IMSI Sharing based Dynamic and Flexible Traffic Aggregation for Massive IoT Networks

Amirahmad Chapnevis, *Member, IEEE,* Ismail Güvenç, *Fellow, IEEE,* and Eyuphan Bulut, *Senior Member, IEEE,*

***Abstract*—International Mobile Subscriber Identity (IMSI) sharing based aggregated communication aims to connect multiple Internet of Things (IoT) devices to the mobile operator's core network over the same subscriber line. IoT devices with low data rates and long data sending intervals are first grouped together and assigned the same subscriber identity. They then take turns to perform their data exchanges using the same cellular connection, yielding huge saving in resource (e.g., number of active bearers) usage. Current solutions however do not consider different device traffic characteristics, the flexibility in traffic patterns, and dynamic network environments where new IoT devices join and existing ones leave the network. In this paper, we study the problem of grouping of IoT devices that will share the same subscriber identity based on their traffic patterns which can also be slightly shifted. We also study the efficient regrouping of these devices as the set of devices in the network changes. We first solve the optimal grouping and traffic aggregation problem for the initial and updated network states using Integer Linear Programming (ILP). Then, to avoid the high complexity of ILP solutions, we develop heuristic based solutions. Through extensive simulations, we show that heuristic based algorithms can provide close to optimal ILP based results while running much faster. The results also show that shifting based grouping provides more resource saving compared to no shifting based aggregation and the proposed solution for dynamic environments can maintain the resource saving with a much lower complexity.**

***Index Terms*—Massive IoT, IMSI sharing, core network, machine type communications (MTC), resource optimization.**

I. Introduction

Internet of Things (IoT) technology has revolutionized our daily lives through many applications (e.g., smart cities [1], environmental monitoring [2], home automation [3]) that use various types of devices with ubiquitous connectivity. This has caused a paradigm shift from human based communications to machine based communications and increased the volume of machine-type devices (MTD) [4]. Thus, mobile network operators (MNO) have faced new challenges due to the limited wireless spectrum and scarce resources available in their core networks.

In order to address such challenges generated by massive IoT networks, there have been many studies performed with solutions in different network layers and new standards (e.g., Narrowband or NB-IoT [5]) for next generation IoT networks have been developed. These efforts mostly focus on solving the radio side issues such as spectrum efficiency [6], [7], energy efficiency [8]–[10] through the usage of massive Multiple-Input-Multiple-Output (MIMO), relays [10], mmWave systems [11] and multi-operator based resource sharing [12].

In this study, we focus on the utilization of mobile core network resources (e.g., *bearers* or data paths in Evolved Packet Core (EPC)) in order to provide scalable communication architecture for massive number of MTDs. As gateways in a core network are primarily designed to handle the traffic from mobile users, the resources and limitations are set considering current communication characteristics of mobile users. Thus, for example, for the MTDs that rarely send data, the resources regarding their connection to the core network will be underutilized if each MTD directly connects to the macrocell base station (BS) and the core network separately. Note that power saving mode (PSM) [13] introduced in 3rd Generation Partnership Project (3GPP) Release 12 turns the device's radio off when the device is not sending data and reduces the load on macro BS by releasing the channel resources allocated to the device. However, in the core network side, the device continues to use some resources as it is still registered with the network. For example, in the case of EPC, the connection between Serving Gateway (SGW) and Mobility Management Entity (MME) is deleted by switching to PSM, but Packet Data Network Gateway (PGW) and MME still keep the connection (i.e., bearer) information of the device thus continue to utilize their memory resources.

One common approach to reduce the usage of connection resources at the core network for such MTDs is to connect the nearby MTDs to a local gateway device and have them send their data traffic using this gateway's connection. Note that this can be achieved through a star topology or multiple hops among devices using device-to-device (D2D) communication [14], [15]. The main issue with such an approach is that due to the limited range of the D2D technology (e.g., Bluetooth Low Energy (BLE), WiFi-direct) used, it will only be a local solution and the number of MTDs that can be connected to the gateway will be limited. Moreover, the capacity of the single backhaul connectivity from the gateway to the macro BS should be large enough to carry all traffic from the connected devices. There are also solutions that aim to manage the connection of MTDs to a macro BS using a group-based Radio Resource Connection (RRC) and bearer establishment [16] but again these solutions can only be applied for the MTDs in the range of the same macro BS.

For a more scalable solution, recently a group-based connection to the core network has been introduced through the sharing of subscriber identity among the devices [17]. The

A. Chapnevis and E. Bulut are both with the Department of Computer Science, Virginia Commonwealth University, Richmond, VA, 23284. E-mail: {chapnevisa, ebulut}@vcu.edu.

I. Güvenç is with the Department of Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695. E-mail: {iguvenc}@ncsu.edu.

goal is to connect a group of MTDs with same data sending intervals over the same subscriber identity i.e., International Mobile Subscriber Identity (IMSI) and have them take turns for their data communication. This allows grouping of devices at the level of core network; thus, the MTDs within the same serving region of a core network gateway (i.e., which can include many macro BSs) can potentially be grouped together. Note that from the core network side, the communication from each of the devices in the same group will be considered as if it is coming from the same device which is turning on and off (i.e., establishing bearer and releasing it). The core network maintains only one bearer for them, thus achieves a huge resource saving. While this initial study [17] looks at the challenges and develops solutions (i.e., call flow updates) towards realization of such an approach, it does not study how the grouping of the MTDs should be made based on the traffic patterns of devices. Recently, we have looked at this problem, and developed a genetic algorithm based solution for grouping of MTDs with different data sending intervals [18]. In this study, we also take these approaches further and by considering some flexibility in the traffic patterns of IoT devices, we let the devices shift their traffic slightly so that more devices can be grouped together and the resource saving can be further increased. Moreover, we consider the dynamic nature of IoT environments where new IoT devices can join the network and some existing ones can leave the network and study the rearrangement of device groups to maintain resource saving at every network state/moment. The challenging part in all these scenarios is to design practical algorithms that have low complexities; thus we look for heuristic based solutions.

Our contributions[1] can be summarized as follows:

- We define the traffic shifting based aggregated IoT communication problem and develop the Integer Linear Programming (ILP) based models to solve it optimally at each network moment.
- We introduce a greedy heuristic based polynomial-time algorithm for grouping of devices at a given moment by leveraging a new metric based on traffic characteristics.
- We also provide another polynomial-time algorithm that rearranges the grouping of devices after new IoT devices join and some others leave the network.
- We provide extensive simulations to evaluate the proposed algorithms in various scenarios and show their benefits in resource saving.

The rest of the paper is organized as follows. We provide background information and discuss the related work in Section II. In Section III, we provide the system model and problem statement together with ILP formulations. In Section IV, we then elaborate on the heuristic based solutions. In Section V, we present the evaluation of the proposed solutions through simulations under different settings. Finally, we end up with conclusion in Section VI.

[1]The preliminary version of this study appeared in [19], in which we considered only the shifting of device traffic patterns without considering the dynamic environments.

II. Background

A. IMSI Sharing based Aggregated Communication

Overview. Subscriber identity sharing based connection and communication [17], [20] aims to efficiently use the core network resources by aggregating the traffic of multiple IoT devices which have usually low data rates and long data sending intervals. This is achieved by assigning a common IMSI number (which is used by MNOs to identify subscribers and is a key component of the Subscriber Identity Module (SIM) profile) to a group of IoT devices that have a common data sending interval and letting the core network consider them as the same device. The data communication of each device over this common connection line is achieved by having them take turns without overlapping their traffic patterns. There is also a recent patent application [21] by Qualcomm related to the development of apparatuses and methods for managing the subscription for a network of such wireless devices communicating in aggregation fashion.

Note that the IMSI sharing based aggregated communication reduces the utilization of core network resources such as the number of cellular bearers, for which there is usually a limit on core network gateways e.g., PGW in EPC. Considering all the IoT devices in the service region of a core network gateway, which usually covers hundreds of base stations or eNodeBs, it provides a resource optimization in a wider area compared to earlier approaches. On the other hand, in these studies [17], [20], only the devices that share a common data sending interval are considered and the list of devices that will share the same subscriber identity or IMSI (which is achieved at the initial provisioning of these devices with multiple instances of the same physical SIM) are pre-determined and not allowed to change. In a more recent work [18], this aggregation method has been extended considering all IoT devices with varying data upload cycles and with a dynamically determined list of devices that will share the same subscriber ID. Dynamic grouping of devices is achieved through new generation subscriber ID solutions including but not limited to virtual SIMs [22] and e-SIM cards [23], [24]. These solutions help subscribers change their mobile operators without changing their SIM cards but could easily be used for online provisioning of the network connectivity for IoT devices and assign them a new subscriber ID dynamically [25].

Call flow updates. Once the MTDs that will share the same connectivity (and subscriber ID) are determined by the MNO, the previous studies [17], [18], [20] address the necessary minimal changes that need to be made in the traditional call flows of several operations under this IMSI sharing model.

- *Device Attach.* When a new IoT device turns on, it sends an attach request to the core network. If the current time slot is in use by another IoT device that is sharing the same IMSI with this new device, its request is rejected and a new request is made after an assigned back-off timer expires. The procedure is repeated until a successful attachment is accomplished.
- *Data Communication.* The time is divided into equal slots and each device sharing the same link takes turns to connect and send their data to their corresponding

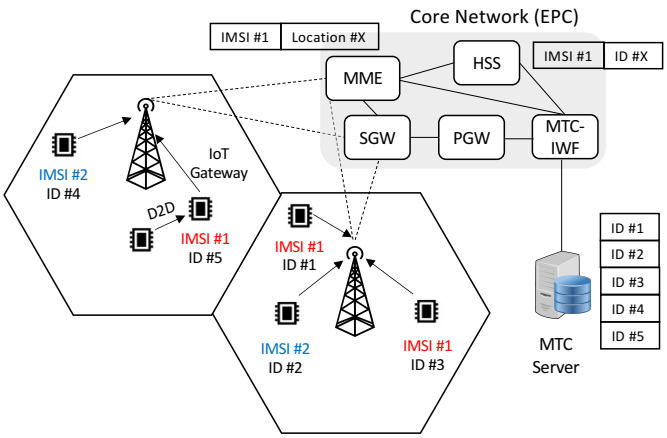


Fig. 1: Overview of IMSI sharing based aggregated IoT communication in EPC, as a representative of mobile core network.

destinations. A guard time is introduced between the time slots to avoid potential overlap that may occur due to delay in communication.

- *Paging.* Home Subscriber Server (HSS) coordinates with MTC server to keep track of the active IoT device of an aggregated cellular line, and manages the paging of the right device accordingly.

Consider the EPC network in Fig. 1, as a representative core network architecture which is currently the most common system in use. The IoT devices that share the same IMSI are considered as the same device by the core network. However, the list of the IoT devices using the same IMSI are still being tracked by the MTC server in the background through the usage of external identifiers (EID) and MTC interworking function (MTC-IWF) [13] that is serving as an intermediary function between the core network and the MTC server. Note that MTC server does not deal with IP addresses and cellular IDs (e.g., IMSI), which is managed by PGW, and just uses EIDs to communicate with the IoT devices. The mapping of IMSI and application port ID to EID is achieved through communication of MTC-IWF with HSS. The interested readers can refer to [17], [18], [20] for further details.

B. Related work

There are several studies in the literature that aim to address the increasing connection demand from massive number of IoT devices. These solutions include modifications and re-architecturing of core network and its functions [26], separating the control and user planes with Software Defined Networks (SDN) and Network Function Virtualization (NFV) (e.g., [27], MMLite [28], CleanG [29], [30], Softcell [31]) and device side based solutions (e.g., virtual bearers [32], group-based communication [33]). While some of these approaches are promising and yet to be tested in actual deployments, most of them come with some limitations for practical applications. For example, the solution proposed in [32] requires devices to be in D2D communication range of each other, and the solution proposed in [33] requires devices to be in the same eNodeB service area. Similarly, while a lightweight, functionally decomposed, and stateless MME design is proposed in [28], the optimization and resource saving happens in only one core network gateway, thus the solution is limited and does not provide benefit to the entire core network.

