REPLISOM : Disciplined Tiny Memory Replication for Massive IoT Devices in LTE Edge Cloud

Sherif Abdelwahab*, Bechir Hamdaoui*, Mohsen Guizani†, and Taieb Znati‡
* Oregon State University, abdelwas,hamdaoui@eecs.orst.edu
† Qatar University, mguizani@ieee.org
‡ University of Pittsburgh, znati@cs.pitt.edu

***Abstract*—Augmenting the LTE evolved NodeB with cloud resources offers a low-latency, resilient, and LTE-aware environment for offloading the Internet of Things (IoT) services and applications. By means of devices memory replication, the IoT applications deployed at an LTE integrated edge cloud can scale its computing and storage requirements to support different resource-intensive service offerings. Despite this potential, the massive number of IoT devices limits the LTE edge cloud responsiveness as the LTE radio interface becomes the major bottleneck given the unscalability of its uplink access and data transfer procedures to support a large number of devices that simultaneously replicate their memory objects with the LTE edge cloud. We propose REPLISOM ; an LTE-aware edge cloud architecture and an LTE-optimized memory replication protocol which relaxes the LTE bottlenecks by a delay and radio resource efficient memory replication protocol based on the Device-to-Device communication technology and the sparse recovery in the theory of compressed sampling. REPLISOM effectively schedules the memory replication occasions to resolve contentions for the radio resources as a large number of devices simultaneously transmit their memory replicas. Our analysis and numerical evaluation suggest that this system has significant potential in reducing the delay, energy consumption, and cost for cloud offloading of IoT applications given the massive number of devices with tiny memory sizes.**

***Keywords*—Internet of things, Mobile edge computing, Memory replication, Compressed sampling, Long Term Evolution (LTE).**

I. INTRODUCTION

The LTE all-IP architecture, built-in security, and spectral efficiency nominate LTE to become the dominant connectivity technology for the Internet of Things (IoT), while IoT services and applications create unprecedented traffic growth for 3GPP LTE/LTE-A networks [1]. For IoT applications, it is becoming a global consensus that cloud computing technology is an essential driver for IoT computation speedup, energy consumption, and service realizations [2], [3]. IoT applications include, for example, connected vehicles, smart grids and cities, and wireless sensors and actuators networks [4]. In such applications, IoT devices offload its computations by replicating small-sized (tiny) memory objects and transferring these memory replicas through LTE networks to a back-end cloud computing infrastructure that enables the IoT applications to scale its computing resources on elastic infrastructure instead of resource limited devices besides many other benefits. Despite the potential of LTE to efficiently transport these memory replicas to the scalable cloud infrastructure, the massive number of devices per cell, which is projected to reach 50,000 devices by 2020 (see [5]), renders an LTE network as the major communication bottleneck for cloud offloading that can significantly limits offloading performance gains.

Memory replication is a disciplined process in which consistency must be ensured such that a write operation is followed by a memory update operation with the cloud for each device [6]. On the other hand, the current LTE network access and uplink scheduling procedures introduce a significant latency and energy inefficiency to update a large number of tiny memory replicas. An LTE cell can become easily blocked, if only 10% of the devices it covers became active simultaneously to update their memory replicas with the cloud. Unsurprisingly, this bottleneck is *not* a result of bandwidth limitation; as an IoT device memory replica is typically a few Kilobytes and the LTE network is optimized for high throughput and low latency applications [7], [8], [9]. Nevertheless, the LTE standard is not optimized to support a large simultaneous access from devices while remaining delay and energy efficient; as this requires allocating a large number of control channels and wastes radio resources, which are primarily intended for transporting conventional mobile users data. To remain a disciplined process, we design an *LTE-optimized* memory replication architecture and protocol to harvest the benefits of both LTE and cloud computing technologies.

In this paper, we propose REPLISOM [1], a memory replication architecture and protocol based on: the emerging mobile edge computing paradigm [10], the Device-to-Device (D2D) communication technology [11], and the compressed sampling theory [12]. We summarize our contribution as follows:

- Improve the cloud responsiveness for IoT services and applications by distributing cloud resources geographically close to the IoT devices. Unlike Cloudlets, MAUI, CloneCloud, and COSMOS [3], [13], [6], [14], the proposed architecture enables the design of LTE-aware cloud procedures in general and memory replication protocols optimized for tiny-sized memory replication from a massive number of IoT devices in LTE in specific.
- Reduce the memory replication delay by diminishing the need of initiating the LTE random access procedure for each replica transfer, which introduces an undesirable delay, increased energy consumption, and risk of instability given the large number of simultaneously active

This work was supported by National Science Foundation (NSF) (CNS-1162296).

