Dynamically Shared Wide-Area Cellular Communication for Hyper-dense IoT Devices

Eyuphan Bulut* and İsmail Güvenç†
*Dept. of Comp. Science, Virginia Commonwealth University, Richmond, VA 23284
†Dept. of Elec. and Comp. Engineering, North Carolina State University, Raleigh, NC 27607
Email: ebulut@vcu.edu, iguvenc@ncsu.edu

***Abstract*—The emerging Internet of things (IoT) technology will enable a whole new set of applications, imposing far reaching influences on multifarious aspects of the society. At the same time, it also poses grand challenges to the wireless industry, such as the difficulty of allocating sufficient control/data channel resources to a large number of IoT devices operating in the same spectrum. While there has been some work proposing solutions for efficient radio spectrum efficiency, the core network aspect of the hyper-dense IoT communication is not well studied. In particular, the core network is not optimized for IoT traffic characteristics as it is mainly designed for human users' mobile traffic characteristics. In this paper, to address these issues, we develop a dynamically shared connectivity model to manage hyper-dense IoT traffic using the same resources (i.e., bearers). Then, using an evolutionary clustering method, we group the IoT devices based on their traffic patterns. Simulation results show that the proposed solution can result in smaller number of bearer usage for a given IoT traffic compared to previous solutions, while increasing the resource utilization and decreasing signaling cost.**

***Index Terms*—5G, cellular network, clustering, core network, Internet of Things (IoT), machine type communications (MTC).**

I. Introduction

The Internet of Things (IoT) technology is expected to connect billions of intelligent objects in the near future. It will enable applications ranging from smart cities to self driving cars, from industry automation to wearable devices [1]–[3]. According to the 2017 Ericsson Mobility Report [4], there will be over 20 billion IoT cellular connections in the world by 2023, forming a massive, hyper-dense IoT network. In order to support such networks, standard organizations such as 3GPP and IEEE have been recently working on developing the next generation (5G) IoT standards [5]–[8].

The machine type devices (MTDs) in the IoT network may be directly connected to a macrocell base station (BS). Alternatively, they may communicate with a local IoT gateway (e.g., capillary network concept [9]), which then connects to the macrocell BS through a dedicated link. Finally, the MTDs can form a Device-to-Device (D2D) network among themselves, with one of the D2D nodes providing backhaul connectivity to the macro BS.

Different forms of IoT have been recently supported by the 3GPP standard specifications up to Release-14, such as eMTC, NB-IoT, and EC-GSM-IoT [10]. There also exist other standardized radio access technologies for IoT operation (including 802.11 WiFi variants, Bluetooth Low Energy (BLE), IEEE 802.15.4, ZigBee), as well as various proprietary ones (e.g., LoRaWAN, Neul, Sigfox). While the ranges of these alternatives extend from personal area networks (PANs) to wide-area networks (WANs), none of these have been truly designed for massive IoT deployments.

With the deployment of MTDs, a wide area coverage with low power and low cost communication is usually targeted. Depending on the application that the MTDs are used for, they usually need to transmit small amount of data with low frequency. Thus, when they do not need to send data to the server, to reduce the power consumption and prevent resource underutilization in core network [11], they enter to the power saving mode (PSM) [12]. Furthermore, to reduce the cost of communication when they need to send data, the radio frequency (RF) hardware is simplified with a single antenna and half-duplex communication to achieve the required bandwidth and data rate for IoT services. However, depending on the data size uploaded to the server, in some cases, the signaling cost that includes the Radio Resource Control (RRC) connection and corresponding radio bearer setups with core network can be more than the actual transmitted data size [13]. This not only increases the signaling cost but also wastes the core network resources (i.e., number of bearers) with a low utilization.

There are several works that address these issues and provide solutions for scalable and low-cost functioning and connection of hyper-dense MTDs. In [13] geographically close MTDs with similar QoS characteristics are grouped together and a shared cellular communication is achieved via defining a virtual bearer. There are also solutions aiming to connect several nearby MTDs through D2D communication, and let them connect to the cellular network using one of those MTDs [9]. The main limitation of such studies is that the MTDs need to be close to each other. However, in reality the MTDs that are part of an IoT service could be deployed to a wide area and they can not be close to each other to achieve D2D communication.