Study	Goal	Groups devices: connected to same	Groups devices: with traffic pattern	Dynamic network
[16]	Multi-cast communication for software updates etc.	eNodeB	same	No
[17]	Connect multiple devices under same IMSI	core network	same	No
[32]	Backhaul sharing through D2D communication	Local IoT gateway	any	No
[33]	Reduce signaling load	eNodeB	same MTD [70,75]/120	No
This study	Connect multiple devices under same IMSI	core network MTD [30,35]/60	varying and shifted MTD [20,24]/80	Yes

TABLE I: Comparison of proposed solution with closest previous studies that also aim traffic aggregation by grouping.

Different from these works, a more scalable and practical approach using IMSI sharing based aggregated connection and communication model is studied in [17], [18], [20] without changing the current architecture of core network drastically. The idea is to group a set of IoT devices and let them share the same subscriber identity and take turns for their actual data communication. Since the data communication happens infrequently for most of the machine type IoT devices (e.g., humidity measurement in field two times a day) and there is usually some flexibility especially when the collected data is not critical, we consider the shifting of scheduled communication times (to an earlier or later time) slightly to further decrease the number of active cellular bearers used. Moreover, we consider dynamic network environments where new IoT devices join and existing ones leave the network occasionally, and aim to maintain the grouping of devices as best as possible to increase resource saving. Note that the aggregated communication studied in this work is different from group-based or multi-cast communication considered in some previous work (e.g., [16]) as the latter usually considers simultaneous data (e.g., software updates) transmission toward a set of devices, thus cannot be applied for devices with different spatio-temporal traffic patterns. A summary of differences of this study from the other studies is also given in Table I.

III. System Model

A. Assumptions

Data traffic model. We assume that there is a set $G = \{I_1, I_2 \dots I_M\}$ of M IoT devices or MTDs[2] where each of them sends their data (e.g., measurements, computations) to their servers in some constant intervals. Their data sending intervals and the required connectivity duration within each of these intervals vary due to different application specific requirements but are known. To this end, we assume that for each device $I_i \in G$, the data upload happens at every λ_i time units and each data upload occurs for a duration of δ_i time

[2]We use IoT devices and MTDs interchangeably throughout the text.

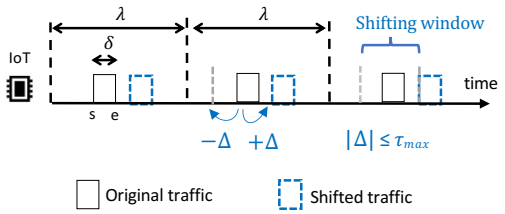


Fig. 2: Original and shifted traffic model for IoT devices.

units, starting at s_i and ending at e_i, within each λ_i duration (i.e., $\delta_i = e_i - s_i$), as shown in Fig. 2. We have chosen this model for simplicity, however, it could be extended to more complicated models (e.g., Gaussian distribution with a mean). We assume that the time is also divided into equal slots and all time related parameters are a multiple of the slot size so that the problem can be modeled in a discrete manner. We also assume that each MTD uses its own bearer initially, and after aggregation process they are partitioned into groups. The group of MTDs that use the i^{th} bearer is denoted by G_i, and for convenience, we will also refer to the i^{th} bearer as G_i.

Flexible traffic model. As it is shown in Fig. 2, we consider some slight shifts in the traffic pattern of the IoT devices. That is, the timing of each data upload instance for an IoT device can either happen earlier or later than its originally scheduled upload time without exceeding a given time threshold denoted with $\tau_{\max}$. Note that this threshold can be defined by the network management considering the application requirements (e.g., 5 minutes shifting for humidity measurements that is happening twice a day may be considered fine).

In the preliminary version of this study [19], we also considered an inconsistent model for the traffic pattern changes. That is, we let the individual data sending instances of the same IoT device to be shifted (i.e., delayed or scheduled earlier) differently without exceeding the threshold. While this gives more flexibility to the data uploads of the IoT devices, and hence provides an opportunity to group more IoT devices in the same cellular line, due to its complexity in modeling as well as only a slight benefit over consistently shifted traffic pattern model, we did not consider it in this extended version.

Dynamic network model. The number of IoT devices deployed and connected to the network of MNOs has been growing massively. Similarly, the existing IoT devices have been replaced, moved or upgraded. In order to model such a dynamic network environment, we first define each time frame without a change in the set of devices as a network *moment*, and use two parameters to define the node joins and leaves between consecutive moments. That is, we assume that x of the existing IoT devices in the current moment will be leaving the network and there will be y new devices will be joining the network in the next moment. Note that depending on the relation between x and y values the network size can be affected differently i.e., when $x < y$, the network size will grow; when $x > y$, it will shrink; otherwise it will stay the same. In any case, the existing group structure among IoT devices can be affected dramatically and regrouping of devices or introduction of new cellular lines may be needed to carry the traffic of all IoT devices.

Notations	Description
I_i	MTD or IoT device i
M	Number of MTDs
G (G^t)	The set of all MTDs (at time t)
G_i	The group of MTDs on i^{th} bearer, which is also denoted as bearer G_i.
G_{new}	The set of new MTDs joined to the network.
λ_i	Data sending interval of MTD i
δ_i	Duration of data communication in each data sensing interval for MTD i
s_i	Starting time of data communication within each interval by MTD i
e_i	Ending time of data communication within each interval by MTD i
$\mathcal{T}$ ($\mathcal{T}_j$)	Least Common Multiple (LCM) of data sending intervals (λ) of all MTDs (in group j)
x, y	Number of MTDs leaving and joining the network in every moment, respectively, in dynamic environments
b_j	Set to 1 if bearer j is used by at least one MTD and at any time (otherwise 0)
b_{jk}	Set to 1 if bearer j is used by at least one MTD at time slot k (otherwise 0)
$IMSI_j^t$	Temporary IMSI number or bearer ID assigned to MTD j at network moment t
b_{ijk}	Set to 1 if MTD i uses bearer j at the time slot k (otherwise 0)
$\tau_{\max}$	Maximum shifting allowed
$diff_j$	Set to 1 if IMSI number for MTD j is not equal to its IMSI number in previous moment.

TABLE II: Notations and their descriptions.

The notations used throughout the paper and their descriptions are summarized in Table II.

B. Problem Statements and ILP Models

Initial Network. The objective of aggregating the traffic from multiple MTDs through IMSI sharing is to minimize the number of active bearers used by all devices and optimize the resource usage in core network. When there is no shifting allowed in the originally scheduled traffic patterns of MTDs, the devices can still be grouped to some extent as long as there is no overlap in the traffic patterns of different devices in the same group. If the devices are allowed to shift their upload times slightly (i.e., less than $\tau_{\max}$) within their long data sending intervals, there will be more opportunity to decrease the number of groups and the number of actual bearers that will be used, and thus the resource saving will be increased. Using ILP, we define the problem (P1) of finding the optimal aggregation at the initial network moment considering the flexible traffic model (which can be shifted) as follows:

$$(P1):$$

$$\min \quad \sum_{j=1}^{M} b_j \tag{1}$$

$$\text{s.t.} \quad b_j = \min\left\{1, \sum_{k=1}^{\mathcal{T}} b_{jk}\right\}, \forall j \in [1, M] \tag{2}$$

$$b_{jk} = \min\left\{\sum_{i=1}^{M} b_{ijk}, 1\right\}, \forall j \in [1, M], \forall k \in [1, \mathcal{T}] \tag{3}$$

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/JIOT.2022.3155923, IEEE Internet of Things Journal

$$\sum_{i=1}^{M} b_{ijk} \leq 1, \forall j \in [1, M], \forall k \in [1, \mathcal{T}] \quad (4)$$

$$\exists! \Delta \in [-\tau_{\max}, +\tau_{\max}] : \sum_{d=1}^{\delta_i} b_{ij(r\lambda_i + ((s_i+\Delta+d) mod(\lambda_i)))} = \delta_i \quad \forall i, j \in [1, M], \forall r \in [0, \mathcal{T}/\lambda_i - 1] \quad (5)$$

$$b_{ij((r-1)\lambda_i+d)} = b_{ij(r\lambda_i+d)}, \forall d \in [1, \lambda_i] \quad \forall i, j \in [1, M], \forall r \in [1, \mathcal{T}/\lambda_i - 1] \quad (6)$$

where, $\mathcal{T} = LCM\{\lambda_1, \ldots \lambda_M\}$

$$b_{ijk} = \begin{cases} 1, & \text{if } I_i \text{ uses bearer } j \text{ at time slot } k, \\ 0, & \text{otherwise} \end{cases}.$$

Objective function in (1) aims to minimize the number of bearers used actively, where b_j is set to 1 if there is an MTD device using it. The usage of each bearer (there can be up to M total active bearers when each MTD uses a separate bearer) is determined by (2) and (3), by checking if there is at least one MTD using it at any time slot. Note that as the data sending intervals (λ) can be different for different MTDs, we first find the longest common multiple (LCM) of their data sending intervals and use it as a common timeline defined by $\mathcal{T} = LCM(\lambda_i, \forall i)$. (4) allows usage of each slot by a single MTD at most and (5) requires that there exists at least one and only one ($\exists!$) *shifting* amount (Δ) between $-\tau_{\max}$ and $\tau_{\max}$ which makes all δ_i consecutive slots utilized for the i^{th} MTD (i.e., I_i) at a given bearer j. Here r is used to find out the timing of repeated data upload times within the common timeline ($\mathcal{T}$). We also use (6) to make sure the shifting between the different data sending intervals of an MTD at the bearer it uses is achieved consistently.

Dynamic Network. The above model only solves the optimal grouping of the IoT devices (i.e., aggregation of their traffic) within a given network moment; thus, it will only be used at the beginning. In dynamic environments, we also need to consider the transition from the current moment to the next. Here, our goal is not only to decrease the number of bearers used but also to minimize the changes made in group structure from previous moment as any change in subscriber identity of existing devices requires reprovisioning of devices thus incurs some control traffic and delay. However, as the latter is a secondary goal, we use scalarization method to apply such prioritization in the objective function (7) of this second problem (P2) of finding optimal aggregation in dynamic environments as follows:

$$(P2): \quad \min \quad \left(\sum_{i=1}^{M} IMSI_i^t\right) \times \mathcal{L} + \sum_{i=1}^{M} diff_i \quad (7)$$

$$\text{s.t.} \quad IMSI_i = j, \text{if} \sum_{k=1}^{\mathcal{T}} b_{ijk} \geq 1, \forall i, j \in [1, M] \quad (8)$$

$$diff_i = \begin{cases} 1, & IMSI_i^t \neq IMSI_i^{t-1} \\ 0, & \text{otherwise} \end{cases}. \quad (9)$$

Here, (8) ensures that the devices on the same bearer uses the same IMSI number. We also simply assign the ID of the bearer (e.g., j) that the devices are on as the temporary IMSI number of these devices, which could be mapped to a real IMSI number from a SIM card. (9) finds the devices whose IMSI will change in the current moment (i.e., t) compared to previous one (i.e., $t-1$). In objective function (7), we use a constant $\mathcal{L}$ such that the sum of IMSI changes of all devices in the system will not affect the optimization more than decreasing the number of different IMSIs (i.e., groups or bearers) used by the devices. Note that in the first part of the optimization function we take the sum of IMSI numbers of all devices. This not only ensures that the number of groups used is minimized but also puts devices into bearers in order without leaving an empty bearer in between (e.g., to avoid using bearers 1 and 5 instead of 1 and 2). This design choice is used to minimize the IMSI changes of devices between consecutive moments.

An MNO knowing the traffic patterns of all devices can then run these ILP based optimal models to determine which IoT device will be in which bearer at every network moment and update their network registration information (e.g., IMSI numbers) through an online provisioning process as discussed in Section II. On the other hand, while these ILP models will find the optimal (i.e., minimum) number of bearers possible that can allocate all MTD traffic at the beginning and at every new moment with new set of MTD devices, respectively, their running time will be very long even with a small number of MTDs (e.g., 10-15) in the network. Thus, if the optimization models have to be run frequently (e.g., when the set of IoT devices or their traffic characteristics change often), it may not be a practical solution. To this end, in the next section, we provide heuristic based solutions with reduced complexities.

