that node. The node (1,2,0) has three children and we move to the child with the smallest band number. The node (1,2,1) provides a feasible solution. This solution is better than the $bestFeasible$ whose value is 3. Then, we update the $bestFeasible$ to be max(1,2,1)=2. We also update $optimalAllocation$ to (1,2,1). Now that we obtained the $bestFeasible$, we do not need to visit the sibling nodes, since they will provide worse solutions. So, we move to the parent node (1,2,0), and mark it and all its unexplored children explored, before moving back to its parent node (1,0,0). Since node (1,0,0) has no unexplored children, we move to the parent node, the root node, at which point the algorithm terminates as we explored all the nodes. The optimum solution is therefore 2, and the optimal allocation vector is (1,2,1) since this provides the best feasible solution obtained throughout the search.

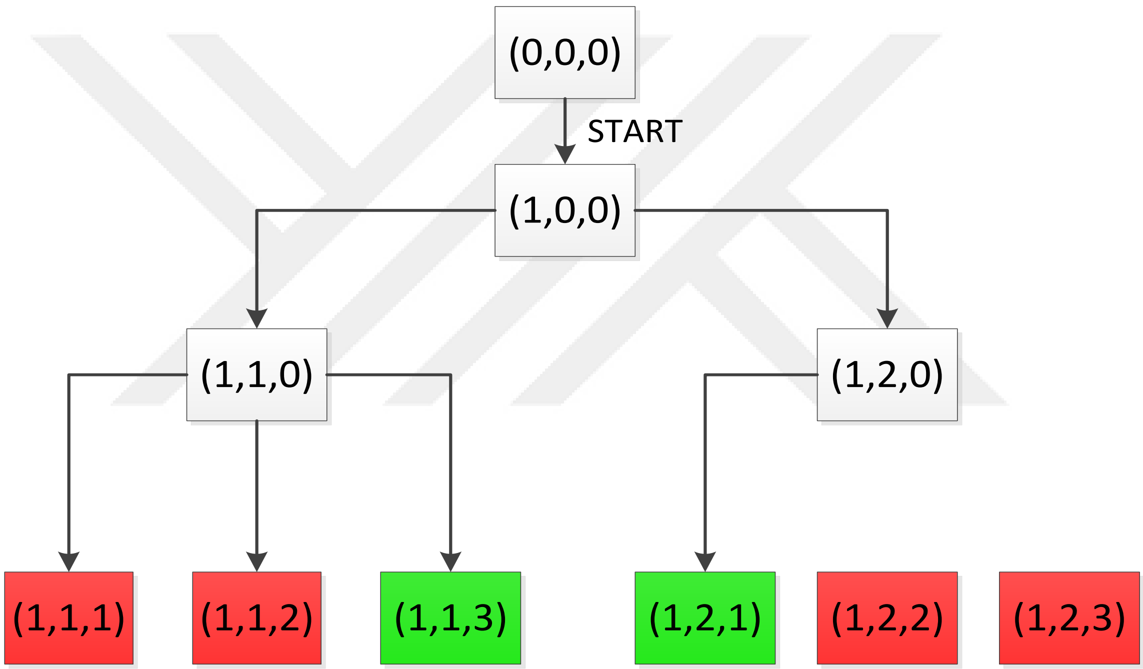

Figure 4.1: Fathoming-Based Smart Enumeration

Chapter 5

Minimum Frequency First-Fit Allocation Algorithm

There is already a huge burden on the base stations due to the excess computational complexity of the algoritms for H2H communications and the massive number of MTC devices will put a further strain on LTE/5G stations. Therefore, the radio resource management schemes for MTC devices should be computationally simple besides being effective [22]. Our heuristic algorithm is closely related to FBSE algorithm. The heuristic algorithm is equivalent to a modified-FBSE algorithm in which instead of exploring all the tree, we always move to the feasible child node with the minimum band number. We further group devices with the same traffic and QoS characteristics under clusters, and the benefit of clustering is to reduce the complexity of handling a large number of devices by managing them as a group of cluster, further improving the efficiency of our heuristic algorithm.

5.1 Algorithm Description

We propose the Minimum Frequency First-Fit Allocation (MFFFA) Algorithm as described next. MTC devices are clustered into M clusters based on packet arrival period, maximum allowable jitter and transmission time. $\mathbf{N}_i$ correspond to number of unallocated devices from cluster i. $crossJitter$ is jitter the device experiences due to devices in higher priority clusters. d_i stands for jitter tolerance of devices in cluster i. $\mathbf{UFB}^k$ is a $M-$dimensional vector storing the number of devices from each cluster in band k. $\mathbf{UFB}^k$ is initialized to a zero vector (Line 7). The algorithm allocates the devices starting from the highest-priority cluster (Line 6). First, $crossJitter$ on each cluster i from higher priority

Algorithm 2 Minimum Frequency First-Fit Allocation (MFFFA) Algorithm

1: **Input**: $d_i, p_i, \tau_i, \mathbf{N}_i$ for $i \in [1, M]$
2: **Output**: n, UFB^k, for $k \in [1, K]$
3: $k = 1, n = 1$;
4: **while** $\sum_{i\in[1,M]} \mathbf{N}_i \neq 0$ **do**
5:     $\mathbf{UFB}^k = zeros(1, M)$;
6:     **for** $i = 1 : M$ **do**
7:         $crossJitter = 0$;
8:         **for** $j = 1 : i$ **do**
9:             $crossJitter = crossJitter + \mathbf{UFB}_j^k * \lceil \frac{p_i}{p_j} \rceil * \tau_j$;
10:         **end for**
11:         $remJitter = d_i - crossJitter$;
12:         **if** $remJitter > 0$ **then**
13:             $\mathbf{UFB}_i^k = \min(\mathbf{N}_i, \lfloor \frac{remJitter}{\tau_i} \rfloor)$;
14:         **end if**
15:     **end for**
16:     $\mathbf{N} = \mathbf{N} - \mathbf{UFB}^k$;
17:     $k + +$;
18:     $n = n + 1$;
19: **end while**

clusters is calculated (Lines 7-9). The $remJitter$ refers to the difference between jitter tolerance and experienced jitter and it is used to determine how many devices from cluster i can be allocated on band k. If the number of devices from cluster i that can be allocated on a band according to remaining jitter and transmission time is larger than number of remaining devices from cluster i, then all remaining devices in cluster i are allocated to that particular band(Lines 11-13). Then, the set of remaining devices and number of UFBs used, n, is updated (Line 16-18).

5.2 Algorithm Description: An Example

In Fig. 5.1, we describe the workings of our algorithm through an example. The number of devices in each cluster and their traffic characteristics are described in Table 5.1. Clusters are first assigned priorities where lower period implies a higher priority. Accordingly, cluster A has the highest priority and cluster C has the lowest priority. Then, we start allocating devices from cluster A. Since there is no any higher priority cluster, there is no jitter imposed by other clusters on A. Then, cross jitter for cluster A is 0. The remaining

Clusters	A	B	C
N	3	2	4
p	2	3	6
d	2	3	3
τ	1	1	1

Table 5.1: Parameters for the example

jitter is $d_1 - crossJitter = 2 - 0 = 2$. We can allocate $remJitter/\tau_1 = 2/1 = 2$ devices from cluster A on the first UFB. We update the $\mathbf{UFB}_1^1 = 2$, since the first cluster on the first band has two devices. Now, we try cluster B on the first UFB. The jitter imposed on cluster B from cluster A on the first band is, $\mathbf{UFB}_1^1 * \lceil \frac{p_2}{p_1} \rceil * \tau_1 = 2 * \lceil 3/2 \rceil * 1 = 4$. However, $4 > d_2 = 3$, i.e. the jitter imposed on cluster B is larger than its tolerance. Then, we cannot allocate any devices from cluster B on the first UFB. We update $UFB_2^1 = 0$. Similarly, the jitter imposed by cluster A on cluster C is $2 * \lceil 6/2 \rceil * 1 = 6$ and $6 > d_3 = 3$. Thus, no device from cluster C can be allocated on the first band. We update $UFB_3^1 = 0$.