In order to address this drawback, Ito et. al [14] propose an aggregated cellular communication line for multiple MTDs through sharing of International Mobile Subscriber Identity (IMSI). The MTDs, regardless of where they are located within the service area of the same core network, which could be as wide as a state's territory, share their communication with the

978-1-5386-5097-4/18/$31.00 ©2018 IEEE

server over a time-division cellular communication line. Each time an MTD needs to send data to the server, it registers to the network to obtain the necessary bearer and releases it when it is done. From core network's perspective this is considered as a single device's alternating communication and movement. While the proposed IMSI sharing based connection aggregation solution provides a wide area solution for the aforementioned resource utilization problems, it has the following drawbacks: i) it requires pre-determined list of MTDs that will share the same communication line, ii) it needs MTDs having same cycle of upload and manages the data uploading of different MTDs at different times. However, in real-world deployments of hyper-dense MTDs, these may not be possible as many MTDs may show varying traffic characteristics due the application requirements, as well as the number of MTDs being served can change dynamically.

In this paper, we address these issues and provide an efficient and adaptive core network connectivity for hyper-dense IoT cellular communication. To this end, we propose a dynamically shared cellular communication for multiple MTDs. Not only the MTDs that have the similar data upload cycles but also the other MTDs that can achieve a communication with minimal overlap with other MTD devices' traffic share the same communication line to the core network. We study a dynamic attach procedure for such MTDs after which they upload their data without wasting the connection resources. Moreover, we propose an evolutionary clustering algorithm for MTDs in order to dynamically determine the MTDs that will share the same communication. With simulation results, we show that fewer bearers can be used to support the same MTD traffic compared to previous solution [14] with a much faster clustering algorithm than brute-force, making it suitable for online clustering in dynamic environments.

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 discuss the details of the proposed dynamically shared cellular communication system for IoT devices. Section IV discusses the clustering of IoT devices using a genetic algorithm based solution. In Section V, we present our evaluation of the proposed solution. Finally, we end up with concluding remarks in Section VI.

II. Literature Review

A. Background

As a representative of mobile core network, in Fig. 1 we illustrate a simple view of Evolved Packet Core (EPC)[1]. When an MTD with cellular interface is turned on, it needs to *attach* to the core network. The MTD first communicates with a base station (i.e., eNodeB) through a Radio Access Network (RAN) wirelessly and sends a connection request to the core network to be authenticated and allocate all necessary resources. PGW is the main entity that connects the mobile carrier network to the public Internet. It is the termination point for all external connections. The IP address of the device is allocated and maintained by the PGW as long as the device is attached. MME is the main gateway that tracks the locations of mobile devices. It is basically a user database that manages all the state information for every UE in the network. When a UE's location within the network changes (from one tower to another), the location update is sent to the MME.

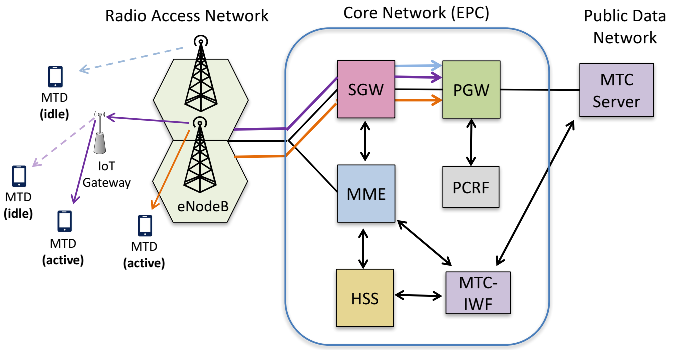

Fig. 1: IoT cellular core network architecture.