IV. Heuristic based Solutions

A. Initial Aggregation

1) Overview: In order to aggregate the traffic of multiple MTDs on the minimum number of bearers possible, we consider an iterative approach and try to select the best option at every step greedily. The overview of this process is provided in Fig. 3. Initially, we assume that each device is on a separate bearer or group. Then, we first find all eligible bearer pairs that can be merged. This is determined by checking if there is an overlapping allocated time slot by both of these bearers. Out of all eligible bearer pairs (having no overlap), we first find the pair that provides the highest *addition score (AS)* as follows:

$$(I_x^{max}, I_y^{max}) = \underset{\substack{\forall I_x, I_y \in G \\ I_x \neq I_y}}{\arg\max} AS(I_x, I_y). \quad (10)$$

Then, we merge these two bearers' traffic into one bearer (we call it *root bearer* and denote with G_{root}), and release the other one.

In consecutive steps, we check all other MTDs on their own bearers to see if they are eligible to be merged with this root bearer traffic. Among eligible ones, we find the one that gives the highest addition score and bring its traffic into the

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/JIOT.2022.3155923, IEEE Internet of Things Journal

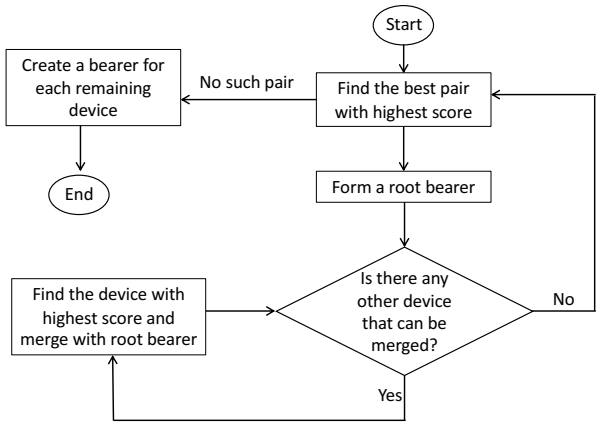


Fig. 3: Overview of heuristic based initial aggregation (HIA) procedure.

root bearer. That is, assuming G' denote the set of MTDs not merged in any other bearer yet, we find

$$I_z^{max} = \underset{\forall I_z \in G'}{\arg\max}\, AS(I_z, G_{root}). \tag{11}$$

This process continues until no more new MTD traffic is eligible to be merged into the current root bearer. Then, we continue the process with the formation of a new root bearer out of the remaining single-MTD bearers not aggregated yet. We again find the pair of bearers that gives the best score, merge their traffic on one of them and try to add other device/bearer traffic on this bearer one by one until no more eligible bearer remains. Here, note that, if there is no eligible pair of bearers that can be merged and assigned as root bearer, we stop the entire process and leave each of the single-MTD bearers as a separate bearer without any aggregation. A formal description of this greedy heuristic based approach is given in Algorithm 1. Root bearer formation is done in lines 4-11 and addition of other bearers on it one by one is achieved in lines 16-32. If no more root bearer can be formed, each remaining MTD is kept on its own bearer as shown in lines 35-39.

2) Addition Score (AS) Function: In this iterative and greedy heuristic based approach, the critical part is the score function. As our goal is to aggregate the traffic of as many MTDs as possible on a single bearer, at each aggregation step we aim at aggregating the bearers that will have a higher likelihood for the aggregation of others on the same bearer. To this end, after studying several criteria empirically, we ended up with the following three criteria:

- *Active Timeline* ($\mathcal{A}$): It is the duration from the first allocated time slot until the last allocated one. For bearer or group j, G_j, we find the minimum start time and maximum end time of all IoT devices on this bearer, and take the difference:

$$\begin{aligned} \mathcal{A}_j &= e^j_{max} - s^j_{min}, \text{where} \\ & s^j_{min} = \min\{s_i,\ \forall I_i \in G_j\} \\ & e^j_{max} = \max\{e_i,\ \forall I_i \in G_j\}. \end{aligned}$$

- *Utilization* ($\mathcal{U}$): It refers to the percentage of time slots allocated within the active timeline. Given that $b_{ijk} = 1$ when MTD i allocates bearer j at time slot k, for all MTDs on a given bearer or group j, G_j, we calculate

$$\mathcal{U}_j = \left(\sum_{k=s^j_{min}}^{e^j_{max}} a_k \right) / \mathcal{A}_j, \text{ where}$$
$$a_k = \begin{cases} 1, & \text{if } \exists I_i \in G_j \text{ s.t. } b_{ijk} = 1 \\ 0, & \text{otherwise} \end{cases}.$$

- *Border Score* ($\mathcal{B}$): This indicates how close the active timeline is to the end points of the entire timeline. As the allocated time slots get close to the sides of the entire

Algorithm 1: Initial Aggregation (G)

Input: G: Initial set of MTDs

1 $AS_{max} = 0$
2 $\alpha = 0$ // Next bearer id to assign MTDs
3 **while** $|G| > 0$ **do**
4 **foreach** (I_x, I_y) *s.t.* $I_x, I_y \in G, I_x \neq I_y$ **do**
5 **if** I_x *and* I_y *are eligible to be merged* **then**
6 **if** $AS(I_x, I_y) > AS_{max}$ **then**
7 $AS_{max} = AS(I_x, I_y)$
8 $(I_x^{max}, I_y^{max}) = (I_x, I_y)$
9 **end**
10 **end**
11 **end**
12 **if** $AS_{max} \neq 0$ **then**
13 $G_\alpha = \{I_x^{max}, I_y^{max}\}$
14 $G = G \setminus G_\alpha$
15 $E = G,\ \ AS_{max} = 0$
16 **while** $|E| > 0$ **do**
17 **foreach** $I_z \in E$ **do**
18 **if** I_z *can be merged on* G_α **then**
19 **if** $AS(I_z, G_\alpha) > AS_{max}$ **then**
20 $AS_{max} = AS(I_z, G_\alpha)$
21 $I_z^{max} = I_z$
22 **end**
23 **end**
24 **end**
25 **if** $AS_{max} \neq 0$ **then**
26 $G_\alpha = G_\alpha \bigcup \{I_z\}$
27 $E = E \setminus \{I_z\}$
28 $AS_{max} = 0$
29 **else**
30 $E = \emptyset$
31 **end**
32 **end**
33 $\alpha = \alpha + 1$
34 **else**
35 **foreach** $I \in G$ **do**
36 $G_\alpha = \{I\}$
37 $G = G \setminus G_\alpha$
38 $\alpha = \alpha + 1$
39 **end**
40 **end**
41 **end**

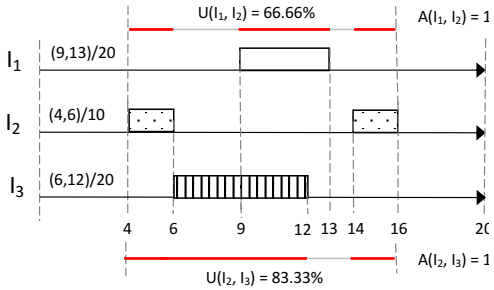


Fig. 4: Example: Only devices (I_1, I_2) and (I_2, I_3) a
to be merged on the same bearer as their traffic p
not overlap. They both have the same active time
but latter has larger utilization score, thus is selecte

timeline, the likelihood of allocating another M
same bearer increases. Thus, we first find the mi
distances to the start and end of entire timeline
start and end of active timeline and take their sur
for bearer or group j, G_j, we compute

$$\mathcal{B}_j \quad = \quad \min\{\mathcal{T}_j - s^j_{min}, s^j_{min}\} + \min\{\mathcal{T}_j - e^j_{max}, e^j_{max}\}.$$

Here, $\mathcal{T}_j$ is the timeline considered for bearer j and defined by $LCM(\lambda_i, \forall I_i \in G_j)$.

We consider these criteria in a prioritized manner with the following order:

$$\min(\mathcal{A}_j) \succ \max(\mathcal{U}_j) \succ \min(\mathcal{B}_j).$$

That is, we first aggregate the bearers that would result in a shorter active timeline after aggregation. Then, if there are multiple of those bearer pairs with the same active timeline, we prefer the pair that would provide higher utilization (due to either more MTDs involved or larger traffic served). Finally, if there is still a tie, we prioritize the bearer pair that would result in an active timeline closer to the borders. Consider the example in Fig. 4 with three MTDs. Here, we can either aggregate devices I_1 and I_2 or the devices I_2 and I_3, as I_1 and I_3 are not eligible to be merged due to overlapping traffic. Computing active timeline score for both, we get $\mathcal{A} = 12$. Then, we look at the second selection criteria of utilization, and we get $\mathcal{U} = (2+4+2)/12 = 66.66\%$, and $\mathcal{U} = (2+6+2)/12 = 83.33\%$, respectively. Thus, we prefer to aggregate I_2 and I_3 traffic. Note that border score for both cases is the same i.e., $\mathcal{B} = 4+4 = 8$, but we do not consider it in this example as it is the third criteria.

Running time. In Algorithm 1, there can be at most $\binom{M}{2}$ single-MTD bearer pairs that need to be checked to find the best candidate for a root bearer. If any other single device can be added to the current root bearer, the cost of finding the best one will be less than $\mathcal{O}(M)$. If none can be added to the root bearer and a new root bearer needs to be determined over score comparison of pairs in the remaining set of unassigned devices, there will be another $\binom{M-2}{2}$ pairwise comparison. The worst case scenario will happen if the process always continues with root bearer selection without adding any third device to the bearer, and it will generate a total of $\mathcal{O}(M^3)$ eligibility check and score calculations. Note that, the cost of score calculations does not change with shifting, but eligibility check cost increases. Without shifting, it only compares each time slot within $\mathcal{T}$ to see if there is an overlap, making overall complexity $\mathcal{O}(M^3\mathcal{T})$. However, with shifting, a shifting in range of $[-\tau_{max}, \tau_{max}]$ is considered for each device. Thus, it requires a comparison of $\mathcal{O}(\tau^2_{max})$ combinations, each with $\mathcal{T}$ cost, making the overall cost as $\mathcal{O}(M^3\mathcal{T}\tau^2_{max})$.

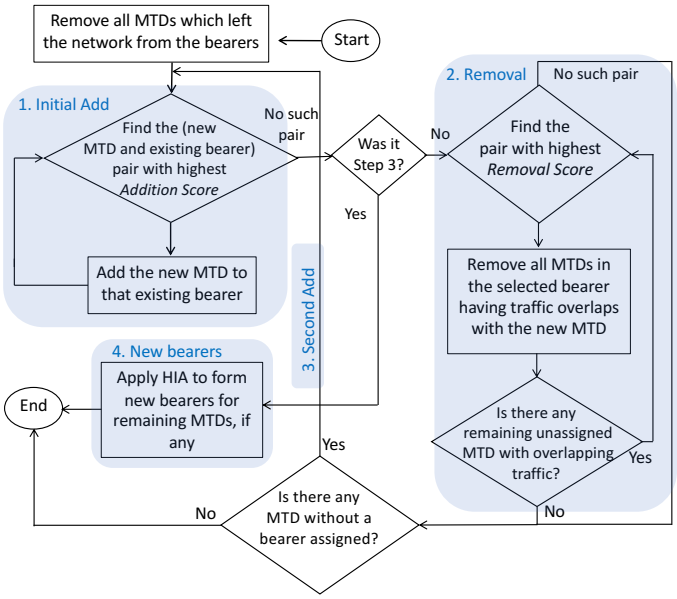


Fig. 5: Overview of heuristic based dynamic aggregation (HDA) procedure consisting of four steps.

