Share-Not-To-Waste: Scalable IoT Networking through Subscriber Identity Sharing

Amirahmad Chapnevis*, and Eyuphan Bulut*
*Dept. of Comp. Science, Virginia Commonwealth University, Richmond, VA 23284
Email: {chapnevisa, ebulut}@vcu.edu

***Abstract*—In this demo, we design and implement traffic aggregation of multiple mobile devices using the same subscriber identity and resources in the mobile core network. Using the off-the-shelf smartphones, programmable SIM cards, Amarisoft Callbox with both core network and eNB components, we demonstrate the setup and implementation of an experiment where each of these devices shares a clone of the same SIM card and connects to the core network at different times for download tasks. The results show that core network resources can be efficiently used through this sharing compared to the communication through separate cellular lines for each device, yielding a promising solution to massive IoT communication.**

***Index Terms*—Massive Internet of Things, mobile core network, resource optimization**

I. Introduction

Subscriber identity (i.e., International Mobile Subscriber Identity (IMSI)) sharing based connection and communication [1]–[3] aims to use the mobile core network resources efficiently for the Internet of Things (IoT) devices that have low data rates and infrequent communication needs (e.g., twice a day of moisture data upload from a sensor deployed in an agricultural field). This is achieved by providing a group of IoT devices with a common Subscriber Identity Module (SIM) profile that uses the same IMSI number and lets them connect to the core network in turns and perform their data communication. Each time the turn comes to a device, it first registers/attaches to the core and after setting up the call and completing the download/upload tasks, it deregisters/detaches from the core to release the resources at the core (like turning off the phone). Thus, the core network considers each of these connections as if they are coming from the same device. Note that this is different from the UEs going idle when they are not actively sending/receiving data, because such UEs still stay registered in the core and use the memory resources. Moreover, this solution provides a more scalable traffic aggregation compared to the commonly used traffic aggregation method that connects multiple IoT devices over a local IoT gateway because it provides aggregation for any device in the serving region of a core network (that can cover hundreds of base stations) while the latter only allows aggregation among the nearby devices.

This IMSI sharing based system has been studied in a few recent works [1]–[3] in detail with a focus on the changes needed in call flows and core network architecture towards its realization in existing deployments, as well as by looking at the problem of efficient grouping of the IoT devices based on their traffic patterns. However, in all these studies, only simulations have been considered during evaluation. To this end, in this demo, *for the first time,* we show a realization of this system through experiments with off-the-shelf devices and a commercial core network.

This work is supported in part by National Science Foundation (NSF) award CNS-1815603, and by Commonwealth Cyber Initiative (CCI).

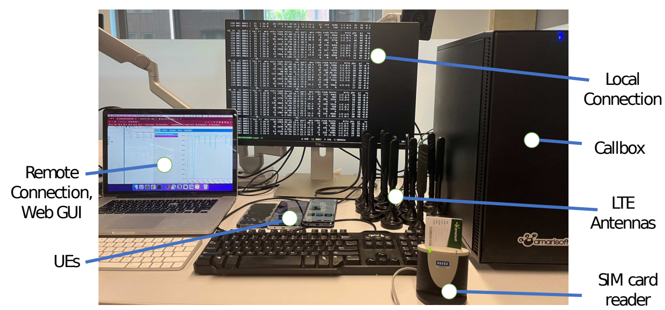
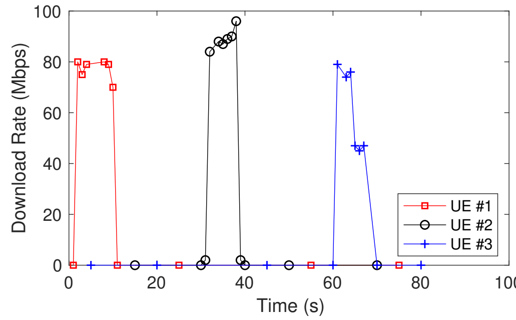


Fig. 1: Our lab setup with Amarisoft core network, LTE antennas, and smartphones.

II. Design and Implementation

Our lab setup is shown in Fig. 1. We implemented the proposed system using Amarisoft Callbox Classic [4] which implements the 3GPP core functionality (i.e., LTE, 5G NSA) with all components and provides antennas for the UE connections. The core network settings in the Amarisoft machine allows using three different algorithms for authentication and key generation; namely, XOR, $MILENAGE$ and $TUAK$. The SIMs we had from Amarisoft had the same IMSI number and they were using XOR algorithm. Thus, we used the SIMs in their default settings when we test the scenario with all devices having the same IMSI number. However, for the scenario where each device has a different IMSI, we reconfigured the SIM cards with $MILENAGE$ algorithm as reconfiguration of SIM cards with different IMSI numbers under XOR algorithm could not make them connect to the core network (we are investigating this issue). $MILENAGE$ algorithm could also have been used to reconfigure SIMs with the same but different from the default IMSI number. We used HID OMNIKEY smart card reader and the PySim application to reconfigure the SIM cards. In the configuration of XOR algorithm (in */root/enb/config/enb.cfg* file), there is also a specific parameter called $multi\_sim$, which indeed allows