We still have the third device from cluster A. We allocate this device on the second UFB. Now, we proceed to cluster B, the second highest priority cluster. The cross jitter is $\lceil 3/2 \rceil * 1 = 2$. The remaining jitter is $d_2 - crossJitter = 3 - 2 = 1$. We can allocate $remJitter/\tau_2 = 1/1 = 1$ devices from cluster B on the second band. B2, second device from cluster B, is allocated on the third band. For cluster C, the devices on the second UFB would impose $\lceil 6/2 \rceil + \lceil 6/3 \rceil = 5 > d_3$ jitter, thus cluster C cannot be allocated on the second UFB. With this procedure, we allocate the remaining devices on the third and the fourth UFB.

5.3 Worst Case Performance With Implicit Deadlines

In this section, we find the approximation bound of MFFFA algorithm under the following conditions: devices generate data at integer multiples of $1RB$ and devices have implicit deadlines; i.e. deadlines are equal to period. In order to find the approximation bound, we will use the approximation bound results provided for Rate Monotonic First Fit (RMFF)

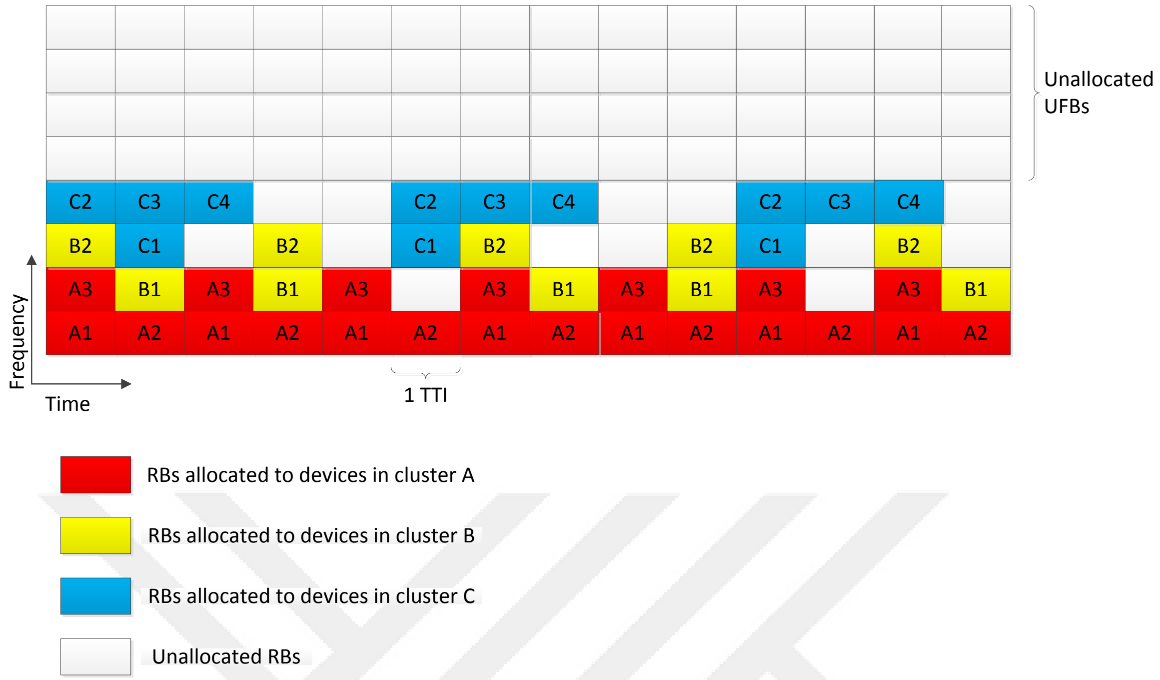

Figure 5.1: Algorithm Description: An example

algorithm in [41]. We will need *Lemma 1* and *Lemma 2* to use the results provided for RMFF algorithm. Thus, we first provide these lemmas, and then establish the approximation bound for MFFFA algorithm using RMFF approximation bound.

Consider a set S of devices $a_1, .., a_N$ with implicit deadlines; and with transmission times $\tau_1, ..., \tau_N RB$ where τ_j is an integer for j=1,...,N. Note that a device a_j requires multiple RB's when $\tau_j > 1$. In order to satisfy jitter constraints, it is sufficient that all τ_j RB's are scheduled before the device makes the next request.

For each device a_j with period p_j and transmission time τ_j RB , define a set S'_j of devices $b_{j_1}, .., b_{j_{\tau_j}}$ with 1 RB transmission time and period p_i. Define $S' = \cup_{j=1}^{N} S'_j$.

Lemma 1 *Set S is schedulable on a single band if and only if set S' is schedulable on a single band.*

Proof: *Assume set S is schedulable. Then, at each period p_j of the device a_j, τ_j RB's are allocated for device a_j. Similarly, there are τ_j devices with 1 RB transmission time in set S'_j with period p_j. Allocate each $b_{j_i} \in [1, \tau_j]$ at the time instant when jth RB of device*

a_i is allocated. Note that device b_{j_i} has identical period and transmission time with the jth RB of a_j, for $i = 1, .., \tau_j$. This results in two identical sets and schedules; which implies set S' is also schedulable.

Assume set S' is schedulable. Then, at each time interval $(t, t+p_j)$, all devices in subset S'_j are scheduled. There are τ_j devices with $1RB$ transmission time in subset S_j. Allocate ith RB of device a_j at the time instant when device b_{j_i} is allocated, for $i = 1, .., \tau_j$. With this allocation scheme, all τ_j RB's of device a_j are allocated in time interval $(t, t+p_j)$; a period of device a_j. Thus, the device a_j is schedulable with the allocation scheme described. Since this holds for an arbitrary j, the set S is schedulable. □

Lemma 2 *For a set of synchronous devices $a_1, .., a_N$ with the uniform tranmission time $\tau = 1RB$, for any scheduling policy the preemptive allocation yields the same schedule as the non-preemptive allocation.*

Proof: *Assume that a packet from the device a_i starts transmission at time $t = t'$ where $t' \in \mathbb{Z}^+$, . Since all service requests are done on integer multiples of 1 RB, there can be no preemption in $(t', t'+1)$ interval. Since transmission times are equal to 1 RB, the packet from device a_i completes its transmission at time $t = t'+1$. Thus, the arbitrary task from an arbitrary device a_i cannot be preempted and this implies that for any scheduling policy the preemptive allocation yields the same schedule as the nonpreemptive allocation.* □

Based on these results, we state the following theorem:

Theorem 2 *Let S be a set of devices. Define N_0 as the minimum number of UFB bands required to schedule the set S and define N as the minimum number of UFB bands required to schedule the set S by MFFFA algorithm. Then, the following relation holds:*

$$N \leq [2 + \frac{(3 - 2^{\frac{3}{2}})}{2^{\frac{4}{3}} - 2}]N_0 + 1 \approx 2.33N_0 + 1 \tag{5.1}$$

Proof: From *Lemma 1*, we conclude that the scheduability of our set S is equivalent to that of S' which has devices of 1 RB transmission time. Based on *Lemma 2*, we observe that the preemptive schedule of set S', therefore S , is equivalent to its non-preemptive schedule.