[1]The name,REPLISOM , is inspired by *replisomes* that carries out replication of DNA.

devices. Unlike the application of compressed sampling in sensor network which relies on spatial and temporal correlation of sensor data [15], [16], the proposed memory replication protocol relies on two level of sparsity structures at the network and memory levels.

- Enhance the LTE signaling overhead and resource usage for memory replication by avoiding the allocation of unnecessary dedicated control channels per device, which wastes the scarce radio resources and risks the blocking of human communications. Unlike other protocols used in general purpose machine type communications in LTE [17], [18], our work relies on the disciplined nature of memory replication to design a pull based memory replication protocol which uses a significantly less number of control channels compared to direct memory replication using the conventional LTE access and data transfer procedures.

A. Solution Outline

The REPLISOM *architecture* is a mobile edge cloud architecture (see Fig. 1) in which we augment the evolved NodeB (eNB) with cloud computing resources and refer to this augmented architecture by the LTE edge cloud. The LTE edge cloud is a new *integrated* radio access network element that provides virtualizable computing, storage, and networking resources to clone device specific IoT applications and services. The architecture is a highly responsive system that neutralizes the back-hauling and routing bottlenecks which exist in current conventional cloud architecture. Deploying cloud computing resources in the proximity of the IoT devices allows *developing an LTE radio interface which is optimized for memory replication utilizing already in place technologies*. For example, an LTE capable IoT devices already incorporate D2D technologies that support efficient proximal devices discovery and direct communication. By utilizing the capabilities of the D2D technology and the existing LTE control and data channels, we show the possibility to improve the memory replication performance through an LTE-optimized protocol.

The REPLISOM *protocol* is an LTE-optimized memory replication protocol(s) that relies on pulling the memory replicas from the IoT devices instead of pushing the replicas from the devices to the LTE edge cloud. We observe two sources of sparsity in memory replication. The first source is at the network level, where the ratio of the active devices to the total number of devices is small even if the number of simultaneously active devices is large under the traffic models defined by 3GPP for machine communication. This source of sparsity is *independent* on any assumption about the devices memory contents (e.g. spatial or temporal correlation). The second source of sparsity is at the memory replica level, where the deltas of memory replicas typically exhibit few non-zero memory blocks. In REPLISOM , a device sends its updated memory replica to some other neighbor devices using D2D communication, while a receiving device compresses all the received memory replicas into a single compressed replica. The edge cloud then selects a number of devices, which is much less than the total number of devices in the LTE cell, and pulls the compressed replicas from these devices. By compressed sampling reconstruction algorithms, the cloud can recover the original replicas exactly utilizing the *sparsity at the network level*. Moreover, we show possible further improvements to the devices energy consumption and replication delay by utilizing the *sparsity at the memory replica level*.

Since the cloud pulls compressed replicas from a number of devices that is proportional to the number of updated replicas, there is no need for initiating the random access procedure. As an LTE device is already synchronized with its serving LTE cell to decode the cell's control channels, with REPLISOM a device just wakes up in predefined sub-frames to verify its pulling occasions and transmit its compressed replica while remaining in a deep sleep state if it is not pulled. Unlike directly pulling the original replicas from each device in the cell, which also does not require initiating the random access procedure, the number of control channels allocated for the proposed protocol is significantly less than the number of control channels allocated to pull replicas from each device in the cell.

The remaining of this paper is organized as follows. We first discuss the related work in Section II. Then, we present the proposed architecture in Section III where we discuss the LTE specific challenges and the architectural rule of the D2D technology. Section IV delves into the proposed memory replication protocol and its relation to the compressed sampling theory and shows how we use the two sources of sparsity, at the network level and at the memory replica level, to design an efficient pull based memory replication protocol. In Section V, we describe our performance benchmarks and provide numerical evaluations of REPLISOM in comparison to replica transfer using the conventional LTE procedures. Finally, we conclude our paper in Section VI.

II. BACKGROUND

Creating computing infrastructure back-ends for devices such as cloud platforms has been in the heart of the IoT research since its inception in 1991 [19]. The vision of cloud computing for IoT has evolved through the years to what we know today as Edge computing [10], [3] or Fog Computing [4]. These evolved platforms extend the cloud computing paradigm with new characteristics such as: location awareness, low latency networking, geographically distributed infrastructure, support of mobility, wireless access awareness, and cloud interoperability [2]. Our work focuses on the efficient design of memory replication protocol for the purpose of computation offloading in the LTE edge cloud with support of massive number of IoT devices.

a) ***Cloud offloading near the edge****:* The idea of augmenting resource constrained devices with a resource-rich cloud infrastructure accompanied the evolution of mobile computing more than a decade ago [20], [21]. Computation offloading with a fine-grain memory replication has been the focus of research since then [6], [13], [22], [14]. New forms of cloud platforms (e.g. cloudlets) emerged to provide computing resources for proximate devices with a minimal communication delay. The success of computation offloading to improve the

computational capacity and energy consumption in the Internet of things era is conditioned by the limits of the underlying networking technologies that support memory replication from massive number of devices (see results in [23], [24]). Our work investigates these limitations, architectural evolution, and protocol design to support cloud-centric IoT services and applications at LTE eNB.