A device attached to the network stays in active state to transmit or receive data. When there is no data exchanged for a certain period of time (e.g., 5 sec), the device switches to idle state to conserve computing and memory resources in air interface, the eNodeB (i.e., cell tower) and the device. This also results in release of the channel and deletion of connections between SGW and MME. However, PGW and MME still keep the information about the device connection and consume corresponding memory resources at these gateways (only way sessions on these gateways will reduce is by turning off these devices). While this facilitates the reconnection of UEs to the public network when they get back to active state, it causes a wastage in resources on the main gateways which can only support limited number of device connections. For IoT communication, an MTC interworking function (MTC-IWF) is added [12] to serve as an intermediary function between the EPC and MTC server that the IoT services run on. MTC server does not deal with IP addresses and cellular IDs (e.g., IMSI), which is done by PGW, and just uses external identifiers (EID) to communicate with the IoT devices. The mapping of IMSI and application port ID to EID is made through communication of MTC-IWF with HSS.

B. Related work

There are several works that aim to decrease the communication tunnels in EPC. While some propose EPC side modification, some propose device side based solutions. The EPC-side solutions include enhanced standardized functions [15], [16], separated C-plane and U-plane based implementations [17], [18], and an SDN based new architecture [19]. However, each of these methods have limitations for practical applications [14], [20]. In device side, for example, in [13], virtual bearers are proposed to combine the data traffic from

[1] As the standards for 5G core network architecture has been currently developing, we provide background core network information based on well established EPC architecture. However, the ideas proposed in this paper could be easily extended to the 5G architecture.

multiple IoT devices through D2D communication and send all aggregated data over one single device (e.g., aggregator). However, this will not help when the IoT devices are spread over a wide area in which they are not in proximity of each other to form D2D communications.

A more practical solution is presented in [20], in which an IMSI sharing based solution is proposed. However, it is assumed that a single IMSI is shared among the IoT devices with same communication patterns, which may be unrealistic considering the variety of IoT services and vendors using the same mobile network operator's infrastructure. In this paper, we address this issue and propose a dynamically aggregated cellular connection for multiple MTDs with heterogeneous communication patterns.

III. Dynamically Shared Cellular Connection

The hyper-dense IoT deployment will challenge the current core network architecture. Each device has to manage a connection with the core network to communicate with MTC server even when they have low data rates and long data sending intervals. In order to resolve underutilized and wasted connectivity resources (i.e., bearer), we propose an aggregated communication model in which multiple IoT devices use the same bearer for sending data in turns. To achieve that these devices first need to share the same subscriber ID (e.g., IMSI) so that EPC will treat them the same and their communication with core network could be controlled. Thus, both on the device side and EPC side, several enhancements have to be made.

Sharing of the same subscriber ID could be achieved at the initial provisioning of these devices with the classic concept of physical cellular SIM. However, this will only make grouping of MTDs with certain others and limit dynamic regroupings (as a result of updated traffic characteristics, or node leaves/joins) will not be possible. New generation subscriber ID solutions such as virtual SIMs [21] and e-SIM cards [22], [23], which are mainly proposed to facilitate service provider (e.g., operator) switch for subscribers without changing the SIM card and overcome the potential serious storage limitation for large-scale tiny IoT devices, could be leveraged for "over the air" provisioning of network connectivity [24]. This can provide dynamic subscriber ID to MTDs that will share the bearer.

Once the MTDs that will share the same connectivity (and subscriber ID) are found, their attach process to EPC should follow a new procedure. Fig. 2 shows the overview of this procedure. The main goal is to control the communication timings of the MTDs that share the same subscriber id (e.g., IMSI) within a data upload cycle. The data upload cycle is divided into time slots during which an MTD is allowed to attach to EPC and upload its traffic to the MTC server. Between time slots there is a guard time introduced similar to [14], to block the overlapping of device communications due to latency. When there is a single MTD attached, it can use every slot for its communication (if needed). When a new MTD sharing the same subscriber id turns on, its attach request to the EPC will be rejected if it corresponds to the guard time or to the time slot that is utilized by another MTD. In latter case, EPC will notify the current MTD active in that time slot for new cycle notification that will include reserved time slots for the new MTD. The rejected MTD will retry to attach after a backoff timer expires and if it corresponds to the slots that its communication is expected, EPC will accept the attach. Similarly, when a third MTD turns on, EPC will communicate with current available MTDs on the same bearer for data upload cycle extension and will allow acceptance once the correct time slot is found in the retried attach attempts. There will be a maximum cycle duration that could satisfy the requirements of device communications with the MTC server. If all time slots within that maximum cycle is allocated by different MTDs, no more attach will be accepted (indeed this will happen rarely as the number of MTDs sharing the same session will be determined accordingly). Note that this scheme is inspired by packet reservation multiple access [25] and the work in [20]. However, the proposed approach does not waste any time slots as adaptive upload cycles are utilized.