Thus, the results for preemptive RMFF can be used for non-preemptive MFFFA, since MFFFA is equivalent to RMFF except it does not allow preemption. The worst case analysis for preemptive RMFF algorithm with variable transmission times, which is provided in equation (5.1), also holds for the special case in which all devices have 1 RB transmission time; which provides an equivalent schedule with non-preemptive MFFFA algorithm with multiple integer transmission times. □

5.4 Worst Case Performance With Simply Periodic Set of Devices

In this section, we find the approximation bound of MFFFA algorithm under the following conditions: The transmission time of devices are $1\ RB$ and devices are simply periodic; i.e. for any two devices x_i and x_j with periods $p_i < p_j$, p_j is an integer multiple of p_i. Devices have implicit deadlines. For such a set of devices, we obtain the following approximation bound to the optimal:

Theorem 3 *Define N_0 as the minimum number of UFB bands required to schedule the set S and define N as the minimum number of UFB bands required to schedule the set S by MFFFA algorithm. Then, the following relation holds:*

$$N \leq 1.5N_0 \tag{5.2}$$

Proof: First, we shall show that our scheduling problem is equivalent to bin-packing problem. Bin packing problem can be described as follows: We have objects with different sizes and bins with identical capacities. We need to pack these objects into bins such that the number of bins used is minimized. In order to show that our problem is equivalent to bin-packing, we use the following result in [42]: for a preemptive, simply periodic system with implicit deadlines, the rate monotonic (RM) algorithm is schedulable on a uniform processor if and only if the total utilization of devices is equal to or less than 1. If we treat the utilization of devices as objects and UFBs as bins with size 1; our scheduling problem becomes a bin-packing problem where each UFB has size 1, and the objective is to minimize the number of UFBs used; given that preemptive RM algorithm is used at individual UFBs. Further note that since we use devices generating data at integer multiples of RBs with 1 RB transmission time; preemptive RM schedule is equivalent to nonpreemptive RM schedule due to *Lemma 2* .

Having established the equivalence between our problem and bin-packing; we can use our proposed heuristic algorithm. MFFFA first orders devices in increasing period (i.e. decreasing utilization), and allocates them with RM algorithm on individual UFBs. MFFFA is equivalent to First Fit Decreasing algorithm for which the approximation bound is proved to be 1.5 in [43].

□

Chapter 6

Call Admission Control

In this section, we provide an admission control scheme to manage admission of newly joining time-triggered MTC devices. Our admission control scheme is designed such that newly arriving devices use minimum number of UFBs while providing QoS guarantees for existing and newly arriving devices. We further require that devices that are already allocated are given priority over newly arriving devices; i.e. newly arriving devices do not affect the performance of allocated devices.

6.1 Call Admission Optimization Problem

In accordance with this objective and constraints, we would like to obtain the minimum achievable number of bands used throughout semi-persistent period P while new devices arrive to the network. At each TTI, the number of bands required to allocate newly arriving and already arrived devices are minimized. Note that any algorithm implementing this optimization problem has to be offline since it requires the information regarding all devices that will arrive until the particular TTI for which we minimize the number of bands used. On the other hand, our resource allocation algorithms must be online since we do not have information on newly arriving devices. However, this optimization problem is useful in providing a theoretical upper bound with which we can compare the performance of our online algorithms.

For the optimization problem, we consider the following system: There are T TTIs, and new time-triggered MTC devices may arrive at each TTI $t \in [1, T]$. The priority ordering of devices are as follows: Devices arriving earlier have higher priority; i.e. for any two devices i and j, if $t_i < t_j$, then device i has higher priority than device j. If two devices arrive at the same TTI, then the priority ordering is the same as the one described in Section 2. With the exception of a modified priority ordering, our optimization problem is exactly the same as the optimization problem described in Section 4.

Call admission optimization problem is NP-hard. In order to prove this, take a specific instance of the problem where $|T| = 1$, i.e. we have only one TTI. In that specific instance, our call admission optimization problem is equivalent to the optimization problem provided in Section 4; and this problem is proved to be NP-hard in Section 4-C. Therefore, solving the call admission optimization problem requires an exponential runtime in the number of newly arriving devices. The optimization problem further requires an offline algorithm which is not possible to implement in practice. Thus, we need an online heuristic algorithm for a fast and efficient solution.

6.2 Call Admission Mechanism

Our heuristic algorithms for the call optimization problem works as shown in Fig.6.1. We first run the First Fit Occupied Bands (FFOB) Algorithm to allocate newly arriving devices to already occupied UFBs. If there are still unallocated devices after running the FFOB algorithm, the base station allocates remaining devices to new bands using MFFFA. However, if call admission mechanism requires more bands than k_{max}; then the lowest priority devices are dropped until the allocation can be made on k bands such that $k \leq k_{max}$.

6.2.1 First Fit for Occupied Bands Algorithm

FFOB algorithm is based on the same principle as the MFFFA: Devices are ordered according to priority assignments, and then these devices are allocated to the first UFB on which they can be allocated. However, since FFOB algorithm is exclusively applied to the UFBs on which some devices are already allocated and thus have absolute priority, it necesarily differs from MFFFA. Already occupied devices impose jitter on newly arriving devices since they have absolute priority regardless of their periods. Further, note that in FFOB