*b) **Massive IoT devices in LTE**:* The energy consumption and delay performance characteristics of Internet of Things (Machine Type Communication) in LTE has been one main focus of the cellular networks research and standardization efforts [8], [25]. Particularly, the delay and energy characteristics of the LTE random access and uplink transmission procedures resemble the major bottlenecks under network overload from massive number of IoT devices [9]. The existing approaches to improve the LTE performance in such overload situation focus on finding improvements for existing uplink transmission mechanisms [17], [8]. Our work is related to these research efforts as we anticipate the impact of the LTE bottlenecks on the memory replication performance, hence cloud offloading. Our proposed protocol is *specific to the memory replication traffic*, and not to any uplink traffic type, and its validity is conditioned by the evolution of Device-to-Device Communication technologies [11], [26], [27]. The LTE-aware design of the proposed memory replication protocols reduces the dependency on the LTE random access procedure and requires significantly less number of control channels.

*c) **Compressed Sampling in networking**:* Our work is an application of the theory of sparse recovery in compressed sampling [28], which has several applications in networking. Approaches to a decentralized compression of networked data has gained a lot of attention in the last decade [15] and had applications in: network coding [29], [30], sensor measurements collection [31], [32], [16], network tomography [33], and medium access [34]. The application of compressed sensing in such applications utilizes the sparse properties of the data in different forms. In sensor networks, for example, spatial and temporal correlations of sensor measurements are the main sources of sparsity [31] which requires finding a network transformation to sparsify the network data (e.g. using graph wavelets, or diffusion wavelets) [15]. Our work relies on the sparsity of having a limited number of simultaneously active devices out of a large number of devices at the network level besides the sparsity of having a limited number of non-zero memory blocks in memory deltas at the memory level. These sources of sparsity do not require any particular transformation to sparsify the memory replicas.

III. LTE Architecture Evolution for Edge Computing and Massive IoT Devices

The current LTE architecture performs specialized processing that supports radio communication with LTE devices and traffic back-hauling. When an IoT device, in an LTE cell, offloads its computation to a cloud computing platform, it carries out LTE network access and uplink data transfer procedures including: initiating a random access, sending uplink scheduling requests, receiving uplink grants, and transmitting its uplink data (memory replica) using radio resource blocks (see Fig. 6). Then, a memory replica (received packets at the LTE cell) passes through multiple stages of packet forwarding from the LTE cell through the serving gateway, to the packet data network gateway, to the Internet routers, to the data center routers until it finally reaches a cloud computing node that hosts a virtual machine - corresponding to the IoT device - that updates the replicated memory block. The current LTE architecture is well optimized for voice and data packet communication. However, given the service requirements of IoT including the support of: massive number of devices, reduced complexity, and power efficiency, the LTE architecture is not optimized for cloud computing offloading for IoT services and applications which requires tighter delay bounds on packets transmission.

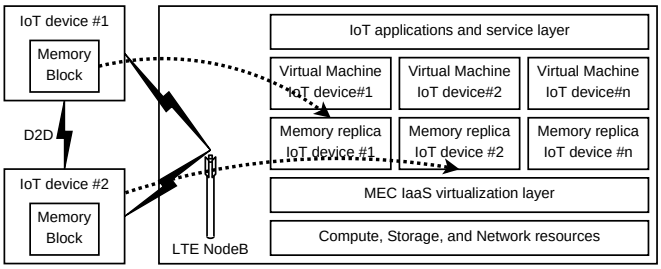

Fig. 1: Proposed Memory replication architecture in LTE/LTE-A with Mobile Edge Computing and Device to Device communication.

A. Proposed Architecture

We propose to address the LTE architectural bottlenecks by deploying local cloud computing resources within the radio access network based on the mobile cloud computing paradigm [10]. Once the device memory replica reaches the LTE eNB, it becomes available to the IoT services and applications deployed at these local cloud resources. This architectural change improves the cloud responsiveness by distributing the cloud resources geographically close to the IoT devices (Fig. 1) and paves the road to: $i)$ an improved IoT services and applications resiliency by splitting the cloud resources to local resources (to devices) and global resources (conventional cloud), $ii)$ simplified analytics and big data by capturing key information from devices with possible direct device access, $iii)$ reduced latency as applications react faster to devices and context changes away from possible congestion in other parts of the LTE network other than the radio network, and $iv)$ optimized cloud protocols that are aware of network information (e.g. radio conditions, performance statistics, and technology limitations).