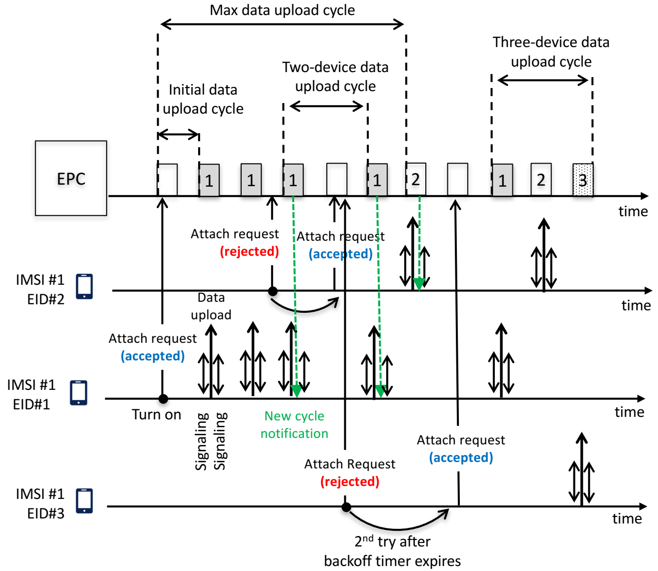

Fig. 2: Attach procedure and data upload cycle management for multiple IoT devices sharing the same cellular connection.

In the proposed method, one issue that needs to be handled is the paging of the devices that will share the same connection. Paging messages or MTC server initiated requests in general need to be delivered to the correct devices and the timing should synchronize with the non-Power Saving Mode (PSM) periods, which is often used by IoT for deep sleeping when their idle times are long. PSM is achieved with two timers (T3324, T3412) that are obtained in attach process [12]. If the devices stay idle for the duration of first timer, it enters PSM for the duration of second timer. After PSM is over, the device executes a tracking area update (TAU) to update its location. PSM can also be canceled by conducting TAU or sending a service request. Since in PSM, IoT devices do not respond to calls from the network (to save battery), the server initiated requests have to be buffered until the devices are

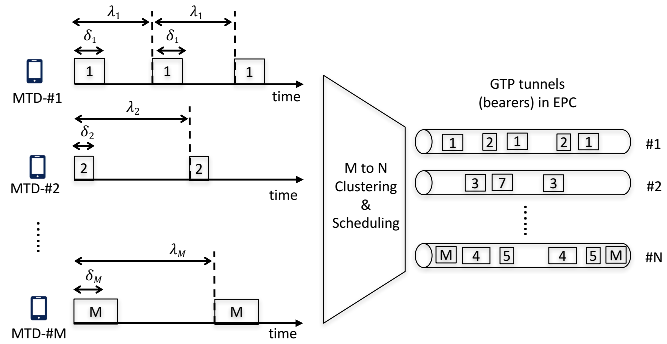

Fig. 3: Clustering of M devices into N tunnels (bearers) and scheduling of their traffic.

back from PSM. Moreover, HSS, which stores the subscriber database, has to coordinate with the MTC server to manage the turns of devices sharing the same connection and paging of right devices through a mapping. Thus, their functionalities have to be extended accordingly [14].

IV. Evolutionary Clustering of MTDs into Shared Cellular Connections

The previous section discusses the communication timings of multiple IoT devices sharing the same connection. In this section, we study the efficient clustering of all operating IoT devices based on their traffic patterns to find out the groups of devices that will share the same subscriber id and use the same connection.