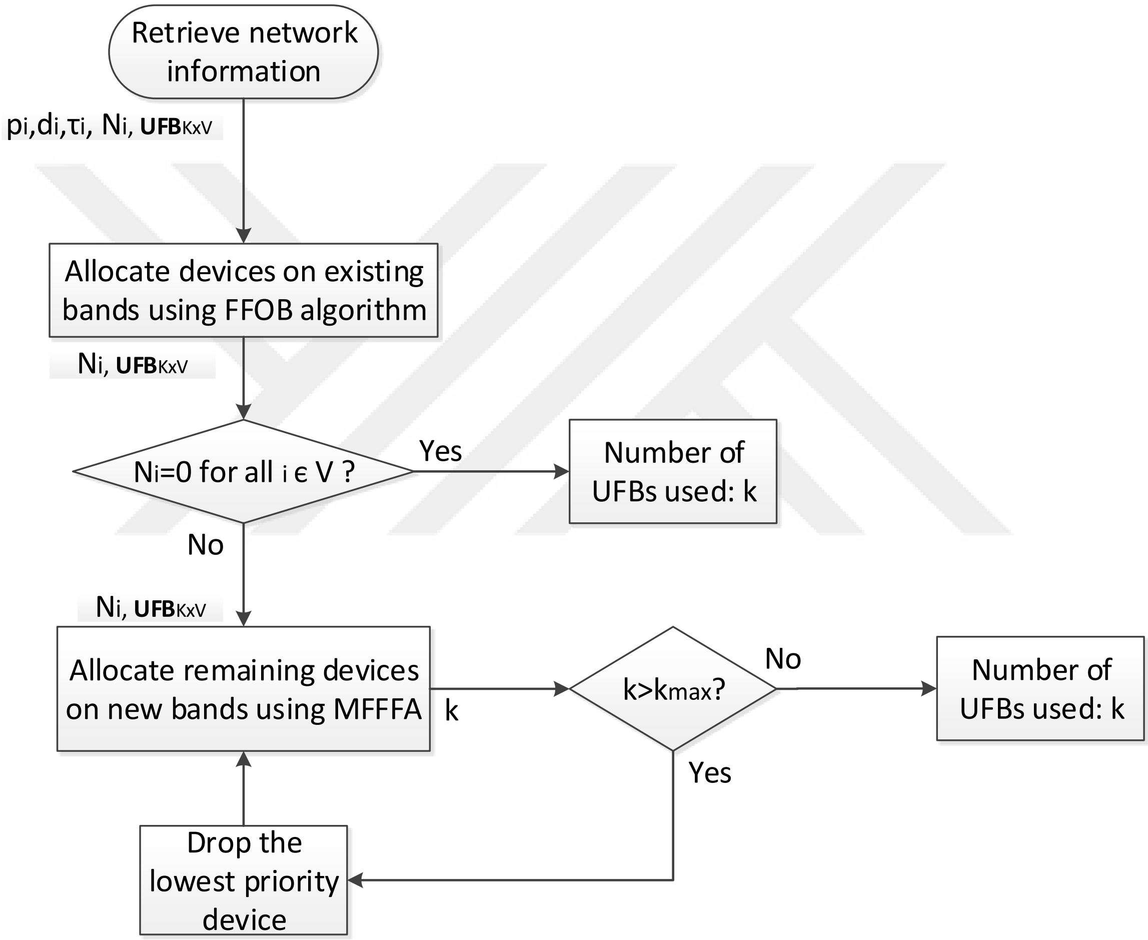

Figure 6.1: Call admission control mechanism

algorithm, unlike MFFFA, we do not aim to allocate all devices. We try devices from each cluster at each band and if it cannot be allocated within already occupied bands, it will be allocated to new bands with MFFFA as shown in Fig. 6.1.

Algorithm 3 **First Fit Occupied Bands Algorithm**

1: **Input**: $d_i, p_i, \tau_i, \mathbf{N}_i$ for $i \in [1, V]$, f_{ik} for $i \in M, k \in [1, K]$
2: **Output**: $\mathbf{N}_i$ for $i \in [1, V]$, $\mathbf{UFB}_{K \times V}$, for $k \in [1, K]$
3: $\mathbf{UFB} = zeros(K, V)$;
4: **for** $i = 1 : V$ **do**
5:     **for** $k = 1 : K$ **do**
6:         $crossJitter = 0$;
7:         **for** $j \in M$ **do**
8:             $crossJitter = crossJitter + f_{jk} * \lceil \frac{p_i}{p_j} \rceil * \tau_j$;
9:         **end for**
10:         **for** $j = 1 : i$ **do**
11:             $crossJitter = crossJitter + \mathbf{UFB}_j^k * \lceil \frac{p_i}{p_j} \rceil * \tau_j$;
12:         **end for**
13:         $remJitter = d_i - crossJitter$;
14:         **if** $remJitter > 0$ **then**
15:             $\mathbf{UFB}_i^k = \mathbf{UFB}_i^k + \min(\mathbf{N}_i, \lfloor \frac{remJitter}{\tau_i} \rfloor)$;
16:             $\mathbf{N}_i = \mathbf{N}_i - \min(\mathbf{N}_i, \lfloor \frac{remJitter}{\tau_i} \rfloor)$;
17:         **end if**
18:     **end for**
19: **end for**

Jitter tolerance, period, transmission time and number of devices arriving for all clusters $i \in [1, V]$ are given and devices are sorted according to their priority assignments. The set of clusters that have devices already allocated on at least one UFB is denoted by M. f_{ik} refers to the number of devices from cluster $i \in M$ that are already allocated on band k (Line 1). The output of the FFOB algorithm is the number of unallocated devices after running the algorithm, $\mathbf{N}_i$, and $\mathbf{UFB}_{K \times V}$ whose entry (k, v) refers to the number of devices allocated from cluster v on band k (Line 2). We first initialize **UFB** (Line 3). Starting from the highest priority cluster (Line 4), we calculate cross jitter, the jitter exprienced due to devices in higher priority clusters, for each cluster at each band k (Lines 4-12). Note that unlike MFFFA, devices experience an additional jitter from all devices that are already assigned (Line 8). Then, we calculate the remanining jitter for each cluster at each band, and allocate devices to UFBs with exactly the same procedure we applied in MFFFA (Line

13-16).

Chapter 7

Performance Evaluation

The goal of this section is to compare the performance of the proposed algorithm MFFFA to previously proposed algorithms and optimal solution in terms of schedulability and usage of frequency band. We will first first show that the existing methods cannot guarantee schedulability for a set of devices, even when the radio resources are unlimited. To show that, we will compare the schedulablility performance of MFFFA algorithm with that of CBSA and two random acess based methods: the basic LTE without ACB, and the LTE system with dynamic optimal ACB obtained in [10]. We will then show that MFFFA algorithm uses the radio resources more efficiently than CBSA algorithm. Finally, we will show that our proposed call admission control mechanism uses frequency resources far more efficient than that of CBSA, and therefore it is also more robust for a dynamic environment with changing traffic load and characteristics.

7.1 Related Works

Our proposed MFFFA algorithm have two major advantages over existing resource allocation methods. First, it eliminates the random access procedure, thereby saving frequency resources while avoiding signalling needed for random acccess. Second, it efficiently uses frequency resources needed for data allocation. Methods employing random access suffer from congestion due to limited random access resources. The congestion in random access leads to packet losses. Thus, for a given set of devices, MFFFA algorithm has better schedulability performance, compared to random access based methods. In order to show that, we compare the schedulability performance of our algorithm with two random access based methods.

We further compare the frequency band usage and schedulability performance of our proposed algorithm with the semi-persistent scheduling algorithm CBSA proposed in [22]. This will prove that our algorithm has better performance than both methods employing random access and methods that eliminate the random access. First, we describe these methods in detail.

7.1.1 Random Access Methods

Before describing proposed random access methods, we briefly summarize random access procedure. In the first step, devices send a preamble to base station. There are a limited set of preambles, and devices pick one of these preambles. If a preamble is selected by multiple devices, base station cannot differentiate between these devices, and therefore these devices cannot be served. The primary component of random access congestion is due to limited number of preambles. We assume that if there is only one device selecting a particular preamble, then this device successfully completes random access and further successfully sends the data. Thus, for random access methods, we are only interested in packet losses due to random access congestion arising from selection of the same preambles by different devices. We do not include a possible frequency band shortage in data allocation since we want to show that even random access procedure alone leads to significant packet losses. With respect to physical layer, in order to get the best possible result out of random access based solutions we will make the following assumptions as in [44]: the channel is ideal, there is no signal loss due to radio propagation problems. We ignore the processing times both in MTC devices and the base station, so that devices do not miss any random access opportunity due to processing delays. We further ignore any congestion or shortage of resources in the downlink channels and assume that once a preamble is successfully transmitted, the device accesses the channel. This enables us to obtain the best possible random access performance with respect to physical layer; thus we can get the theoretical upper bound on the schedulability results.

The basic LTE is the random access method which does not use any access class barring method. Therefore, devices are not barred from access before they attempt random access. In the event of collision at the random access, colliding devices wait for a backoff time. Backoff is used as the load control measure. However, backoff alone may not be a sufficient load control mechanism. Therefore, we also consider a random access method which employs an additional load control mechanism: access class barring (ACB).

In ACB method, devices select a random number in [0,1] and compare that number with ACB parameter broadcasted by base station. If the random number selected is smaller than the ACB parameter, then the device proceeds with the random access procedure. Otherwise, it waits for a backoff time. Hence, ACB parameter can be used to control the traffic load in random access procedure. A small ACB parameter will decrease the traffic load, whereas a large ACB parameter increases the traffic load.

In [10] an LTE system with dynamic optimal ACB is described. The ACB parameter proposed in [10] maximizes the number of devices successfully sending the preamble. This is achieved by dynamically tuning the ACB parameter based on network traffic load. Note that in practice, the base stations cannot obtain the optimal ACB in [10], and it can only be approximated. Thus an optimal ACB scheme is not applicable. Yet, in order to get the best performance out of this method, we assume that base station can obtain the optimal ACB parameter. Thus, this is the best possible load control method for random access.