1) Technical Challenges: Several technical challenges pertain to this LTE edge cloud architecture such as: the design of highly distributed applications, support of optimized applications and virtual machines portability, integration cloud security with current 3GPP-security requirements and practices, improving applications and cloud hardware resilience to match 3GPP availability and service continuity requirements, and design of LTE radio interface aware cloud protocols.

The LTE radio interface, in specific, resembles the major bottleneck for efficient memory replication in LTE edge computing. Three benchmarks capture the system wide efficiency of a memory replication protocol: the total delay, the total consumed energy, and the total number of control channels required to transfer *all the updated memory replicas from all the active devices* to the edge cloud (see Section V for a detailed description and evaluation). For example, the memory replication efficiency implies minimizing the time a single device waits between two successive replica updates. During that time the the eNB is busy transporting memory replicas from other devices (besides conventional human communication). Several characteristics and observations render an LTE radio interface as the major bottleneck and motivate the design of an LTE-optimized memory replication protocol.

a) Large simultaneous replica updates: Although the majority of IoT devices exhibit a memory change every few minutes, the massive number of devices per cell results in a large simultaneous replica updates per minute. Recent 3GPP studies on enhancements of LTE for IoT suggest new traffic models of IoT devices that can cause memory changes every 30 minutes down to 10 seconds in case of major failures which require the design of rapid network access procedures [35]. Let n denote the number of devices in an eNB, and $k = \rho n$ denote the number of devices with an updated replica at time t (active devices) where ρ is the ratio of the active devices to the total number of devices in one minute. Under the suggested 3GPP models, the parameter ρ typically ranges from 0.1 to 0.3 (i.e. 1000 to 15,000 simultaneous active devices per minute). The LTE physical layer, besides other system aspects, restricts a large number of simultaneous replica transfers due to several physical layer design aspects. For example, the finite sounding reference signal periodicity restricts the number of simultaneous active devices so that the eNB is able to estimate the uplink channel quality with a finite accuracy (see [36] for more details on physical layer scalability limitations).

b) Access latency and control channels: As the devices are not engaged in frequent packet transmission and reception, devices will typically remain in idle mode (not connected to the cell) for a long time to save their energy and reduce the cell interference. Unfortunately transitioning back from the idle mode to the connected mode results in an excessive access latency as every memory update involves an initiation of the random access procedure (see Section 5.1 in [37] for details on the random access procedure). We refer to this scenario as the *idle to active* scenario. Every device accessing the network from the idle mode transmits and receives at least four control messages (illustrated in Fig. 2) : the random access request (device initiates the procedure), random access response (cell acknowledges the request and assigns initial resources), uplink connection/scheduling request (device requests uplink resources), and uplink grant (cell allocates uplink resources for data transmission). Random access requests can also collide. Let L denote the random access opportunities per second and γ denote the random access requests per second, the probability of collisions during the random access procedure is given by the $Pr_{collision} = 1 - e^{-\gamma/L}$ [5]. Even if an operator was able to increase L such that the random access initiation is collision free ($Pr_{collision} \approx 0$), recent studies suggest that in such hypothetical case the average random access latency *per device* ranges from 47 ms to 55 ms (measured from the initiation till the first uplink transfer) in such hypothetical collision free scenario [5], [9].

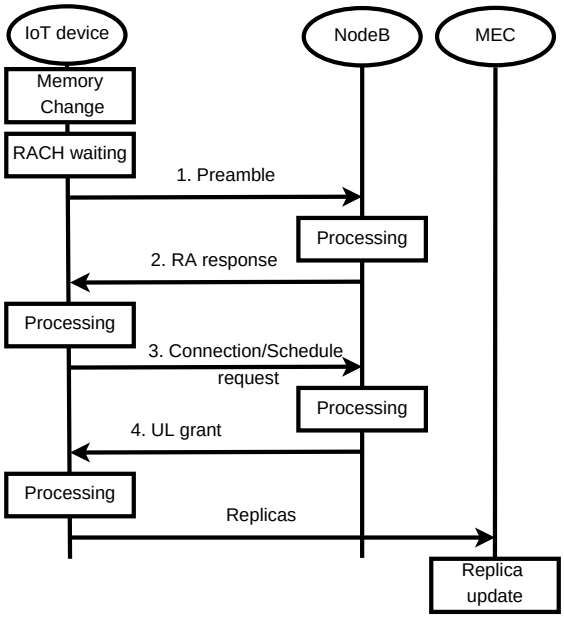

Fig. 2: The *idle to active* scenario: LTE signaling for a replica transfer from the idle mode to the active state.