The problem here is different than job scheduling problem [26], in which multiple jobs with different processing requirements and arrival rates have to be assigned to one of the available machines and run continuously until its completion. The goal here is to minimize the number of connections or bearers (analogy to machines) while assigning the packets of same users to the same bearer always. Moreover, there could be some flexibility in the timings of packet arrival rates (e.g., they could be shifted). Similarly, it is different than the bin packing problem [27], in which the goal is to pack different weighted items into the smallest number of bins possible because a scheduling has to be provided too. As a result, while a clustering of MTDs has to made based on their traffic pattern requirements, an efficient scheduling has to be managed to prevent overlaps in communication with minimal divergence from original communication needs.

Fig. 3 sketches the overview of the problem with a set of M devices that will need to be mapped to N connections (i.e., bearers in GTP tunnels), $M > N$. Traffic pattern requirements of each device i is simply modeled with δ_i duration of data upload at every λ_i time (could be extended to more complicated models).

The objective of clustering is to minimize the number of bearers while keeping the upload patterns of each MTD as stable as possible. That is, as the MTDs using the same bearer may have overlapping data upload schedules, some of them need to delay their uploading. However, such a delay should be kept within a reasonable timeframe to meet the application requirements. To this end, we define a latency threshold, $\tau_{\max}$, and consider a grouping feasible if each MTD in that group can successfully upload their data not more than a $\tau_{\max}$ delay than their expected time frame. More formally, the objective function can be defined as:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} u_j \\ s.t. \quad & u_j = \min\{1, \sum_{\forall i} b_{ij}\}, \forall j \in [1, M] \\ & \sum_{\forall j} b_{ij} = 1, \forall i \in [1, M] \\ & \mathcal{L}_i \le \tau_{\max}, \forall i \in [1, M] \end{aligned}$$

where,

$$b_{ij} = \begin{cases} 1, & \text{if MTD } \#i \text{ uses bearer } j, \\ 0, & \text{otherwise.} \end{cases}$$

$\mathcal{L}_i$ is the latency MTD #i's traffic in current bearer.

Here, in order to calculate $\mathcal{L}_i$, the traffic pattern of all MTDs in the same bearer should be considered. Moreover, when there exist more than two MTDs using the same bearer, finding the minimum possible latency for each MTD could be challenging as different combinations should be considered.

In this paper, we initially consider the case where $\tau_{\max}$ is set to 0. That is, the data communication for each MTD should not be delayed at all in any grouping. The problem of grouping the MTDs in such simplified case is still hard to find. Note that the number of groupings of a set of m items into n different groups is defined by Stirling numbers of the second kind and can be calculated as:

$$\left\{{m \atop n}\right\} = \frac{1}{n!} \sum_{j=0}^{n} (-1)^{n-j} \binom{n}{j} j^m$$

However, as we look for minimization of such groupings as long as the delay constraint is met, all possible cases with different group counts between 1 and m should be considered. Thus, we need to check $\sum_{n=1}^{n=m} \left\{{m \atop n}\right\}$ cases (which is defined as Bell numbers, B_m). For example, for M=10, there are 115,975 cases. Integer Linear Programming (ILP) based solutions could be used but in hyper-dense IoT environments the running time will be very long. Thus, we propose to use evolutionary genetic algorithms, which can provide near-optimal grouping results with comparably fast running times [28], [29].

Algorithm 1 shows the details of the clustering of MTDs with evolutionary genetic algorithm. Each chromosome consists of N numbers indicating the group of each device from $1 \dots N$. For example, a sample chromosome $\langle$1, 4, 2, 1, 2, 5, 4, 2, 3, 5$\rangle$ in Fig.4 indicates that devices (indexes starting from 1) 1 and 4 will use bearer 1, devices 2 and 7 will use bearer 4, devices 3, 5 and 8 will use bearer 2, devices 6, and 10 will use bearer 5 and finally the device 9 will use bearer 3. The crossover function is achieved through standard single point crossovers at a random location. For mutation