B. Dynamic Aggregation during Transitions between Moments

1) Overview: When the set of devices or their traffic patterns change in a dynamic network environment, a new network moment starts; thus, existing groups or aggregated bearer formations should be revisited. We illustrate an overview of this process in Fig. 5. Note that at every moment, we could use Algorithm 1 to get a new grouping from scratch among the new set of devices without taking into account the previous group assignments of devices that are also present in the current moment. However, this might result in some unnecessary changes (i.e., new provisioning) in device IMSI assignments which come with additional delay and control traffic cost.

Thus, in order to mitigate this, when the new devices join the network and some existing ones leave, we first try to add newly joined MTDs to the available groups (step 1). This is performed similar to the process of adding other devices to the root bearer in Algorithm 1 (lines 16-32). However, this time we consider all possible new device (i.e., $I_z \in G_{new}$) and existing group (i.e., $G_j \in G_{cur}$) pairs and determine the order of adding by keep finding the pair with maximum score after every addition. That is, we find

$$(I^{max}_z, G^{max}_j) = \underset{\substack{\forall I_z \in G_{new} \\ G_j \in G_{cur}}}{\arg\max}\ AS(I_z, G_j), \tag{12}$$

and add I^{max}_z to G^{max}_j, and exclude I^{max}_z from G_{new}. We repeat this process until no more addition is possible.

Note that the first step may end up with locating each new device to an existing bearer if their traffic patterns do not

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/JIOT.2022.3155923, IEEE Internet of Things Journal

overlap (i.e., $G_{new} = \emptyset$). However, if that is not the case, we could ideally generate new bearers and assign the remaining new devices to them. But before doing that, as the second step, we first consider removing some of the existing MTDs temporarily in order to optimize the new bearer allocations. The steps of this *smart removal* process is given in Algorithm 2 (lines 2-23), in which we first find the pair of an existing group and a new MTD with the highest *removal score (RS)* (lines 5-13) and then remove the MTDs in that group that overlap with this new MTD (line 16) and put them into a set G_{tba} of MTDs (together with the new MTD) that will need to be assigned a new group id and IMSI together (lines 17-18). We continue similarly by finding the next best pair until no more overlap is found. Note that the step 2 will end either due to the processing of all new MTDs joined or if no more overlaps exist between existing groups and remaining new MTDs. If it is the latter case, we carry all remaining new MTDs to the set of MTDs to be assigned a new group (line 23). Then, we start the recursive addition process again (i.e., step 3) until no more addition is possible after which we start step 4 and create new bearers for the remaining devices (as in lines 35-39 in Algorithm 1).

2) Removal Score (RS) Function: The critical part during this process is the removal score function, for which we consider the following three criteria:

- *Count of Intersecting MTDs ($\mathcal{C}$):* It is equal to the number of MTDs in an existing bearer j's timeline having a data sending interval intersection with the newly joined MTD. That is, assuming that I_{new} is temporarily assigned to G_j and abusing the notation b_{ijk} (which is set to 1 when MTD i allocates bearer j at time slot k), we get

$$\mathcal{C}_j(I_{new}) = |\{I \in G_j \mid \exists k \in \mathcal{T}, b_{I_{new}jk} = 1, b_{Ijk} = 1\}|. \quad (13)$$

  Removing the MTDs from a bearer with more number of intersecting MTDs provides more opportunity to allocate them in other bearer options (during final addition process) and helps reduce the total number of bearers. This is because, each of these removed MTDs can be assigned to different bearers, providing more efficient bearer allocation opportunity. In Fig. 6, bearer G_j has two MTDs (I_1, I_2) intersecting with the new MTD.
- *Duration of Intersection ($\mathcal{D}$):* This refers to the portion of intersecting data sending intervals between the new MTD and an existing bearer j's timeline. That is:

$$\mathcal{D}_j(I_{new}) = \sum_{\forall k \in \mathcal{T}} \mid (b_{I_{new}jk} + b_{jk} = 2). \quad (14)$$

  Recall that b_{jk} is set to 1 when bearer j is used by at least one MTD at time slot k. The lesser the duration of the intersection, the more likely it is that removing MTDs from that group will help reduce the total number of bearers. This is because smaller intersection gives more chance for further aggregation especially when shiftings are considered. In Fig. 6, $\mathcal{I}_{new}$ has intersection from time slot 6 to 14 with MTDs that are already in the bearer j's timeline.

Algorithm 2: Dynamic Aggregation (G_{cur}, G_{new})

Input: G_{cur}: Set of existing groups of MTDs
G_{new}: Set of new MTDs joined

1 Keep merging (I_z^{max}, G_j^{max}) from (12) until no more possible.
2 $RS_{max} = 0$
3 $G_{tba} = \emptyset$ // Set of MTDs to be assigned a group
4 **while** $G_{new} \neq \emptyset$ **do**
5 **foreach** $I_{new} \in G_{new}$ **do**
6 **foreach** $G_i \in G_{cur}$ **do**
7 **if** I_{new} *overlaps with an MTD in* G_i **then**
8 **if** $RS(I_{new}, G_i) \geq RS_{max}$ **then**
9 $RS_{max} = RS(I_{new}, G_i)$
10 $(I_{best}, G_{best}) = (I_{new}, G_i)$
11 **end**
12 **end**
13 **end**
14 **end**
15 **if** $RS_{max} \neq 0$ **then**
16 Remove each MTD in G_{best} that overlaps with I_{best} and add to G_{tba}
17 $G_{tba} = G_{tba} \cup I_{best}$
18 $G_{new} = G_{new} \setminus \{I_{best}\}$
19 **else**
20 $break$
21 **end**
22 **end**
23 $G_{new} = G_{tba} \cup G_{new}$
24 Keep merging (I_z^{max}, G_j^{max}) from (12) until no more possible.
25 Add each remaining device to one individual bearer as in lines 35-39 of Alg.1.

- *Duration of Non-intersection ($\mathcal{E}$):* This is the non-intersecting duration of data sending intervals of intersecting MTDs and the new MTD, which is defined, for bearer j, as

$$\mathcal{E}_j(I_{new}) = \sum_{\forall k \in \mathcal{T}} \mid (b_{I_{new}jk} + b_{jk} = 1). \quad (15)$$

  The more duration of non-intersection, the more likely it is that removing MTDs from that group will help reduce the total number of bearers. The main reason for this is that by removing these MTDs from existing groups, more space can be freed thus more unassigned devices can fit in. In Fig. 6, non-intersecting parts are from time slot 4 to 6 and also from 14 to 18.

We consider these criteria again in a prioritized manner as in addition score calculation using the following order:

$$\max(\mathcal{C}_j) \succ \min(\mathcal{D}_j) \succ \max(\mathcal{E}_j).$$

That is, we first prefer the cases that provide more number of intersecting MTDs. If there is a tie, next, we consider the one with lesser intersection duration. If the tie does not break, we then select the one with more non-intersection duration (a random selection is made if the tie continues).

Running time. In Algorithm 2, during the initial addition process (line 1), in the worst case, all MTDs could be in a

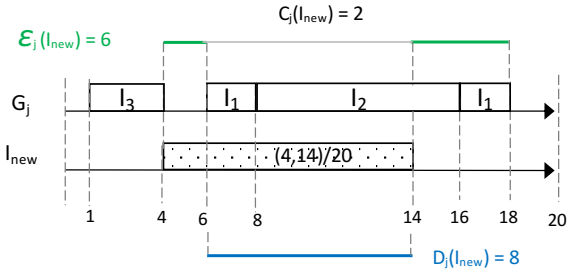
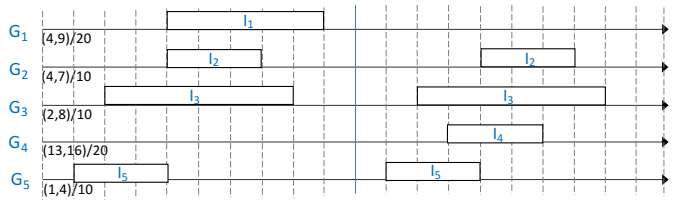


Fig. 6: Example scenario for smart removal process between an existing bearer and group (G_j) from previous moment and a new MTD (I_{new}) joined.

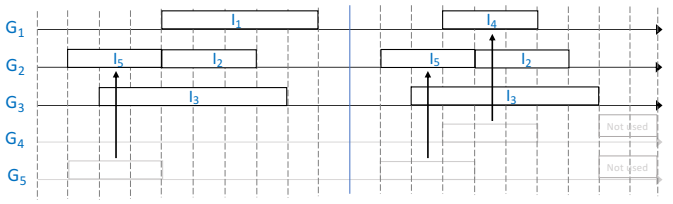
separate bearer (i.e., M bearers from previous moment) and with joining of y new devices, we may need to check all yM pairs over a timeline of $\mathcal{T}$ duration. With shifting allowed, this has $\mathcal{O}(y\tau_{max}^2 M\mathcal{T})$ complexity. It is possible that in the worst case, most of the new MTDs may fit to the existing bearers one by one and later need to be removed (e.g., due to one very long MTD) during smart removal process. Selection of which one will be added next requires calculation of scores for each of the remaining new MTD and existing bearer pair combinations (i.e., $(y-1)M, (y-2)M \ldots$). Overall cost of initial addition can then get close to $\mathcal{O}(y^2\tau_{max}^2 M\mathcal{T})$. In the removal process, in the worst case, we can remove all MTDs in existing groups one by one, which can cost $\mathcal{O}(y^2 M\mathcal{T})$, as we do not consider shiftings during removal process. Finally, after a removal process that ends up with removing all MTDs from all bearers, we start the last addition process (lines 24-25) and get new bearers similar to Algorithm 1 with complexity $\mathcal{O}(M^3\mathcal{T}\tau_{max}^2)$. Thus, the overall complexity of this deterministic Algorithm 2 per network moment is $\mathcal{O}(M\mathcal{T}\tau_{max}^2(M^2+y^2))$. However, this complexity can be improved further for more scalability by computing both the addition and removal scores of considered pairs at every step in parallel as they will be independent. This can reduce the overall complexity to $\mathcal{O}(M\mathcal{T}y)$, which includes reduced Algorithm 1 complexity of $\mathcal{O}(M\mathcal{T})$.

C. Toy Example

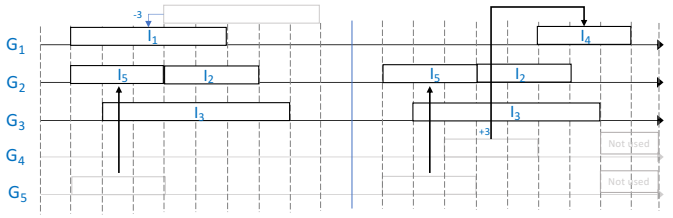
In this part, we provide how the proposed algorithms work on an example set of MTDs and two network moments. We consider a set of 5 MTDs, each of which initially uses a separate bearer (e.g., I_1 on bearer G_1) as shown in Fig. 7a. The traffic patterns are also shown in Fig. 7a. That is, for example, I_1 is sending its data between 4-9th time units in every 20 time units. As the LCM of the data sending intervals of these 5 MTDs is 20, we show all the repetitions of data communication for each device in this entire common timeline. When the initial aggregation algorithm is run with no shifting model, it finds that there are several eligible pairs that can be merged e.g., (I_2, I_5) and (I_1, I_4). Calculating their addition scores, the algorithm finds that (I_2, I_5) has the maximum score, thus merges their traffic on one of the bearers (i.e., G_2). Trying to add other MTDs on this root bearer does not help further, thus a new pairwise checking process starts among the remaining MTDs (i.e., I_1, I_3, I_4). In the second iteration, (I_1, I_4) is selected and merged to form a new root bearer (i.e., G_1).



(a) Original traffic without grouping



(b) Grouping MTDs with no shifting



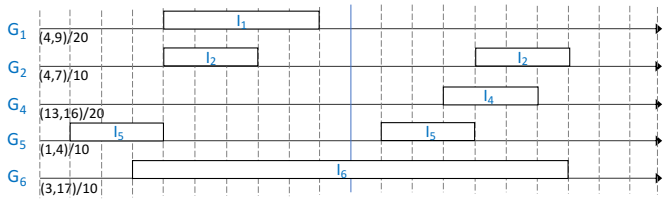
(c) Grouping MTDs with shifting

Fig. 7: Initial moment of the network with five MTDs and active bearer utilization in different scenarios.