c) Over-allocated scheduling opportunities: Preventing devices from transitioning to the idle mode can improve the access latency significantly. In such scenario, a device stays in a dormant state for monitoring the control channels in predefined occasions, and does not need to initiate random access except if, for example, it lost frame synchronization, or there were no uplink resources available to send scheduling requests (see [37]). Optimized discontinues reception/transmission achieves energy saving for always connected devices (in dormant state) where devices go into deep sleep and wake up only in predefined occasions to maintain, for example, frame synchronization and decode other control channels. Fig. 3 illustrates replica transfer from the dormant state where the device first identify a scheduling opportunity and sends an uplink scheduling requests. Once, the LTE cell allocates uplink radio resources for the device, the cell sends an uplink grant message to the device to start its transfer. We will refer to this scenario as the *dormant to active* scenario. This procedure exhibits the least possible latency, but ideally requires allocating scheduling opportunities for n devices, which is not necessarily feasible for a large number of devices (see [36] and [37]) and is inefficient due to the unnecessary allocation of control channels as we will detail in Section V.

d) Untraceable memory changes: A memory replication protocol can be pull based, where the edge cloud initiates the replicas transfer through paging the IoT devices, hence the random access procedure is not initiated and the scheduling

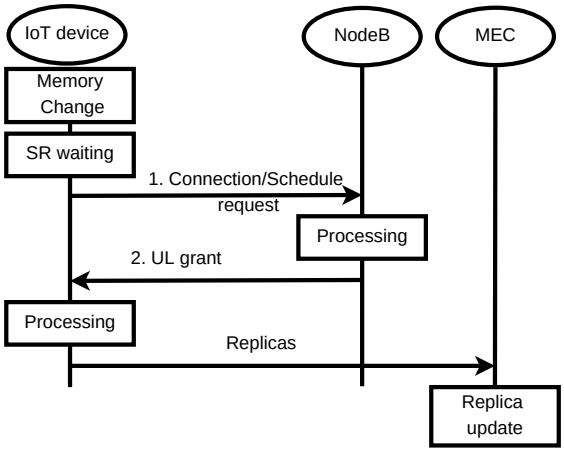

Fig. 3: The *dormant to active* scenario: LTE signaling for a replica from the dormant state to the active state.

requests are allocated only for pulled devices. With the current LTE specification, the cell can page a device to initiate an uplink transfer. There are two challenges accompanying this process. First, the paging process is not ideal and involves latency, collisions, and capacity challenges that are as difficult as the random access procedure. Second, it is not trivial for the edge cloud to determine which devices are active to pull replicas from, without initiating unnecessary paging or pulling replicas from all the devices in the worst case.

2) Architectural Rule of D2D Communication: We address the previously discussed challenges by designing a pull based memory replication protocol using D2D communication (in specification starting 3GPP Release 12) and compressed sampling (Section IV). *Architecturally*, IoT devices can communicate directly with each other using the licensed cellular spectrum (in-band), or the unlicensed spectrum (out-band). In-band D2D can use the same operating band of the LTE cell in an underlay mode or a different band in an overlay mode. D2D communication requires: interference management, resource allocation, and device discovery services that devices can perform autonomously or by the LTE infrastructure assistance (i.e. small and home cells other than the macro cell in Fig. 1) [11].

In the proposed architecture, device pairs can communicate autonomously in parallel with low power radio. Typically, the D2D transmission power, P_{d2d}, is a fraction of mW (e.g. 1 mW [38]). A device is also capable of communicating with a group of devices in multicast, and the total time required in transferring a replica from one device to the other is comparable to the idle to active scenario (Fig. 2). Let r denote the maximum communication range of the LTE D2D technology[2]. A device is connected to a $Neigh(i) = \{j : Dist(i,j) \leq r\}$ neighbor set, where $Dist(i,j)$ denote the Euclidean distance between any two devices i and j. With an LTE cell that covers an area a and serves n devices [3] , a device i is directly connected to $\mathcal{E}\{Neigh(i)\} = \eta\pi r^2$ on average, where $\eta = \frac{n}{a}$ denote the network density.

[2]Early commercial solutions show LTE D2D communication range up to 500 meters and we assume $r < 200$ meters.

IV. Memory Replication Protocols

The main intuition behind the proposed REPLISOM memory replication protocol is to recognize that during a short time interval, the memory replicas of all the n devices in an eNB resemble a sparse vector, x, of length n that has k non-zero entries which represent replicas from k active devices. We refer to this observation as *the sparsity at the network level*. Hence. it is possible to recover all the replicas (the vector x) from few memory replica samples $m < n$ by the use of compressed sampling reconstruction algorithms [12], [15]. Efficient replica recovery is possible with compressed sampling: if we designed a *low complexity* protocol that samples the replicas *incoherently* with *few control channels*; and if we treated the memory replicas as *blocks of finite precision floating numbers* instead of low level binary bit streams. These insights enable the development of the proposed pull based memory replication protocol that does *not* have to learn which devices are active with an updated memory contents, while it pulls only memory replicas from a number of devices that is *far less* than n. The protocol works as follows (see Fig. 4 for the messages flow).