Algorithm 1: Genetic Algorithm for Clustering of MTDs

```
Input: $\alpha$: Size of population $\mathcal{P}$
       $\beta$: Elitism rate
       $\gamma$: Mutation rate
       $\kappa$: Number of iterations
Output: Solution X
/* Initialization                                    */
Generate $\alpha$ feasible solutions with random group
  assignments
Add them to the population set $\mathcal{P}$
/* Loop until terminal condition                     */
for $i$=1 to $\kappa$ do
    /* Elitist selection                             */
    Number of elitism $\mathcal{E} = \alpha.\beta$
    Find out the best $\mathcal{E}$ solutions in $\mathcal{P}$ and generate a
      new set $\mathcal{P}_{best}$ from them
    /* Crossover                                     */
    Number of crossover $\mathcal{C} = (\alpha - \mathcal{E})/2$
    for $j$=1 to $\mathcal{C}$ do
        Select two solutions $X_i$ and $X_j$ from $\mathcal{P}$
        Generate one point crossover solutions $X_m$ and
          $X_n$ from them
        Add $X_m$ and $X_n$ to $\mathcal{P}_{new}$
    end
    /* Mutation                                      */
    for $j$=1 to $\mathcal{C}$ do
        Select a random solution $X_r$ from $\mathcal{P}_{new}$
        Mutate each gene with probability $\gamma$ by assigning
          a random group number and generate $X_r^m$
        Replace $X_r$ in $\mathcal{P}_{new}$ with $X_r^m$
    end
    /* Set new generation                            */
    $\mathcal{P} = \mathcal{P}_{best} \cup \mathcal{P}_{new}$
end
/* Return the best solution                          */
return the solution $X$ in $\mathcal{P}$ with maximum fitness
```

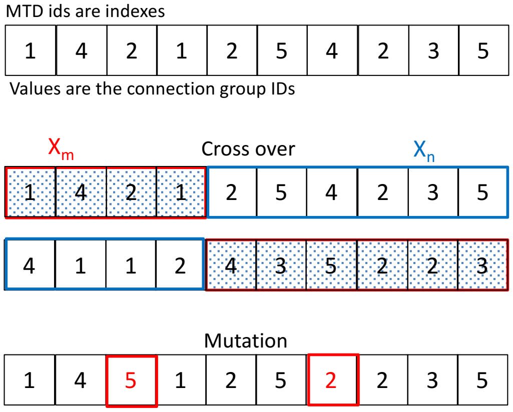

Fig. 4: Chromosome and evolutionary operations used for genetic algorithm.

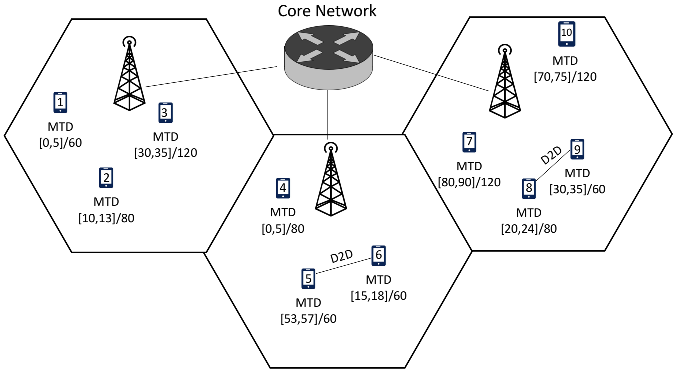

Fig. 5: The simulation scenario with 10 MTDs spread over three base station service area. Each MTD traffic ([x,y]/z) is represented by data upload between x-y minutes in every z minutes.

operation, current group of each device is assigned a new one with mutation probability, γ.

We define the fitness function of each chromosome (i.e., grouping) as the inverse of the number of groups as we aim to find the minimum possible bearers without any overlap among the traffic patterns of all MTDs in the same bearer.

V. Numerical Example

In this section, we evaluate the performance of the proposed scheme with a numerical example. We generated a simulation scenario shown in Fig. 5 with 10 MTDs in the service area of three different base stations. We have generated periodic data upload patterns for each MTD with the format of $[x-y]/z$, where z represents the period and x and y represent the start and end times of uploading in each period, respectively.