7.1.2 CBSA

The clustering-based scheduling algorithm (CBSA) proposed in [21] is described as follows: Devices are grouped into clusters according to their periods and jitter tolerances. Transmission times of all devices are assumed to be equal and 1 RB. Devices in the same cluster are allocated resources at the same time, using different UFBs. Once a cluster is allocated resources at a particular TTI, no device from other clusters is allocated at this TTI. Clusters are assigned priorities according to their periods, where clusters with lower periods are assigned higher priorities.

7.2 Simulation Scenario

Since the CBSA assumes that all devices have 1 RB transmission times, we also make this assumption for MFFFA algorithm in our simulations, for the sake of comparison. The optimal solution is denoted by OPT. The simulation results are obtained from 1000 random network topologies in which devices are uniformly distributed within a circle of radius r with the base station at the center. For random access based solutions, we define the schedulability ratio as the ratio of total number of packets that are successfully sent (i.e. packets that are not lost) and the total number of packets generated. Packet losses occur under the following conditions: if the number of retransmissions for a packet exceeds the maximum number of retransmissions or if a device produces the next packet before successfully transmitting a packet. For random access based methods, the simulation time is 2 seconds and LTE parameters are adapted from [45]. The packet generation period and jitter tolerance of clusters are uniformly chosen from [10, 2000] ms and [10,p_i] for $i \in [1, M]$ where p_i is packet generation period of cluster i, respectively.

We will further compare the frequency band usage in data allocation for MFFFA, CBSA and OPT. The number of frequency bands used by each algorithm is calculated for all random network topologies and the average number of frequency bands used for each algorithm is obtained. The simulation results for MFFFA and CBSA are obtained in MATLAB; whereas optimal solution is obtained in GAMS, a high-level language for modeling and solving optimization problems.

7.3 Schedulability Performance

Fig. 7.1 compares the schedulability performance of our proposed MFFFA algorithm, CBSA, optimal dynamic ACB (denoted by DACB) method and basic LTE for constant number of clusters where the number of clusters are fixed to 12. The performance of random access based methods deteriorate as the number of devices increase. As expected, the deterioration in performance is larger in basic LTE. Note that these performances for random access are obtained under the ideal conditions and all available random access

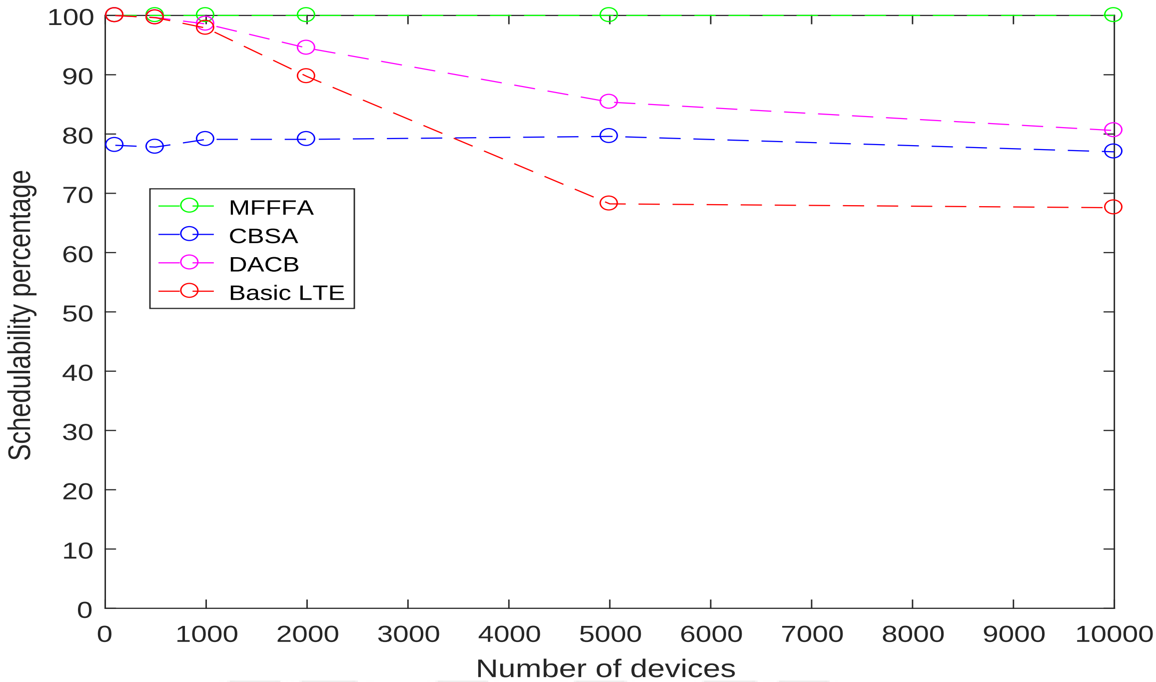

Figure 7.1: Schedulability results for constant number of clusters

preambles are allocated to M2M communications. Hence, in more realistic scenarios, the random access based methods have even worse performance. For CBSA algorithm, note that given we have sufficient number of frequency bands, the only parameter determining schedulablility is number of clusters. Therefore, the performance of CBSA is not affected by increasing number of devices. However, the performance is persistently low as only around 80 percent of devices can be served. Further note that the entire bandwidth is allocated to each cluster in CBSA. However, even for medium network sizes where the number of devices are on the order of hundreds, the frequency band available will not be sufficient for allocation of these devices. Thus, in practice, where the frequency band is not unlimited, the schedulability performance of CBSA would be much worse.

7.4 Frequency Band Usage Performance

Fig. 7.2, shows the number of frequency bands used by MFFFA and CBSA for different number of clusters. The number of devices belonging to each cluster are uniformly chosen

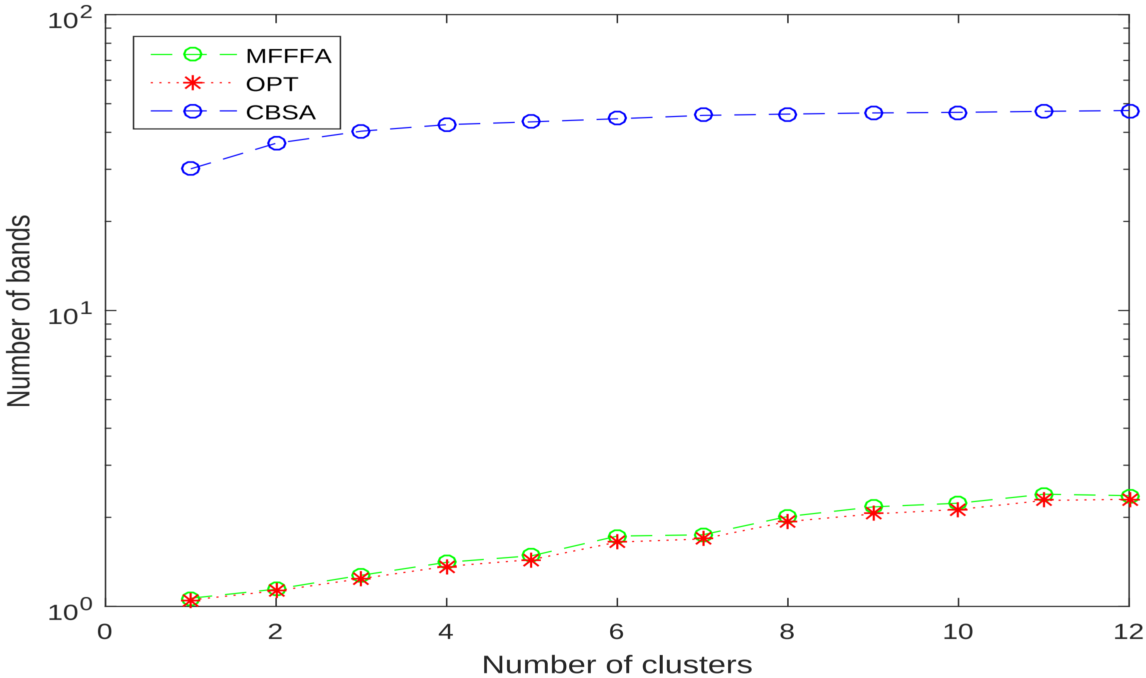

Figure 7.2: Average number of frequency bands used by each algorithm and the optimal solution