A. Proposed Protocol

Suppose that k devices are active and updated their memory. Let p denote the memory page of a device i which is split into l blocks that are represented as finite precision floating numbers.

1. *An active device* i, upon updating p at time t, performs the following:
   - 1.1. creates the i-th memory replica, $x_{\mathrm{i}} \in \mathbb{R}$, as: $x_{\mathrm{i}} = \mathrm{float}(device : i,\ time : t,\ memory : p)$, where float is a function that casts the replica bits to a fixed point floating number.
   - 1.2. pushes x_{i} to randomly chosen neighbors in a *multicast* D2D communication.
2. *A receiving neighbor device* j performs the following:
   - 2.1. solicits memory replicas periodically from neighbor devices until it receives at least $d = O(log(n/k)/\epsilon$ updated replicas ($\epsilon \in (0,1)$) from $N \subset Neigh(j)$ neighbor devices,
   - 2.2. aggregates all the received replicas into one compressed replica $y_{\mathrm{j}} = \phi_{\mathrm{jj}} x_{\mathrm{j}} + \sum_{\forall i \in N} \phi_{\mathrm{ji}} x_{\mathrm{i}}$ and stores only y_{j}, where $\phi_{\mathrm{ji}} \in \mathbb{R}$ is a predefined signature that the LTE edge cloud *initially* generates and assigns to the j-th device upon declaring another device i as a direct neighbor,
3. *The LTE edge cloud* performs the following:
   - 3.1. randomly selects $m = O(k \log(n/k))$ out of n devices,
   - 3.2. sends pulling requests to the selected devices using *pre-scheduled uplink grants* such that: the eNB can pull a device j only at $(j \mod m)$

[3]3GPP suggests $a \approx 1$ square Kilometer as detailed in [35].

occasions in the LTE frame, and the uplink grants include initial radio block allocation information for uplink transfer.

4. *A device* j remains in the dormant state and decodes possible uplink grants only at $(j \mod m)$ occasions; otherwise it remains in deep sleep to save energy. If j is pulled, it transfers $y_{\rm j}$ using the assigned radio blocks in its uplink grant (further dedicated control channels ensure that a radio conditions optimized radio blocks allocation after the initial assignment).
5. *The LTE edge cloud*, upon receiving the replica samples, finally recovers the k updated replicas by solving the l_1-minimization problem:

$$\underset{x}{\text{minimize}} \quad \| x \|_1 \quad \text{subject to} \quad \Phi x = y,$$

where x is the $n \times 1$ column vector such that its i-th element corresponds to the original replica $x_{\rm i}$, y is the $m \times 1$ column vector such that its j-th element corresponds to the compressed replica $y_{\rm j}$, and Φ is the $m \times n$ matrix such that its element in the j-th row and the i-th column corresponds to the signature ϕ_{ji} or equals zero if such signature was not used in computing $y_{\rm j}$ (i.e. j did not receive $x_{\rm i}$) or the cloud never defined it (i.e. i is not a direct neighbor to j).

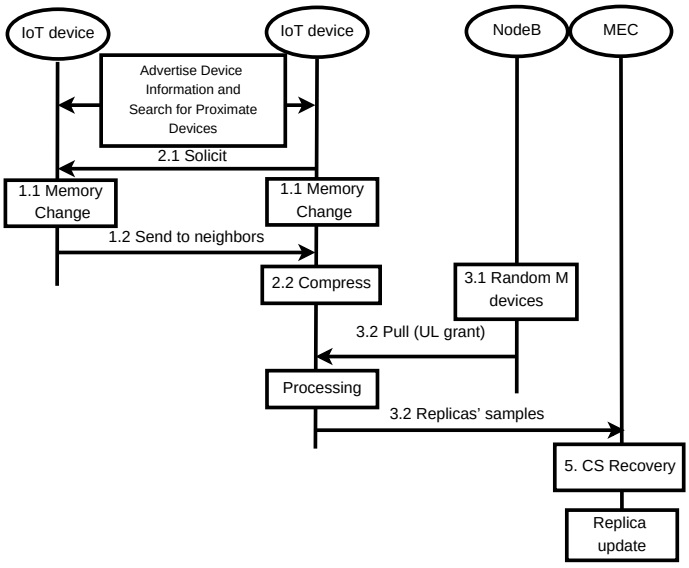

Fig. 4: Proposed memory replication protocol.