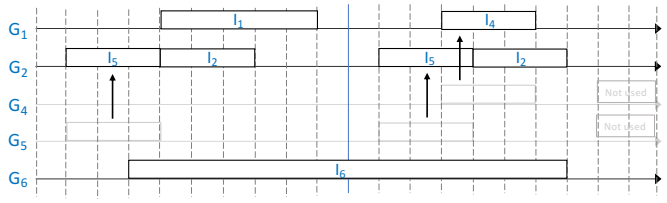
As I_3 is the only remaining and cannot be added to this root bearer and no new root bearer can be formed, the algorithm concludes and keeps I_3 on a separate bearer (i.e., G_3). This concludes the initial aggregation process without shifting with 3 active bearer usage for 5 MTDs' traffic as shown in Fig. 7b.

When traffic shifting is considered, with $\tau_{max} = 3$, we consider shifting of each MTD's traffic in range of $[-\tau_{max}, +\tau_{max}]$ during each pairwise merge eligibility check of MTDs. This time, in addition to the previous two pairs found in no shifting case, thanks to the flexibility through shifting, the algorithm also finds two more pairs that are eligible to be merged, namely, (I_2, I_4) and (I_4, I_5). However, (I_2, I_5) still provides the best score, thus is selected to form the initial root bearer. In the second iteration of the algorithm, as there is only one eligible pair left (i.e., (I_1, I_4)), it is selected and its MTDs are merged on bearer G_1. Remark that the data patterns of I_1 and I_4 have been shifted by -3 and +3 time-slots, respectively, to free more space for the future additions (due to the effect of border score). As no other MTD can be added to this root bearer, and there is only one MTD (i.e., I_3) left not merged with others yet, the process ends with keeping I_3 on a separate bearer, i.e., G_3, as in previous case.

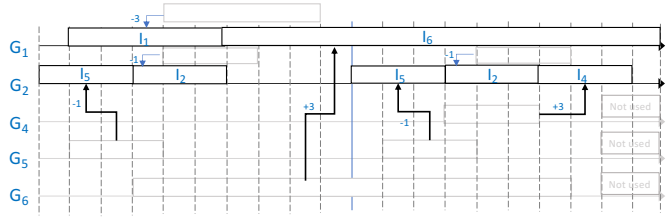
After this initial aggregation, we make one of the MTDs (i.e., I_3) leave and a new MTD (i.e., I_6) join the network, as shown in Fig. 8, and run the dynamic aggregation algorithm



(a) Original traffic without grouping



(b) Grouping MTDs with no shifting



(c) Grouping MTDs with shifting

Fig. 8: Next moment of the network and active bearer utilization after the leave of I_3 and the joining of I_6.

in Algorithm 2. In the no shifting scenario, with the leave of I_3, only two active bearers (G_1, G_2) are used. Since I_6 cannot be added to these bearers, we start the smart removal process. As I_6 overlaps with both groups/bearers G_1 and G_2, we find the one that is preferred based on the removal score. The new MTD I_6 has overlaps with 2 MTDs in each of the bearer's timeline; thus, in terms of the first metric in the removal score function they are equal. Then, we look at the second priority (i.e., duration of intersection) and prefer G_1 as it gives a smaller intersection duration with the new MTD (i.e., 8 vs. 10). We then remove all MTDs in G_1 and start the process of adding unassigned devices (i.e., I_1, I_4, I_6). As we cannot add any of them to the only remaining active bearer (i.e., G_2), we look for pairs of them to form a root bearer. As only I_1 and I_4 are eligible to be grouped without overlap, we put them into a new root bearer (i.e., G_1). We cannot add I_6 to this bearer, thus it is kept in its own bearer. This process then ends up with locating these devices into three bearers, as shown in Fig. 8b.

With traffic shifting using $\tau_{max} = 3$, after I_3 leaves (from the state of the bearers in Fig. 7c), only two bearers are used. With the joining of I_6, as we cannot fit it into bearers G_1 and G_2, we start the smart removal process. For that, considering all possible shiftings in range of [-3, +3] for I_6, we calculate the smart removal scores of it with each active bearer. This time, when I_6's traffic is shifted by +3 time units, we end up with having only one overlapping device (i.e., I_4) in bearer G_1 with I_6 (both MTDs' traffic in G_2 overlap with I_6 for all possible shiftings). Then, we remove I_4 from bearer G_1, and start addition process for unassigned devices (i.e., I_4, I_6). Trying to add them to the existing bearers, we first select I_6 to add bearer G_1 as it provides the highest addition score (with +3 shifting). We then can also add I_4 to bearer G_2 with +3 shifting. This process then ends up with locating these devices' traffic into two bearers, as shown in Fig. 8c.

Note that, in the last scenario, if we were to try to fit the new MTD without removing any existing MTDs (i.e., I_4 in this case), we would end up with 3 bearers. This shows the benefit of smart removal process in reducing the active bearer count further. It is also worth remarking that when we run ILP based solution on this example, we also receive the same number of bearer usage in each setting as in heuristic based solutions. The proposed algorithms may not always find the optimal solution as the ILP solutions, however, as it will be shown in simulation results, they can provide close to optimal results in most of the settings.

	Traffic Load		
Parameter	**Low**	**Medium**	**High**
Data communication per interval (δ in % within λ)	10-15%	15-25%	25-50%
Number of MTDs (M)	5-500		
Maximum shifting allowed (τ_{max})	0-6 time slots		
Data sending interval (λ) array	{10,20,40} time slots		
Start time for data sending (s_i)	Uniformly distributed in λ_i		
End time of data sending (e_i)	$s_i + \delta_i$ if it is $\leq \lambda_i$		
Dynamicity	10-50%		

TABLE III: Simulation parameters and values.

V. Simulation Results

In order to evaluate the performance of the proposed solutions, we perform simulations in different settings. We also compare the heuristic based approximate solutions with the optimal solutions obtained by CPLEX from ILP models.

A. Settings

Following the traffic model introduced in Section III, we first generate a data upload traffic for each MTD. To this end, we set a data upload interval (λ_i) randomly selected from the set $\{10, 20, 40\}$ minutes. Then, we randomly assign a data communication duration, $\delta_i = s_i - e_i$, within each data upload interval using three different traffic load models. In the low traffic load model, we assume 10-15% of the data sending interval or λ_i is used for data communication, and we use 15-25% and 25-50% for medium and high traffic loads, respectively. The start time of the data upload (s_i) within the data sending interval is determined randomly from $[0, \lambda_i - \delta_i]$. The end time of data communication is then set to $e_i = s_i + \delta_i$ automatically. Throughout simulations, we use an MTD count ranging from 5 to 500. In particular, for comparison with ILP based optimal results, we use smaller M values as getting ILP results takes very long with large number of MTDs. For heuristic only results, we consider MTD counts as high as 500

and look at the impact of various parameters. For the dynamic network scenarios, we also consider a dynamicity level which is defined as the percentage of devices join/leave at every moment. For main simulations, we consider an equal number of joins and leaves (i.e., $x = y$) at every network moment, but we also look at a non-equal case. Table III provides a summary of the simulation parameters and their values.

B. Algorithms in Comparison

We compare the performance of the following algorithms.

- *ILP-based Optimal Initial Aggregation (ILP-IA)*: This is the solution of ILP-based model given in (1), which considers only initial aggregation.
- *ILP-based Optimal Dynamic Aggregation (ILP-DA)*: This is the solution of ILP-based model given in (7) considering the dynamic aggregation throughout the network moments.
- *Heuristic-based Initial Aggregation (HIA)*: This is the heuristic-based initial aggregation algorithm given in Algorithm 1.
- *Heuristic-based Dynamic Aggregation (HDA)*: This is the heuristic algorithm given in Algorithm 2.
- *Heuristic-based Dynamic Aggregation without Smart Removal (HDA_noSR)*: This refers to the variant of the heuristic-based dynamic aggregation algorithm without smart removal process (i.e., lines 4-22 in Algorithm 2).

We consider both shifting and no shifting based aggregation scenarios for each of these algorithms.

Note that while aggregated IoT communication has previously been studied in [17], [20] with a no shifting model, their solution assumes that only MTDs with the same data communication duration (δ) within the same data sending interval (λ) will aggregate their traffic on the same bearer. These studies mainly focus on call flow updates to realize IMSI sharing based aggregated communication and do not propose how to actually group IoT devices if their traffic patterns are different as well as how the groupings should be updated in dynamic environments. Thus, these solutions are not directly applicable to our setting as we allow MTDs with varying λ and δ. However, no shifting case (i.e., $\tau_{max} = 0$) especially in static case, or the algorithm *ILP-IA* can be considered as an upper bound for the performance of these benchmark solutions (as in [17], [20] only the devices with same λ and δ are grouped together) and can be used to understand the additional savings offered by shifting based solutions in static case. Note that other solutions [16], [33] that consider group-based communication for IoT devices are also not applicable to our setting, as they consider devices within the service area of the same base station only and they target multi-cast transmission of only certain types of data (e.g., software updates) [16] or group-based RRC connection establishment and release for a set of homogeneous machine type devices owned by the same company for mainly reducing the signaling load [33], assuming that the devices that will be grouped are pre-determined.

C. Performance Metrics

We evaluate the performance of proposed solutions based on the following metrics:

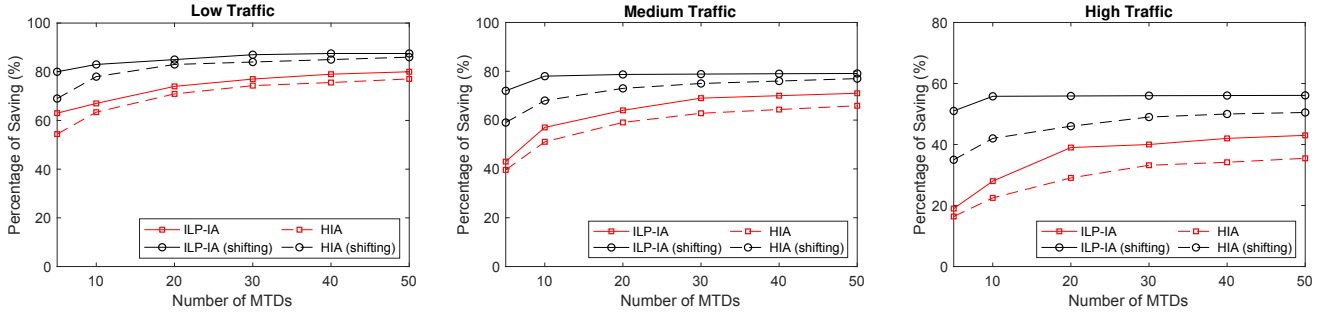
- *Percentage of Saving (%)*: This is defined as the saving in the number of cellular lines (i.e., bearers) utilized. For a given number of MTD devices M, if the aggregation model ends up finding that the number of bearers sufficient to carry the traffic from all of these M devices is X, then the percentage of saving is defined as
$$\left(\frac{M-X}{M} \times 100\right)\%. \tag{16}$$
- *Percentage of MTDs with Updated IMSI*: This is the average percentage of MTDs whose IMSI has changed between consecutive moments in dynamic network scenarios. When some existing MTDs leave the network and some new ones join, reorganizing the groups may help benefit from aggregated communication properly. However, the new grouping structure may require some existing devices change their groups, which triggers a control data traffic for reprovisioning of these MTDs with new IMSI numbers. Keeping such difference in group assignments and associated control traffic as minimum as possible is a secondary goal after maximizing the saving. Note that we consider the IMSI changes only for the MTDs that exist in both network moments and define their percentage as
$$\left(\sum_{t=2}^{Z}\left(\left(\sum_{\forall I_i \in G^t} diff_i\right)/|G_t|\right)\right)/\left(Z-1\right), \tag{17}$$
where Z is the number of moments of the network with different MTDs, and G^t is the set of MTDs at network moment t.
- *Running time*: In order to show the scalability of the algorithms, we also present their running times with increasing number of MTDs on an Intel core i7 processor with 16 GB memory and 2.5 GHz speed.