from [10,50]. The average number of frequency bands used by CBSA approaches to its maximum very quickly as the number of devices increase. The number of frequency bands used by CBSA depends only on the number of devices in the cluster with the maximum number of devices since each cluster uses one frequency band per device. As the number of clusters increase, the probability that a cluster has a higher number of devices also increases. This explains the quick increase in the number of UFBs used by CBSA. MFFFA, on the other hand, significantly outperforms CBSA for all network topologies considered. Furthermore, MFFFA performs very close to the OPT, with an approximation ratio of 1.04 where approximation ratio refers to the ratio of output of MFFFA and OPT.

Fig. 7.3 shows frequency band usage for MFFFA, CBSA and OPT where the number of clusters are fixed to 12. Different number of devices are uniformly assigned to the clusters. As we stated before, the number of frequency bands used by CBSA depends only on the number of devices at the cluster containing the maximum number of devices. Thus, as we increase the number of devices, the probability that the number of devices in the largest

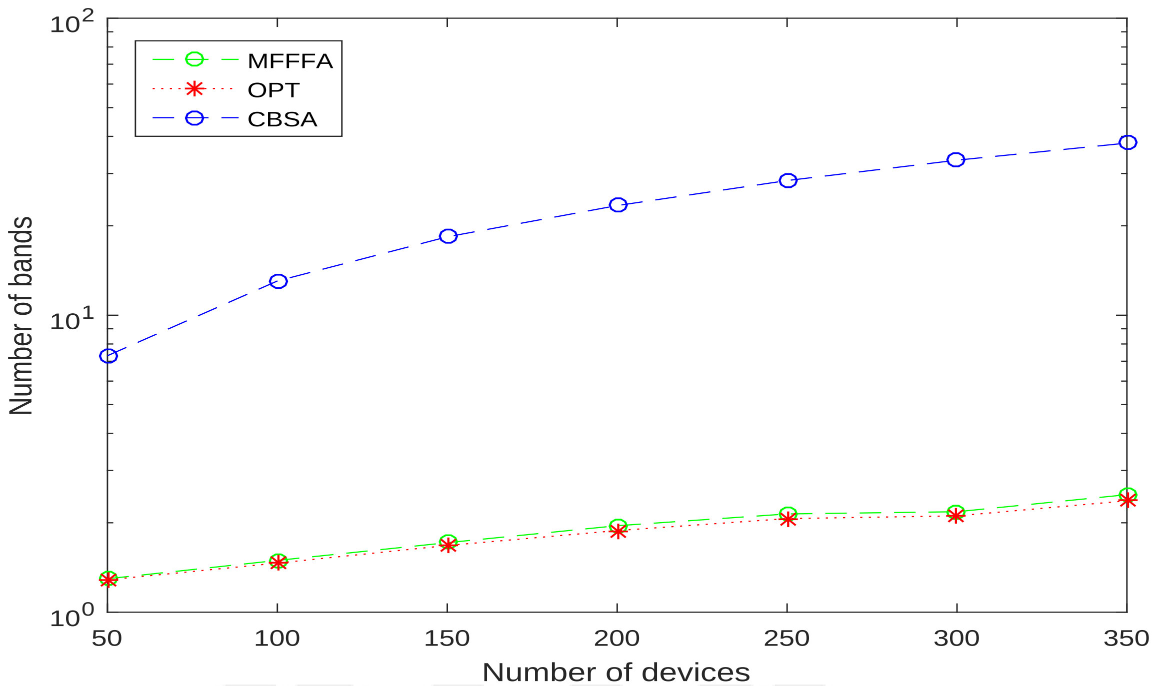

Figure 7.3: Average number of frequency bands used by each algorithm and the optimal solution with fixed number of clusters

cluster will be higher increases. This explains the increase in the number of frequency bands used by CBSA as the number of devices increases. MFFFA, on the other hand, performs much better than CBSA. Moreover, its performance is very close to OPT with the approximation ratio of 1.03. This is a further advantage for MFFFA as it performs close to optimal with respect to both different number of clusters and fixed number of clusters with different number of devices.

7.5 Runtime performance

In this section, we evaluate the runtime performance of MFFFA, FBSE, CBSA and the optimum solution obtained from MATLAB. We use the same simulation environment, MATLAB, for all algorithms to have a better comparison of runtime performances. The runtimes are obtained over 100 random network topologies, and average runtimes are considered. We

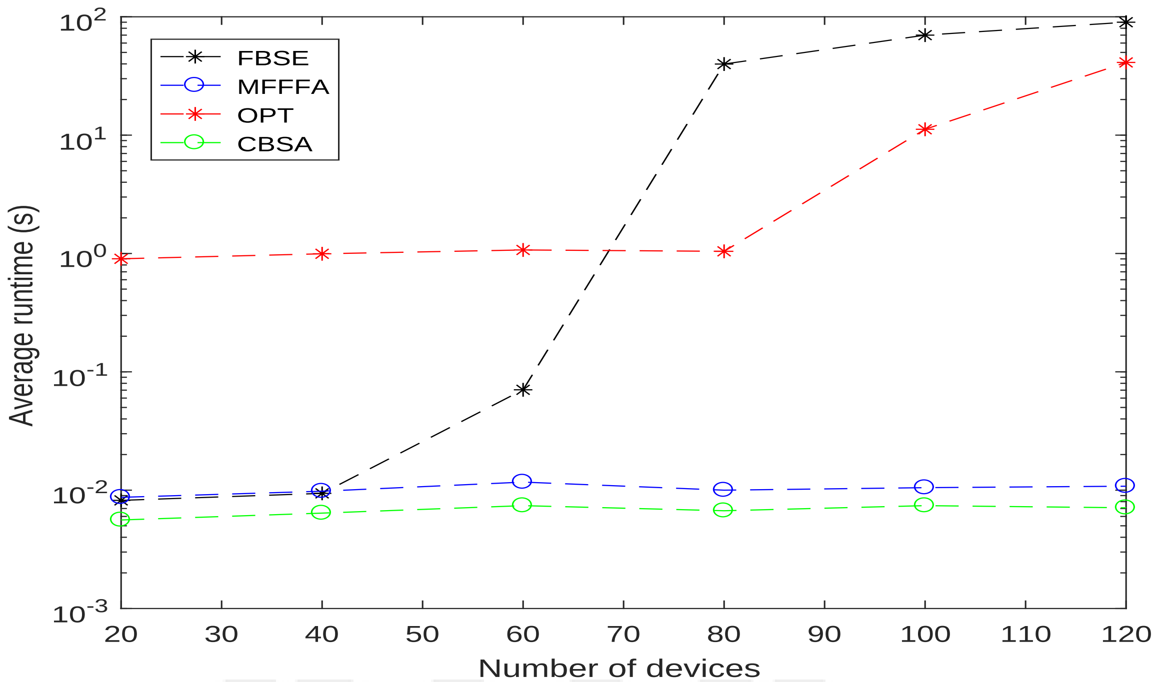

Figure 7.4: Runtime performance comparison of exponential-time optimum solutions and polynomial-time heuristic algorithms

obtain the optimum solution from MATLAB using the optimization toolbox. However, the optimum solution cannot be attained for all network configurations; i.e. for some network configurations the optimization solver stops prematurely when time or resource limit is exceeded. Furthermore, FBSE algorithm may also take a very long time to converge with medium network sizes. This is expected since the optimization problem does not have a solution in polynomial time. In order to overcome this problem for our simulation, we set a time limit of 500 s to find the optimum solution for both optimization solver of MATLAB and FBSE algorithm. If the MATLAB optimization solver or FBSE does not find a solution in 500 s, we stop and store the runtime to be 500 s.