1) Protocol Correctness: The REPLISOM memory replication protocol is an application of the theory of compressed sampling with sparse measurement matrices [28]. The correctness of REPLISOM depends on the properties of the Φ matrix (step 5 in the protocol) from the compressed sampling theory [12], and the minimum number of direct neighbors $|Neigh(i)|$ of any IoT device i. On the other hand, the computation and communication overheads of the D2D communication steps of REPLISOM (steps 1.2 and 2.1) depend on the minimum value of d (step 2.1) for an accurate recovery by the theory of sparse compressed sampling recovery [28], [39].

The matrix Φ shall satisfy the *Restricted Isometry Property* (RIP), to ensure the correctness of REPLISOM and an accurate replicas recovery.

Definition 4.1: An $m \times n$ matrix Φ is said to satisfy the $RIP(q)$ if, for any vector x that is k sparse, there exists a constant δ such that

$$(1-\delta)\|x\|_q \leq \|\Phi x\|_q \leq \|x\|_q$$

An accurate recovery is possible if the matrix Φ is sparse and satisfies the $RIP(1)$ property (see Theorem 4 in [28] for formalism).

Theorem 4.1: The $m \times n$ signature matrix Φ, with $m = O(k \log(n/k))$ and $d = O(log(n/k)/\epsilon)$, satisfies the $RIP(1)$ property for $\delta = 2\epsilon$.

Proof: see Appendix A. ■

Numerically, a correct recovery depends on the exact number of neighbors, d, to which a device sends its memory replica in step 1.2 (i.e. the value of ϵ) and the number of active devices k. Intuitively, as k decreases, a device needs to send its replica to more neighbors to ensure information incoherence and a correct recovery. Fig. 5 shows the probability to recover a single replica (out of k) with at most one-bit error and suggests that it is sufficient to design $d = 2\log(n/k)$ for an accurate replicas recovery.

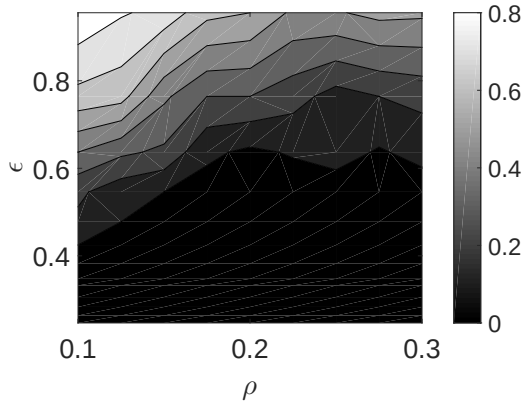

Fig. 5: The probability to recover a memory replica with one-bit error for $n = 2000$, $\rho = k/n$, and $r \geq \sqrt{d/n/\pi}$ (i.e. $r \geq 27$ meters).

B. REPLISOM *Improvement Through Utilizing Memory Sparsity*

It is possible to further reduce the communication overhead of REPLISOM by utilizing the sparsity of memory pages deltas. Consider two consecutive memory pages of an IoT device, p_t and p_{t+1}. Typically, the memory page delta $p_{\rm delta} = p_{t+1} - p_t$ represents an l vector of memory blocks where there are only s blocks that are non-zero. We refer to this as the *sparsity at the memory replica level*.

The straight forward approach to exploit such sparse structure of memory page deltas is to scan through $p_{\rm delta}$ and

represent it using an $O(s\log(l))$ space-efficient sparse vector which contains only the non-zero blocks associated with their relative memory addresses. The memory replica of a device i is then constructed as $x_{\rm i} = \text{float}(device : i,\ time : t,\ memory : p_{\rm delta})$. As the device initially sends p_0 in full to the edge cloud, the cloud simply constructs subsequent pages from $p_{\rm delta}$ (e.g. $p_1 = p_{\rm delta} + p_0$). This approach is not efficient for a large enough l as one must associates every non-zero block with its relative memory page address (i.e. requires $\log(l)$ bits) for sparse representation of $p_{\rm delta}$.

We propose to use compressed sampling to exploit the sparsity of memory page deltas. Compressed sampling requires $w = O(s\log(l/s))$ bits to represent the s-sparse memory deltas. This approach does not require scanning through $p_{\rm delta}$ and works as follows:

1. the cloud generates a random Gaussian and *dense* $w \times l$ matrix, Υ_i, for each device i and sends Υ_i initially to the i-th device (a random matrix Υ_i is sufficient for exact recovery, see [12]),
2. a device i initially includes its p_0 in its replica $x_{\rm i}$,
3. as the device updates its memory, it constructs $x_{\rm i}$ as $x_{\rm i} = \text{float}(device : i,\ time : t,\ memory : p^c)$, where $p^c = \Upsilon_i p_{\rm delta}$,
4. upon recovering $x_{\rm i}$, as discussed in the memory replication protocol, the cloud recovers $p_{\rm delta}$ by solving the l_1-minimization problem:

$$\underset{p_{\rm delta}}{\text{minimize}} \quad \| p_{\rm delta} \|_1 \quad \text{subject to} \quad \Upsilon_i p_{\rm delta} = p^c,$$

5. finally, the cloud determines the full memory replica of the device $p_{t+1} = p_{\rm delta} + p_t$.