We look at the impact of different traffic load models, number of MTDs, dynamicity of the network, and maximum shifting allowed (i.e., τ_{max}) on these metrics. All results presented are averaged over 20 runs.

D. Results

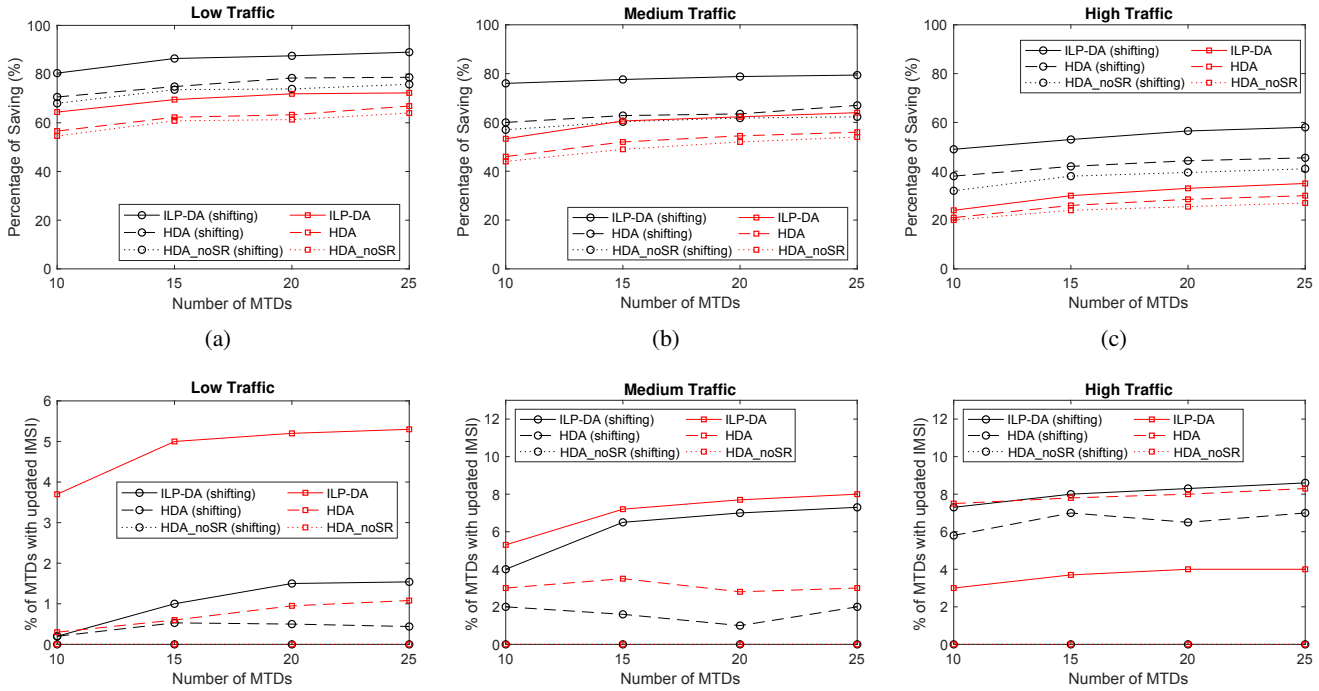
1) Comparison of ILP and Heuristic Solutions: Initially, we compare the ILP based optimal solutions with heuristic based solutions. We first look at the initial aggregation process and grouping of MTDs when they first join the network. Fig. 9 shows the percentage of saving obtained by both the ILP and HIA algorithms for different traffic load models as the number of MTDs in the network increases. First of all, as it is seen in all three graphs, the percentage of saving increases as the MTD count increases and converges to a certain value. Even though we did not obtain results beyond 50 MTDs due to the long running time of ILP, this is not needed as the saving converges already. Comparing the savings achieved, we observe that the highest percentage of saving is achieved when the traffic load is low. Moreover, the percentage of saving increases for all cases as the number of MTDs increases. These are because low

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/JIOT.2022.3155923, IEEE Internet of Things Journal



(a) (b) (c)

Fig. 9: **ILP vs. Heuristic Algorithms** in initial aggregation: Percentage of saving in the initial network moment with a) low, (b) medium and (c) high traffic models ($\tau_{max} = 3$ for shifting based aggregation).



(a) (b) (c)

(d) (e) (f)

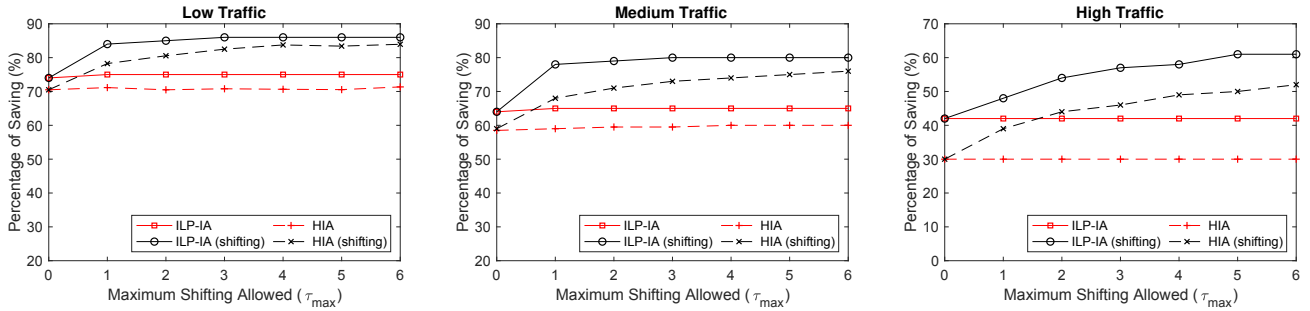
Fig. 10: **ILP vs. Heuristic Algorithms** in dynamic aggregation: (a-c) Percentage of savings and (d-f) Percentage of MTDs with updated IMSI averaged over 100 moments with 10% dynamicity ($\tau_{max} = 3$ for shifting based aggregation).

traffic model gives more opportunity to group more MTDs into a single bearer and more MTD count increases this opportunity further for a given traffic load model, respectively. However, the rate of increase in saving varies in different traffic loads.

Comparing no shifting and shifting based solutions, we clearly see that shifting offers more saving in all cases thanks to the flexible data upload times of MTDs. Looking at the comparison of ILP and heuristic solutions, in general we observe that heuristic solutions can provide close to ILP results. The difference between heuristic and ILP results however gets larger in high traffic case (for the given data points, on average 9.75% and 6.7% absolute saving difference with and without shifting, respectively, while medium and low traffic have (6.33%, 5.22%) and (4.17%, 4.09%) for the same, respectively.), as it gets harder for the heuristic solution to find better groupings with highly utilized timelines of MTDs.

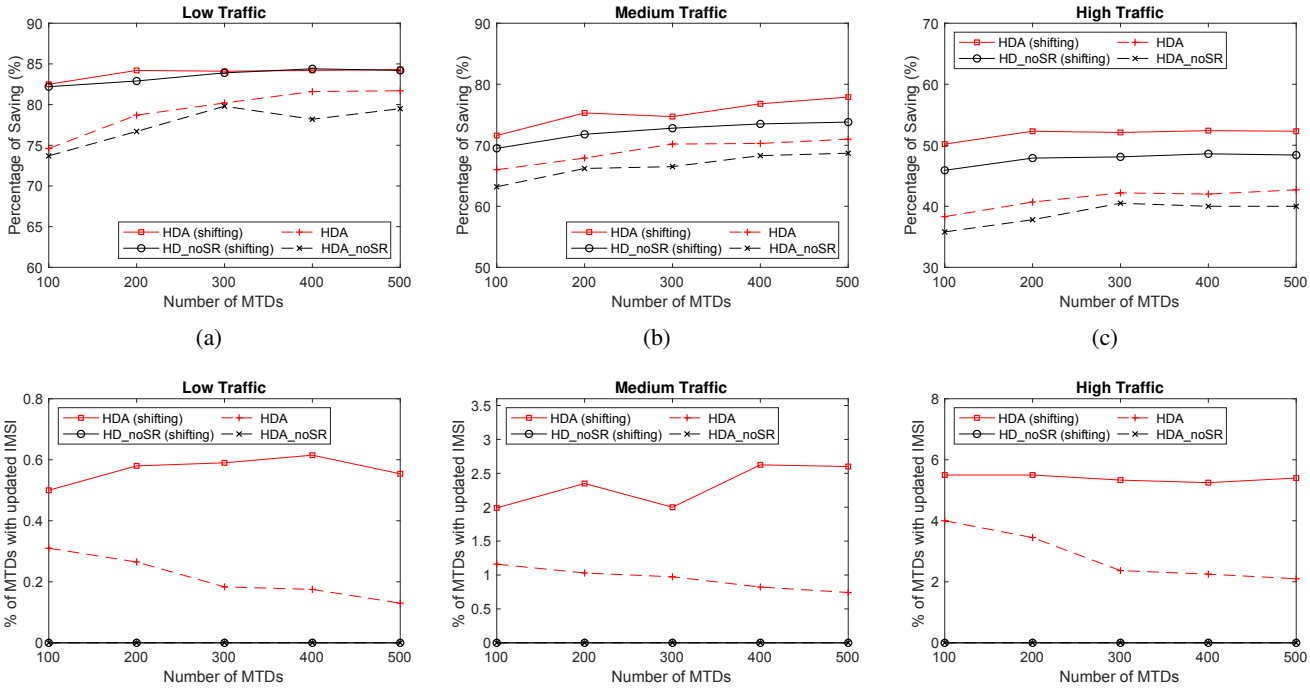
Next, we compare the ILP solution with heuristic based solutions in dynamic environments. Fig. 10 shows the results with different number of MTDs in three traffic models. Here, we show results until 25 MTDs as running ILP-DA takes much longer than ILP-IA. For each MTD count, we performed simulations over 100 moments with 10% of the existing MTDs leaving the network and an equal amount of new MTDs with new traffic patterns joining the network at every new moment. Looking at the saving results in Fig. 10a, we notice a similar relation as in Fig. 9a, but the gap between ILP-DA and HDA is a bit larger (on average 10.2% and 7.3% absolute saving difference with and without shifting, respectively). This is probably due to the fact that ILP-DA can achieve slightly higher saving compared to ILP-IA, but this comes with more MTDs changing their IMSI through different moments compared to HDA, as shown in Fig. 10d. HDA_noSR algorithm

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/JIOT.2022.3155923, IEEE Internet of Things Journal



(a) (b) (c)

Fig. 11: **ILP vs. Heuristic Algorithms** with different maximum shifting threshold ($\tau_{\max}$): Percentage of saving with (a) low, (b) medium and (c) high traffic patterns (M = 20, τ_{max} = 3).



(a) (b) (c)

(d) (e) (f)

Fig. 12: **Impact of MTD count:** The percentage of savings in dynamic environments (10% dynamicity) with (a) low, (b) medium and (c) high traffic patterns (τ_{max} = 3), and corresponding percentages of MTDs with updated IMSI counts in (d), (e) and (f), respectively.

provides slightly less saving compared to HDA, showing the benefit of smart removal process. This benefit gets more clear with more traffic, as shown in Fig. 10b-c, and with large number of MTDs in the network, as will be shown later. Comparing percentage of MTDs with updated IMSI, we see some differences. While HDA algorithm with shifting always results in more such percentage compared to its no shifting run, ILP-DA results vary with different traffic loads. That is, in low and medium traffic (Fig. 10d-e), no shifting based ILP-DA results in more MTDs with updated IMSI compared to the ILP-DA with shifting, however this gets opposite in high traffic. As minimizing the percentage of MTDs with updated IMSI is the secondary goal after minimizing the active number of bearers used, this difference can be understandable. It is also worth remarking that HDA_noSR algorithm does not cause any IMSI update for existing MTDs as it only adds the new joining MTDs to the available bearers or initiate new ones.