Fig. 7.4 shows the runtime performances of both exponential optimal algorithms and heuristic polynomial time algorithms. Note that since MFFFA and CBSA algorithms are polynomial-time algorithms (their computational complexities are $O(NM^2)$ and $O(M^2)$, respectively, where N is number of devices and M is number of clusters), their runtimes are far lower than the optimum algorithms. Note that the runtime of MFFFA algorithm is very close to CBSA algorithm. The slight difference is expected as CBSA has a lower computational complexity. On the other hand, FBSE algorithm performs better than the MATLAB optimization solver for small network sizes, as it provides a more efficient search method. However, as the network size grows, methods that are not directly related to search (such as relaxation methods, heuristics and cut generations) improves the performance of MATLAB solver. Therefore, for larger network sizes, even though FBSE provides a more efficient search method, it performs worse than the MATLAB solver. We also observe that both MATLAB solver and FBSE algorithm become less efficient as network size grows, since they are exponential-time methods with respect to network size. This also justifies our heuristic algorithm from a runtime perspective, as it shows that a polynomial-time heuristic algorithm is necessary for medium and large network sizes.

7.6 Performance Evaluation of Proposed Call Admission Control Mechanism

Fig. 7.5 compares our proposed call admission control mechanism with call admission mechanism proposed for CBSA in [21]. We start with 12 clusters with 50 devices randomly distributed to these clusters. The scheduling update period is determined to be 10 s, where device may arrive at any TTI of length 1 ms. The simulation duration is 20 sn. Since CBSA only considers arrivals of new devices, we also make the same assumption for the sake of comparison. We model the arrival of devices for each cluster at any TTI as a Poisson process with mean $\lambda = 0.001$. This implies that the expected number of devices arriving per cluster is 60 per minute.

We observe that our call admission mechanism is very efficient in the number of frequency bands it uses. Note that the increase in frequency bands used within a scheduling update period is very slight, whereas for CBSA the number of frequency bands occupied increases very rapidly as new devices arrive to the network. Further note that we drop the condition that early arriving devices have the priority at the schedule update points for our call admission algorithm and thus the base station can allocate devices more efficiently. This explains why the optimum solution, which also updates its schedule at schedule update points, improves at the schedule update point (at 10 sn) even though new devices

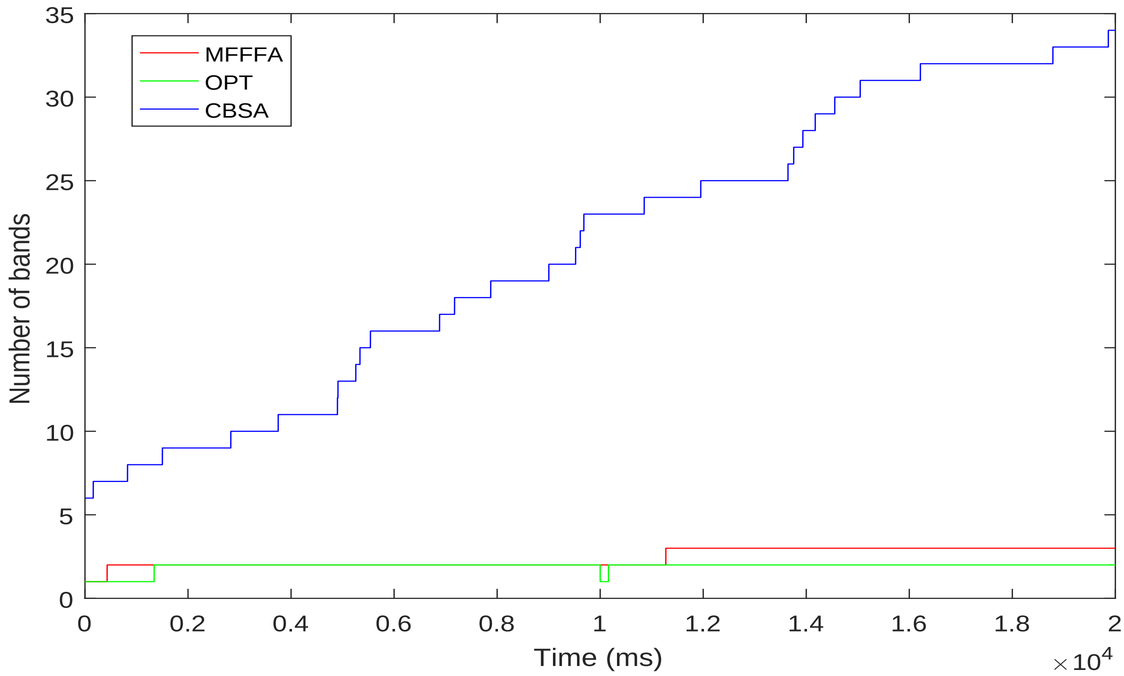

Figure 7.5: Number of frequency bands used in Call admission control

keep arriving to the network. Note that even under such large number of new arrivals, our call admission algorithm performs much better than CBSA algorithm. On the other hand, CBSA algorithm does not use the scarce frequency resources efficiently, leading to resource starvation under high traffic load.

Chapter 8

Conclusion

We studied the problem of M2M packet scheduling with QoS guarantees in cellular networks, particularly considering the radio resource starvation problem in 5G technology with dramatically growing numbers of M2M applications and devices. Accordingly, we formed an optimization problem whose objective is to minimize the number of frequency bands occupied by a given set of MTC devices where the constraint is that jitter constraints of all devices must be satisfied. The optimization problem is proven to be NP-hard upon which we provide an efficient heuristic scheduling algorithm. We further establish performance guarantees for the proposed heuristic algorithm by constructing an approximation bound to the optimal. Extensive simulations show that our proposed algorithm performs much better than the existing scheduling algorithms in minimizing frequency bands and schedulability. We further provide a call admission control scheme to dynamically manage arrivals of new devices to the network and through simulations we show its superiority in comparison to existing scheduling algorithms. In future, this work can be further extended to include the multi-hop networks.

Bibliography

[1] G. Wunder, P. Jung, M. Kasparick, T. Wild, F. Schaich, Y. Chen, S. T. Brink, I. Gaspar, N. Michailow, A. Festag, L. Mendes, N. Cassiau, D. Ktenas, M. Dryjanski, S. Pietrzyk, B. Eged, P. Vago, and F. Wiedmann, "5gnow: non-orthogonal, asynchronous waveforms for future mobile applications," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 97–105, February 2014.

[2] G. Wu, S. Talwar, K. Johnsson, N. Himayat, and K. D. Johnson, "M2m: From mobile to embedded internet," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 36–43, April 2011.

[3] R. Pepper, "The rise of m2m devices," October 2015, 3rd BEREC Stakeholder Forum.

[4] M. T. Islam, A. e. M. Taha, and S. Akl, "A survey of access management techniques in machine type communications," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 74–81, April 2014.

[5] *Backoff Enhancements for RAN Overload Control*, ZTE, Barcelona, May 2011, 3GPP R2-112863.