Total: 3, registered: 1, LTE: 3

IMSI	IMEISV	RAT	M-TMSI	Reg.
001010123456789	9900168402492632	LTE	0x48601d94	true
001010123456789	3555721104828425	LTE	0x813e8165	false
001010123456789	3524931155344245	LTE	0xf61c214e	false

Fig. 2: (i) Three UEs with the same IMSI connect to the core at different times to download data without overlap. (ii) Registration status from 0-10 sec at the core.

multiple devices with the same IMSI but different IMEI to be able to connect to the core network simultaneously (this is for testing environments and uses IMSI+IMEI for identification of each UE). Thus, we set it to false to make sure only one device with the same IMSI is actively connected to the core. The three smartphones (each representing an IoT device) used for the experiments are OnePlus Nord N10 (5G), Google Pixel 5 (5G), and Motorola One 5G ACE.

In order to generate an automated registration/deregistration and data communication process on these smartphones, we used an application named Click Assistant [5] and the airplane mode functionality. Click Assistant allows for modeling automatic click behaviors on the predefined locations of the screen following a schedule repetitively. Similar to [3], we used a data traffic model which makes each device perform a data upload/download of δ time units at every λ time units, starting at time s and ending at time e, within each λ duration. More specifically, we used three UEs that download data between 0-10, 30-40, and 60-70 seconds at every 100 secs (λ). All devices start with airplane mode on. When the turn, for example, for the first UE comes, the Click Assistant app clicks the airplane mode symbol to turn it off to make the UE register/attach to the network. Then, the Click Assistant opens the Google play app and starts to download a large file. After 10 sec, the Click Assistant clicks the airplane mode sign to turn it on which will stop the download and detach the UE from the network.

III. Demo Results

A. Download Rates and Core Memory Usage

In Fig. 2, we show the download rate from each UE during the experiment duration (with UEs having the same IMSI and attaching in turns). Each UE can reach around 80 Mbps and maintain it during their 10 sec duration. The attaching process happens very quickly so that the download starts without a noticeable delay. The download rate of UEs is zero when they are detached from the network. The table in the figure shows that only the first UE is registered between 0 and 10 seconds, confirming memory usage for only one UE at the core.

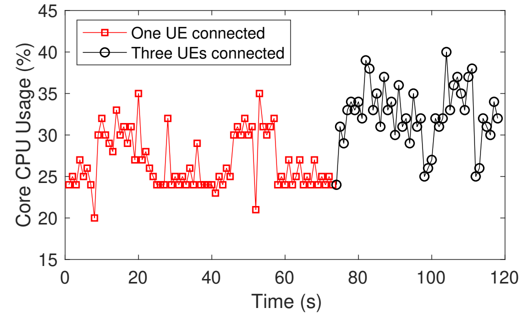


Fig. 3: CPU utilization in the core network when one UE is connected and three UEs are connected at the same time.

B. Core CPU Usage

In Fig. 3, we also show a comparison of the CPU utilization at the core when the UEs connect to the network in turns (i.e., only one UE connected at a time) and when all three UEs connect to the network simultaneously (using different IMSI or using the same IMSI with XOR algorithm). The figure shows that connecting three UEs at the same time causes an increase in the CPU utilization (due to resource management in the eNB and the core), with an average usage of 32.4%, while having one UE connected at a time results in around 26.7% usage. These are measured when UEs do the same download tasks. Note that in the case of three UEs connected, we also see around 26-29 Mbps of download speed per UE, compared to one UE case where we have around 80 Mbps.

IV. Conclusion

In this work, we demonstrated an experiment with three mobile devices each having a clone of the same SIM and connecting to the core network at different times to perform a download task. We showed automation of these in-turn connections from the devices using the off-the-shelf smartphones and an auto clicker app and collected data using Amarisoft core. The results show that this type of traffic aggregation reduces both the memory and CPU resources at the core and provides a promising solution for efficient communication of a massive number of IoT devices.

References

[1] 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.

[2] 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.

[3] A. Chapnevis and E. Bulut, "IMSI Sharing based Dynamic and Flexible Traffic Aggregation for Massive IoT Networks," *IEEE IoT Journal*, 2022.

[4] Amarisoft, "Callbox series," 2022. [Online]. Available: https://www.amarisoft.com/products/test-measurements/amari-lte-callbox/

[5] "Click Assistant - Auto Clicker," 2022. [Online]. Available: https://play.google.com/store/apps/details?id=com.rise.automatic.autoclicker.clicker