Fig. 11 shows the impact of τ_{max} on percentage of saving. Here, we again use a small MTD count (i.e., M = 20) to be able to show a comparison with ILP results. In the case of no shifting, the results do not change but we provide them to show the benefit of shifting based models over this benchmark model clearly. We see that as threshold increases, there is more saving achieved in all traffic load models. However, we see that in low traffic, the convergence happens more quickly than in medium traffic whose convergence happens more quickly than in high traffic case. This is because as the traffic density gets higher, it becomes less flexible for arrangements among groups and thus can only achieve the maximum benefit possible with more flexibility obtained when larger thresholds are used.

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/JIOT.2022.3155923, IEEE Internet of Things Journal

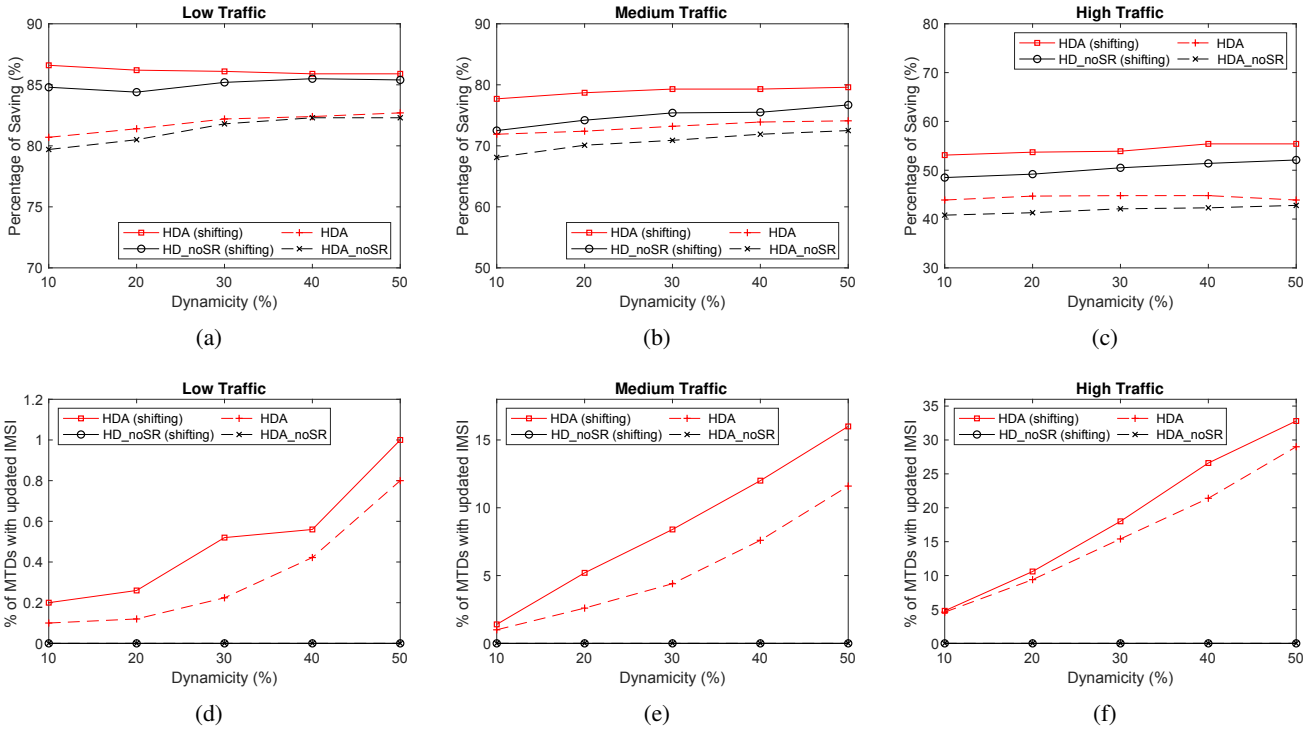


Fig. 13: **Impact of dynamicity:** The percentage of savings with (a) low, (b) medium and (c) high traffic patterns (τ_{max} = 3, $M = 500$), and corresponding percentages of MTDs with updated IMSI counts in (d), (e) and (f), respectively.

Moreover, heuristic based solution in general provides closer results to ILP solution. However, as the traffic density gets higher, the gap between heuristic and ILP results increases similar to the earlier results presented.

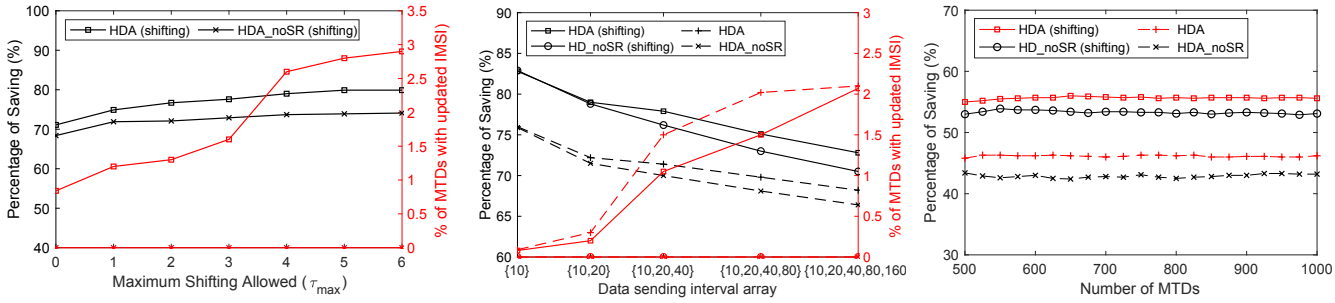
2) Impact of MTD Count: In order to show how the proposed algorithms perform with larger number of MTDs, we also obtained results from 100 to 500 MTDs in the increments of 100. These results are presented for dynamic environments and do not include ILP results due to longer running times. In Fig. 12, we show both the percentage of saving and percentage of MTDs with updated IMSI for three traffic models. The percentage of saving results in Fig. 12a-c show the benefit of shifting as in Fig. 9. The saving is also more or less stable in each of the traffic models. Comparing HDA and HDA_noSR algorithms, we also observe a much clearer benefit of smart removal process included in HDA, as the traffic load increases. This is also true for both shifting and no shifting cases. However, as it is shown in Fig. 12d-f, this comes with some changes in IMSI assignments of MTDs. For example, in high traffic model with shifting and when M=500, while HDA offers around 10% more relative saving compared to HDA_noSR, it causes around 5.4% of MTDs update their IMSI between consecutive moments. On the other hand, HDA_noSR does not cause any update in IMSI assignments of existing MTDs as expected.

3) Impact of Dynamicity: In Fig. 13, we look at the results with different dynamicity levels between consecutive moments. In particular, we consider from 10% to 50% dynamicity. When there are M=500 MTDs in the initial network, these refer to 50 and 250 MTDs joining/leaving at every moment, respectively. Looking at the percentage of saving results in Fig. 13a-c, we observe a more or less stable saving in all cases. HDA again offers larger saving compared to HDA_noSR and shifting helps increase this saving. On the other hand, HDA causes IMSI updates due to smart removal process, as shown in Fig. 13d-f. Note that when dynamicity increases the percentage of MTDs with updated IMSI increases, and in some cases, this gets very large and can cause a lot of control traffic (for provisioning of new IMSI numbers). However, in a real scenario even 10% dynamicity could be very high and we observe that with 10% dynamicity the percentage of MTDs with updated IMSI is relatively low (i.e., 0.2-5%).

4) Impact of Maximum Shifting Threshold: We then look at the impact of maximum shifting threshold (τ_{max}) in dynamic environments (similar results are shown for initial aggregation only while providing comparison with ILP in Fig. 11). Fig. 14a shows the results for τ_{max} in range of [0,6]. As the threshold increases, the percentage of saving first increases and becomes stable. HDA offers up to 10% additional relative saving compared to HD_noSR and this comes with up to 3% of MTDs with updated IMSI. If the control traffic associated with such IMSI updates can be handled by the network without affecting data traffic, then HDA can safely be utilized to increase saving in aggregated communication.

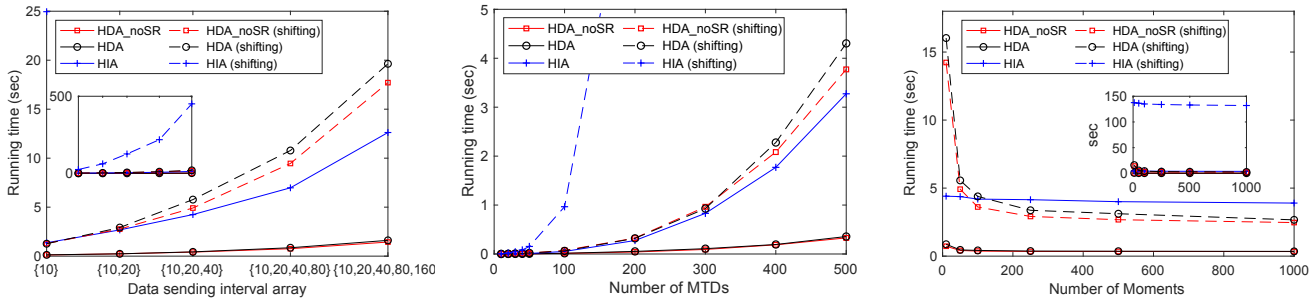
5) Impact of Data Sending Interval Array: In Fig. 14b, we look at the impact of the array from which the data sending intervals of the MTDs are selected on the results. As the figure shows, with more options and larger λ values, the saving reduces in all algorithms. However, in all cases, shifting as well as the smart removal process considered in HDA help

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/JIOT.2022.3155923, IEEE Internet of Things Journal



(a) (b) (c)

Fig. 14: **Impact of various parameters:** Percentage of saving with different (a) τ_{max} and (b) data sending interval arrays in dynamic scenarios (medium traffic, M=500, τ_{max}=3). (c) Percentage of saving in a growing network (high traffic).



(a) (b) (c)

Fig. 15: **Average running time comparison:** Using (a) different data sending interval arrays, (b) different number of devices in the network, and (c) different number of network moments (medium traffic, M=500, τ_{max}=3, dynamicity = 10%).

increase the saving. Up to 2% of MTDs get updated IMSI between consecutive moments on average, which refers to around 10 MTDs thus may not cause too much control traffic. In the preliminary version of this paper [19], we also showed a comparison of these results with ILP using a small MTD count and demonstrated that heuristic algorithms can provide closer results to ILP. These results are omitted here for the sake of brevity.

6) Impact of Growing Network: In earlier results with dynamic environments, we assume that an equal number of MTDs join and leave the network at every network moment so that total MTD count in the network stays stable. In order to see how results change in a growing network, we also obtained results. To this end, we start with M=500 MTDs in the network and let 50 devices leave and 75 new devices join at every moment. Thus, after 20 moments, the MTD count in the network reaches M=1000 devices. Looking at results in Fig. 14c, we see a pretty stable behavior in terms of percentage of saving. The percentage of MTDs with updated IMSI is also similarly stable (around 2%), which is omitted in the graph for clarity.

7) Running Time Comparison: Finally, in Fig. 15, we compare the running times of heuristic based algorithms (running times of ILP solutions are much higher as shown in preliminary version [19], thus skipped here). In these results, we consider a dynamic environment and run HIA algorithm for each moment independently (as if the network is initiated at that moment without considering previous moment). This indeed refers to the algorithm proposed in preliminary version [19] without considering dynamic environments. In Fig. 15a, we first compare running times with different data sending interval arrays. As the results show, running HIA at every moment takes very long time compared to HDA and HDA_noSR both when shifting is considered and not considered. Shifting causes running time increase in all algorithms due to additional computations needed to check all possible combinations to benefit from the flexibility offered by shifting. Comparing HDA and HDA_noSR, we observe that there is only some slight increase in running time with HDA compared to HDA_noSR due to the smart removal process in HDA. However, as it is shown in earlier results, HDA can provide up to 10% additional saving compared to HDA_noSR.