[6] G. Y. Lin, S. R. Chang, and H. Y. Wei, "Estimation and adaptation for bursty lte random access," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2560–2577, April 2016.

[7] J. B. Seo and V. C. M. Leung, "Design and analysis of backoff algorithms for random

access channels in umts-lte and ieee 802.16 systems," *IEEE Transactions on Vehicular Technology*, vol. 60, no. 8, pp. 3975–3989, Oct 2011.

[8] 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.

[9] Z. Wang and V. W. S. Wong, "Optimal access class barring for stationary machine type communication devices with timing advance information," *IEEE Transactions on Wireless Communications*, vol. 14, no. 10, pp. 5374–5387, Oct 2015.

[10] S. Duan, V. Shah-Mansouri, Z. Wang, and V. Wong, "D-acb: Adaptive congestion control algorithm for bursty m2m traffic in lte networks," *IEEE Transactions on Vehicular Technology*, vol. PP, no. 99, pp. 1–1, 2016.

[11] H. S. Dhillon, H. Huang, H. Viswanathan, and R. A. Valenzuela, "Fundamentals of throughput maximization with random arrivals for m2m communications," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 4094–4109, Nov 2014.

[12] S. Cherkaoui, I. Keskes, H. Rivano, and R. Stanica, "Lte-a random access channel capacity evaluation for m2m communications," in *2016 Wireless Days (WD)*, March 2016, pp. 1–6.

[13] C. Anton-Haro and M. Dohler, *Machine-to-machine (M2M) Communications: Architecture, Performance and Applications*, ser. Woodhead Publishing Series in Electronic and Optical Materials. Elsevier Science, 2014.

[14] C. Y. Oh, D. Hwang, and T. J. Lee, "Joint access control and resource allocation for concurrent and massive access of m2m devices," *IEEE Transactions on Wireless Communications*, vol. 14, no. 8, pp. 4182–4192, Aug 2015.

[15] D. T. Wiriaatmadja and K. W. Choi, "Hybrid random access and data transmission protocol for machine-to-machine communications in cellular networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 1, pp. 33–46, Jan 2015.

[16] N. Afrin, J. Brown, and J. Y. Khan, "Design of a buffer and channel adaptive lte semi-persistent scheduler for m2m communications," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 5821–5826.

[17] 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.

[18] J. B. Seo and V. C. M. Leung, "Performance modeling and stability of semi-persistent scheduling with initial random access in lte," *IEEE Transactions on Wireless Communications*, vol. 11, no. 12, pp. 4446–4456, December 2012.

[19] D. Jiang, H. Wang, E. Malkamaki, and E. Tuomaala, "Principle and performance of semi-persistent scheduling for voip in lte system," in *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, Sept 2007, pp. 2861–2864.

[20] "Persistent Scheduling in E-UTRA," Sorrento,Italy, Tech. Rep., January 2007, 3GPP TSG RAN WG1 Meeting 47bis, R1-070098.

[21] 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.

[22] S. Y. Lien, K. C. Chen, and Y. Lin, "Toward ubiquitous massive accesses in 3gpp machine-to-machine communications," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 66–74, April 2011.

[23] A. G. Gotsis, A. S. Lioumpas, and A. Alexiou, “Analytical modelling and performance evaluation of realistic time-controlled m2m scheduling over lte cellular networks,” *Transactions on Emerging Telecommunications Technologies*, vol. 24, no. 4, pp. 378–388, 2013. [Online]. Available: http://dx.doi.org/10.1002/ett.2629

[24] ——, “Evolution of packet scheduling for machine-type communications over lte: Algorithmic design and performance analysis,” in *2012 IEEE Globecom Workshops*, Dec 2012, pp. 1620–1625.

[25] S. Zhenqi, Y. Haifeng, C. Xuefen, and L. Hongxia, “Research on uplink scheduling algorithm of massive m2m and h2h services in lte,” in *Information and Communications Technologies (IETICT 2013), IET International Conference on*, April 2013, pp. 365–369.

[26] A. M. Maia, D. Vieira, M. F. de Castro, and Y. Ghamri-Doudane, “A mechanism for uplink packet scheduler in lte network in the context of machine-to-machine communication,” in *2014 IEEE Global Communications Conference*, Dec 2014, pp. 2776–2782.

[27] E. Dahlman, S. Parkvall, and J. Skold, *4G: LTE/LTE-advanced for mobile broadband*. Academic press, 2013.

[28] Z. Fan, R. J. Haines, and P. Kulkarni, “M2m communications for e-health and smart grid: an industry and standard perspective,” *IEEE Wireless Communications*, vol. 21, no. 1, pp. 62–69, February 2014.

[29] M. N. Shehzad, A.-M. Déplanche, Y. Trinquet, and R. Urunuela, “Overhead control in real-time global scheduling.” in *RTNS*, 2011, pp. 45–52.

[30] L. Doyle and J. Elzey, “Successful use of rate monotonic theory on a formidable real time system,” in *Real-Time Operating Systems and Software, 1994. RTOSS ’94, Proceedings., 11th IEEE Workshop on*, May 1994, pp. 74–78.

[31] K.-C. Chen, "Machine-to-machine communications for healthcare," *Journal of Computing Science and Engineering*, vol. 6, no. 2, pp. 119–126, 2012.

[32] K. K. Chintalapudi and L. Venkatraman, "On the design of mac protocols for low-latency hard real-time discrete control applications over 802.15.4 hardware," in *Information Processing in Sensor Networks, 2008. IPSN '08. International Conference on*, April 2008, pp. 356–367.

[33] V. Misic and J. Misic, *Machine-to-Machine Communications: Architectures, Technology, Standards, and Applications*. CRC Press, 2014.

[34] H. Yao, T. Huang, C. Zhao, X. Kang, and Z. Liu, "Optimal power allocation in cognitive radio based machine-to-machine network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2014, no. 1, pp. 1–9, 2014. [Online]. Available: http://dx.doi.org/10.1186/1687-1499-2014-82

[35] A. Ali, W. Hamouda, and M. Uysal, "Next generation m2m cellular networks: challenges and practical considerations," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 18–24, September 2015.

[36] Y. Mehmood, C. Grg, and A. Timm-Giel, "A radio resource sharing scheme for iot/m2m communication in lte-a downlink," in *2016 IEEE International Conference on Communications Workshops (ICC)*, May 2016, pp. 296–301.

[37] 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.

[38] M. Weiner, M. Jorgovanovic, A. Sahai, and B. Nikoli, "Design of a low-latency, high-reliability wireless communication system for control applications," in *2014 IEEE International Conference on Communications (ICC)*, June 2014, pp. 3829–3835.

[39] J. Penttinen, *Wireless Communications Security: Solutions for the Internet of Things*. Wiley, 2016.

[40] A. Jain and I. H. Hou, “R-pf: Enhancing service regularity for legacy scheduling policy,” *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 258–266, Jan 2016.

[41] Y. Oh and S. H. Son, “Allocating fixed-priority periodic tasks on multiprocessor systems,” *Real-Time Systems*, vol. 9, no. 3, pp. 207–239, 1995.

[42] J. Liu, *Real-Time Systems*. Prentice Hall, 2000.

[43] D. Simchi-Levi, “New worst-case results for the bin-packing problem,” *Naval Research Logistics*, vol. 41, no. 4, p. 579, 1994.

[44] M. Koseoglu, “Lower bounds on the lte-a average random access delay under massive m2m arrivals,” *IEEE Transactions on Communications*, vol. 64, no. 5, pp. 2104–2115, May 2016.

[45] *Study on RAN Improvements for Machine Type Communications*, September 2011, 3GPP TR 37.868 V11.0.0.