We assume that each MTD is equipped with an e-SIM [22] and initially the IMSI number on the card is unique. When an MTD wants to attach to the network for the first time, it connects over a bearer dedicated to itself. This lets it initially communicate with the core network and inform about its data sending duration and frequency intervals. Once its expected traffic pattern is analyzed and considered that it would fit to another bearer used by other MTDs, a reprovisioning of the MTD's connectivity is triggered. During this period, the IMSI of the MTD is updated as the IMSI of the other MTDs on the bearer it is assigned to. The MTD then uses the attach procedure described in Fig. 2 to connect to the new bearer. Note that the assignment of the bearer is achieved after running the evolutionary clustering algorithm defined in Algorithm 1. The genetic algorithm parameters and values used in simulations are shown in Table I.

Under the given scenario, if each MTD attaches to the core network individually, there will be 10 different bearers used from core network resources. With a capillary network concept [9] that achieves a single connection for a group of nearby MTDs while keeping them connected with Device-to-Device (D2D) communications, one may expect 8 different bearer utilization as there are only 2 pairs of MTDs that are in D2D range of each other. On the other hand, with the previous work [14] that proposes the combinations of MTDs with same upload cycles without overlaps, 3 bearers will be required to let them upload their traffic. With the proposed clustering algorithm in this paper, we can further decrease this to 2 bearers in which MTDs $\{1,2,3,6,7\}$ and $\{4,5,8,9,10\}$ share the same bearer without an overlap in their upload schedules. These results are obtained with the genetic algorithm proposed in a much faster time and verified with the brute force approach.

Parameter	Value
Population ($\mathcal{P}$) size (α)	15
Elitism rate (β)	0.2
Mutation rate (γ)	0.1
Number of iterations (κ)	100k

TABLE I: Genetic algorithm parameters.

VI. Conclusion

In this paper, we study resource-efficient connectivity of IoT devices or MTDs. To this end, we propose a dynamically shared cellular line for a group of MTDs without causing overlap in their regular traffic. We first cluster the MTDs based on their traffic patterns using an evolutionary genetic algorithm. Then, we let them share the same cellular connection in a time divisioned manner. The MTDs on the same connection are assigned the same IMSI using a dynamic provisioning with e-SIM technology. We evaluate the potential saving of the proposed system on a simple scenario initially and provide some numerical results showing its benefit. In our future work, we will extend the simulation results and evaluate the proposed schemes in real environments under different massive IoT scenarios. We will also look at the problem with non-zero delay thresholds which could help reduce the number of bearers further. Moreover, we will consider mobile IoT devices carried by people and develop new clustering algorithms with the integration of correlation [30] analysis in their mobility.

References

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (IoT): A vision, architectural elements, and future directions,” *Future Generation Computer Systems*, vol. 29, no. 7, pp. 1645–1660, 2013.

[2] S. Li, L. Da Xu, and S. Zhao, “The internet of things: a survey,” *Information Systems Frontiers*, vol. 17, no. 2, pp. 243–259, 2015.

[3] F. Yucel and E. Bulut, “Clustered crowd gps for privacy valuing active localization,” *IEEE Access*, vol. 6, pp. 23 213–23 221, 2018.

[4] Ericsson, “Mobility report,” White Paper, November 2017. [Online]. Available: https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-november-2017.pdf

[5] J. Gozalvez, “New 3GPP Standard for IoT [Mobile Radio],” *IEEE Vehicular Technology Magazine*, vol. 11, no. 1, pp. 14–20, 2016.

[6] S. Andreev, O. Galinina, A. Pyattaev, M. Gerasimenko, T. Tirronen, J. Torsner, J. Sachs, M. Dohler, and Y. Koucheryavy, “Understanding the IoT connectivity landscape: a contemporary M2M radio technology roadmap,” *IEEE Commung. Mag.*, vol. 53, no. 9, pp. 32–40, 2015.