In Fig. 15b, we compare running times with different MTDs in the network (with 10% dynamicity). The results are inline with the results in Fig. 15a in terms of the order of algorithms with respect to running time. With more MTDs, the running time of HIA increases heavily compared to HDA and HDA_noSR. HDA and HDA_noSR also show similar running time, while HDA offers more saving. Finally, in Fig. 15c, we show the average running times per moment over different number of network moments. As the results show, the average running times of HDA and HDA_noSR algorithms decrease with more network moments. This is because in these

algorithms, there is indeed the high cost of initial grouping (using HIA) of all nodes at the beginning. As the network changes with joins and leaves and these algorithms are applied over more number of moments, their average running time per moment indeed gets lower thanks to the much lower cost of regrouping algorithm used (i.e., Algorithm 2). HDA has again slightly more average running time than HDA_noSR due to the smart removal process, however it is still much lower than applying from scratch grouping (i.e., HIA) at every moment.

VI. Conclusion

In this paper, we studied traffic shifting based aggregated communication model for IoT devices in dynamic environments. The proposed aggregated communication model not only lets the devices use the same subscriber identity (i.e., IMSI) and take turns during their communication but also considers slight shifting in the original traffic patterns of devices for further saving in the resource utilization, namely the number of actively used bearers, in the core network. We considered a dynamic environment where some existing devices leave the network and new ones join the network and form a new network moment. We aimed to aggregate the traffic from these devices as much as possible and also while keeping the bearer and IMSI assignments as stable as possible as the list of the devices in the network changes. To this end, we first modeled ILP based solutions and then in order to avoid the complexity of ILP solutions we proposed heuristic based aggregation algorithms with a much lower complexity. Simulation results showed that heuristic based solutions can offer closer results to ILP results with much less complexity and shifting based aggregation provides more saving in the number of bearers or cellular connections used to carry the traffic from all devices. Moreover, the smart removal process considered between consecutive network moments can offer additional saving compared to naive method of adding new arriving MTDs to the existing bearers directly and creating new ones for not fitting ones (i.e., HDA_noSR algorithm). These results show that the proposed HDA algorithm offers a scalable solution and can efficiently work in dynamic environments.

In our future work, we will consider more complicated and data-driven traffic models for the MTDs, and look at the performance of proposed solutions in real environments. We will also perform experiments for the IMSI sharing among MTDs in dynamic networks. Finally, we will consider erroneous and malicious behavior of the devices and study the robustness and security of the system.

VII. Acknowledgments

This work was supported in part by National Science Foundation awards CNS-1815603, CNS-1814727, and by Commonwealth Cyber Initiative (CCI).

References

[1] W. Ejaz, M. Naeem, A. Shahid, A. Anpalagan, and M. Jo, "Efficient energy management for the Internet of Things in smart cities," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 84–91, 2017.

[2] F. Montori, L. Bedogni, and L. Bononi, "A collaborative Internet of Things architecture for smart cities and environmental monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, 2017.

[3] A. Yang, C. Zhang, Y. Chen, Y. Zhuansun, and H. Liu, "Security and privacy of smart home systems based on the Internet of Things and stereo matching algorithms," *IEEE Internet of Things Journal*, vol. 7, no. 4, pp. 2521–2530, 2019.

[4] F. Guo, F. R. Yu, H. Zhang, X. Li, H. Ji, and V. C. Leung, "Enabling massive IoT toward 6G: A comprehensive survey," *IEEE Internet of Things Journal*, 2021.

[5] 3GPP, "Standards for IoT," Dec. 2016. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1805-iot_r14

[6] G. Zhu, J. Xu, K. Huang, and S. Cui, "Over-the-air computing for wireless data aggregation in massive iot," *IEEE Wirel. Commun.*, vol. 28, no. 4, pp. 57–65, 2021.

[7] B. Liu, C. Liu, and M. Peng, "Resource allocation for energy-efficient MEC in noma-enabled massive iot networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 4, pp. 1015–1027, 2021.

[8] A. Mukherjee, P. Goswami, M. A. Khan, L. Manman, L. Yang, and P. Pillai, "Energy-efficient resource allocation strategy in massive iot for industrial 6g applications," *IEEE Internet Things Journal*, vol. 8, no. 7, pp. 5194–5201, 2021.

[9] D. Zhai, R. Zhang, L. Cai, B. Li, and Y. Jiang, "Energy-efficient user scheduling and power allocation for noma-based wireless networks with massive iot devices," *IEEE Internet Things Journal*, vol. 5, no. 3, pp. 1857–1868, 2018.

[10] T. Lv, Z. Lin, P. Huang, and J. Zeng, "Optimization of the energy-efficient relay-based massive iot network," *IEEE Internet Things Journal*, vol. 5, no. 4, pp. 3043–3058, 2018.

[11] B. P. Sahoo, C.-C. Chou, C.-W. Weng, and H.-Y. Wei, "Enabling millimeter-wave 5g networks for massive iot applications: A closer look at the issues impacting millimeter-waves in consumer devices under the 5g framework," *IEEE Consumer Electronics Magazine*, vol. 8, no. 1, pp. 49–54, 2018.

[12] Y. Xiao, M. Krunz, and T. Shu, "Multi-Operator network sharing for massive IoT," *IEEE Communications Magazine*, vol. 57, no. 4, pp. 96–101, 2019.

[13] 3GPP, "Architecture enhancemens to facilitate communication with packet data networks and applications," v15, 2017. [Online]. Available: TS23.682

[14] N. Saxena, A. Roy, B. J. R. Sahu, and H. Kim, "Efficient iot gateway over 5g wireless: A new design with prototype and implementation results," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 97–105, 2017.

[15] O. A. Amodu and M. Othman, "Machine-to-Machine Communication: An Overview of Opportunities," *Computer Networks*, vol. 145, pp. 255–276, 2018.

[16] G. Tsoukaneri, M. Condoluci, T. Mahmoodi, M. Dohler, and M. K. Marina, "Group communications in narrowband-IoT: Architecture, procedures, and evaluation," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1539–1549, 2018.

[17] M. Ito, N. Nishinaga, Y. Kitatsuji, and M. Murata, "Reducing state information by sharing IMSI for cellular IoT devices," *IEEE Internet Things Journal*, vol. 3, no. 6, pp. 1297–1309, 2016.

[18] E. Bulut and I. Güvenç, "Dynamically Shared Wide-Area Cellular Communication for Hyper-dense IoT Devices," in *Proc. of the IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2018, pp. 64–69.

[19] A. Chapnevis, I. Güvenç, and E. Bulut, "Traffic Shifting based Resource Optimization in Aggregated IoT Communication," in *Proc. of 45th IEEE Conference on Local Computer Networks (LCN)*, 2020, pp. 233–243.

[20] M. Ito, N. Nishinaga, Y. Kitatsuji, and M. Murata, "Aggregating cellular communication lines for IoT devices by sharing IMSI," in *Proc. of IEEE International Conference on Communications (ICC)*, 2016, pp. 1–7.

[21] D. K. Vayilapelli, P. K. Darisi, R. P. Katakam, and R. Katkam, "Network management of subscriptions for IoT devices," Feb. 19 2019, US Patent 10,212,685.

[22] TechNews, "Xiaomi's MIUI Now Features Virtual SIM Card for Overseas Travels," 2017. [Online]. Available: http://technews.co/2015/03/25/xiaomis-miui-now-features-virtual-sim-card-for-overseas-travels/.

[23] McKinsey, "E-sim for consumers – a game changer in mobile telecommunications?" 2016. [Online]. Available: https://www.mckinsey.com/industries/telecommunications/our-insights/e-sim-for-consumers-a-game-changer-in-mobile-telecommunications

[24] GSMA, "Remote sim provisioning for machine to machine," 2017. [Online]. Available: http://www.gsma.com/connectedliving/embedded-sim/.

[25] L. Militano, G. Araniti, M. Condoluci, I. Farris, and A. Iera, "Device-to-device communications for 5G internet of things," *EAI Endorsed Trans. Internet Things*, vol. 1, no. 1, pp. 1–15, 2015.
[26] S. Sakurai, G. Hasegawa, N. Wakamiya, and T. Iwai, "Performance evaluation of a tunnel sharing method for accommodating M2M communication to mobile cellular networks," in *Proc. of IEEE GLOBECOM Workshops*, 2013, pp. 157–162.
[27] V.-G. Nguyen, A. Brunstrom, K.-J. Grinnemo, and J. Taheri, "SDN/NFV-based mobile packet core network architectures: A survey," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1567–1602, 2017.
[28] V. Nagendra, A. Bhattacharya, A. Gandhi, and S. R. Das, "MMLite: A Scalable and Resource Efficient Control Plane for Next Generation Cellular Packet Core," in *Proc. of the ACM Symposium on SDN Research*, 2019, pp. 69–83.
[29] A. Mohammadkhan and K. K. Ramakrishnan, "Re-Architecting the Packet Core and Control Plane for Future Cellular Networks," in *Proc. of 27th IEEE International Conference on Network Protocols, ICNP*, 2019, pp. 1–4.
[30] A. Mohammadkhan, K. K. Ramakrishnan, A. S. Rajan, and C. Maciocco, "CleanG: A Clean-Slate EPC Architecture and ControlPlane Protocol for Next Generation Cellular Networks," in *Proc. of the ACM Workshop on Cloud-Assisted Networking, CAN@CoNEXT, USA*, 2016, pp. 31–36.
[31] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, "Softcell: Scalable and flexible cellular core network architecture," in *Proc. of the 9th ACM conference on Emerging networking experiments and technologies*, 2013, pp. 163–174.
[32] K. Samdanis, A. Kunz, M. I. Hossain, and T. Taleb, "Virtual bearer management for efficient MTC radio and backhaul sharing in LTE networks," in *Proc. of IEEE Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*, 2013, pp. 2780–2785.
[33] Y. Jung, D. Kim, and S. An, "Scalable group-based machine-to-machine communications in LTE-advanced networks," *Wireless Networks*, vol. 25, no. 1, pp. 63–74, 2019.

Amirahmad Chapnevis (Member, IEEE) received a B.S. degree in University of Isfahan (Iran) in 2015 and an M.S. degree in Amirkabir University of Technology (Iran) in 2019. He is now pursuing a Ph.D. degree in the Computer Science Department of Virginia Commonwealth University under the supervision of Dr. Eyuphan Bulut. His current research interests include efficient communication modeling for massive Internet of Things (IoT), and path planning in UAV networks.

Ismail Guvenc (Fellow, IEEE) is currently a Professor with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC, USA. He has authored or coauthored more than 300 conference/journal papers and book chapters, several standardization contributions, four books, and more than 30 U.S. patents. His recent research interests include 5G or 6G wireless networks, UAV communications, millimeter or terahertz communications, and heterogeneous networks. He is the PI and the Director of the NSF AERPAW project and the Site Director of the NSF BWAC I/UCRC center. He is a Senior Member of the National Academy of Inventors. He was the recipient of several awards, including the NC State Faculty Scholar Award in 2021, the R. Ray Bennett Faculty Fellow Award in 2019, the FIU COE Faculty Research Award in 2016, the NSF CAREER Award in 2015, the Ralph E. Powe Junior Faculty Award in 2014, and the USF Outstanding Dissertation Award in 2006.

Eyuphan Bulut (Senior Member, IEEE) received the Ph.D. degree in the Computer Science department of Rensselaer Polytechnic Institute (RPI), Troy, NY, in 2011. He then worked as a senior engineer in Mobile Internet Technology Group (MITG) group of Cisco Systems in Richardson, TX for 4.5 years. He is now an Associate Professor with the Department of Computer Science, Virginia Commonwealth University (VCU), Richmond, VA. His research interests include mobile and wireless computing, network security and privacy, mobile social networks and crowd-sensing. Dr. Bulut is an Associate Editor in IEEE Access and has been serving in the organizing committee of several conferences. He is also a member of ACM.