[7] S. Yang, X. Wen, W. Zheng, and Z. Lu, “Convergence architecture of Internet of Things and 3GPP LTE-A network based on IMS,” in *Proc. Global Mobile Congress (GMC)*, 2011, pp. 1–7.

[8] B. Barret, “Next-Gen Wi-Fi Will Actually Connect the Internet of Things,” WIRED Magazine, Jan. 2016. [Online]. Available: https://www.wired.com/2016/01/wifi-halow-internet-of-things/

[9] R. M. Huq, K. P. Moreno, H. Zhu, J. Zhang, O. Ohlsson, and M. I. Hossain, “On the benefits of clustered capillary networks for congestion control in machine type communications over LTE,” in *Proc. Int. Conf. Comp. Commun. Netw. (ICCCN)*, Aug. 2015, pp. 1–7.

[10] 3GPP, “Standards for IoT,” Dec. 2016. [Online]. Available: http://www.3gpp.org/news-events/3gpp-news/1805-iot_r14

[11] E. Bulut and A. Gosain, “Mobile core network redimensioning for efficient resource utilization,” in *Proc. IEEE Int. Conf. Computer Commun. (INFOCOM) – Perfecto Workshop*, 2017.

[12] 3GPP, “Architecture enhancemens to facilitate communication with packet data networks and applications,” v15, 2017. [Online]. Available: TS23.682

[13] 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. IEEE Int. Symp. Personal Indoor Mobile Radio Commun. (PIMRC)*, 2013, pp. 2780–2785.

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

[15] 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. IEEE GLOBECOM Workshops*. IEEE, 2013, pp. 157–162.

[16] Y. Kitatsuji and H. Yokota, “On-demand session establishment for all IP-based mobile networks,” in *Proc. EURO-NF Conf. Next Generation Internet (NGI)*, 2010, pp. 1–8.

[17] J. Kaippallimalil and H. A. Chan, “Network virtualization and direct ethernet transport for packet data network connections in 5g wireless,” in *Proc. IEEE Global Commun. Conf.*, 2014, pp. 1836–1841.

[18] R. Wakikawa and S. Matsushima, “Stateless user-plane architecture for virtualized epc (vepc),” 2015.

[19] X. Jin, L. E. Li, L. Vanbever, and J. Rexford, “Softcell: Scalable and flexible cellular core network architecture,” in *Proc. ACM Conf. Emerging Netw. Experiments Technol.*, 2013, pp. 163–174.

[20] M. Ito, N. Nishinaga, Y. Kitatsuji, and M. Murata, “Aggregating cellular communication lines for IoT devices by sharing imsi,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–7.

[21] TechNews, “Xiaomis 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/.

[22] 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

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

[24] L. Militano, G. Araniti, M. Condoluci, I. Farris, and A. Iera, “Device-to-device communications for 5G Internet of things,” *IOT, EAI*, 2015.

[25] D. J. Goodman, R. A. Valenzuela, K. Gayliard, and B. Ramamurthi, “Packet reservation multiple access for local wireless communications,” *IEEE Trans. Commun.*, vol. 37, no. 8, pp. 885–890, 1989.

[26] J. K. Lenstra, A. R. Kan, and P. Brucker, “Complexity of machine scheduling problems,” *Annals of discrete mathematics*, vol. 1, pp. 343–362, 1977.

[27] M. De Cauwer, D. Mehta, and B. O’Sullivan, “The temporal bin packing problem: An application to workload management in data centres,” in *Proc. IEEE Int. Conf. Tools with Artif. Intel. (ICTAI)*, 2016, pp. 157–164.

[28] E. Falkenauer, *Genetic algorithms and grouping problems*. John Wiley & Sons, Inc., 1998.

[29] A. H. Kashan, M. Jenabi, and M. H. Kashan, “A new solution approach for grouping problems based on evolution strategies,” in *Proc. IEEE Int. Conf. Soft Computing and Pattern Recognition*, 2009, pp. 88–93.

[30] E. Bulut, S. C. Geyik, and B. K. Szymanski, “Utilizing correlated node mobility for efficient dtn routing,” *Pervasive and Mobile Computing*, vol. 13, pp. 150–163, 2014.