V. Benchmarks and Numerical Evaluation

Before defining and evaluating the performance metrics for Replisom protocol compared to the *idle to active* and the *dormant to active* scenarios (Fig. 2 and Fig. 3), we first review the LTE radio frame structure, timing, control channels, and data channels.

A. The LTE Frame and Channels

The LTE time division duplex frame has an overall duration of 10 ms and consists of two half frames (downlink and uplink) each of 5 ms duration and a half frame consists of five subframes each of 1 ms duration. Each subframe carries physical control and data channels that carries logical channels information (see Section 4.5 from [37] for detailed mapping of logical channels to transport and physical channels). The capacity and timing of these channels specifies the latency and energy consumption in replica transfers to the mobile edge cloud. Fig. 6 illustrates the LTE frame along with the typical timing of these control and data information.

The Physical Uplink Shared Channel (PUSCH) is the traffic channel used for an uplink data transmission. The PUSCH contains g radio blocks in each subframe for data transmissions from at most g simultaneous devices. According to its measured radio condition at time t, the i-th device can transmit at most B_t^i bits defined as the transport block size. The device determines B_t^i according to its radio condition (translated into a modulation coding scheme) and the total number of allocated radio blocks (refer to the Table 7.1.7.1-1 and Table 7.1.7.2.1-1 from [36] for details). It is sufficient for our scope to assume that the transport block size is the same for all devices and is time independent. We refer to the transport block size as B.

Separate dedicated and common control channels are responsible for the transport of the radio interface control messages. The uplink half frame contains the Physical Uplink Control Channel (PUCCH) which carries the uplink scheduling requests, which a device uses to request for the PUSCH resources. Generally, each subframe can contain up to b simultaneous scheduling requests from different b devices. Moreover, the uplink half frame contains occasions of the Physical Random Access Channel (PRACH) which carries the random access request information for the initiation of the random access procedure (see Section 5.7 in [40] for random access timing).

The downlink half frame carries two main control information that are necessary for uplink transmission. First, the random access response in the Downlink Shared Control Channel (DL-SCH) which is addressed to a specific device that previously sent a random access request. The time between sending the random access request and receiving the random access response is the average random access delay, $T_{\rm r}$ (Average PRACH delay in Fig. 6). The collisions during random access, the contention resolution procedure, and the propagation delay determine the actual value of $T_{\rm r}$. Assuming a collision free random access, recent studies suggests that $T_{\rm r}$ is between 47 ms and 55 ms [5], [9]. The second control information is the uplink grants which is sent on the Dedicated Downlink Control Channel (PDCCH) that is designated to a specific device which previously sent an uplink scheduling request. Upon receiving a scheduling request it takes $T_{\rm s} \approx 10$ ms (Time to schedule in Fig. 6) to process the request at the eNB and send the uplink grant to the requesting device. The eNB schedules an uplink transmission for any device after $T_{\rm t} \geq 4$ ms (Time to transmit in Fig. 6) from receiving its uplink grant.

B. Memory Replication Performance

Three performance metrics determine the efficiency of the memory replication from k simultaneous active devices in mobile edge computing. First, *the total allocated control channels*, which defines the total number of control channels, C, that the eNB allocates for devices to access the network and initiate an uplink data transmission. Second, *the total replication delay*, which measures the total time $T = \sum_{i=1}^{k} T_i^{\rm acc} + T_i^{\rm data}$ that is required to update the k replicas; where for a device i, $T_i^{\rm acc}$ denote the access latency and $T_i^{\rm data}$ denote its replica transfer time. Third, *the total consumed energy*, which measures the total energy $E = \sum_{i=1}^{k} P_{\rm rx} \times T_i^{\rm rx} + P_{\rm tx} \times T_i^{\rm tx} + P_{\rm inact} \times T_i^{\rm inact}$ to update the k replicas; where $P_{\rm rx}$ and $T_i^{\rm rx}$ denote the consumed power during a reception and the receive time of device i; $P_{\rm tx}$ and $T_i^{\rm tx}$ denote the consumed power during a transmission and the transmit time; and $P_{\rm inact}$ and $T_i^{\rm inact}$ denote the consumed power during an inactivity and the inactive time at which i only

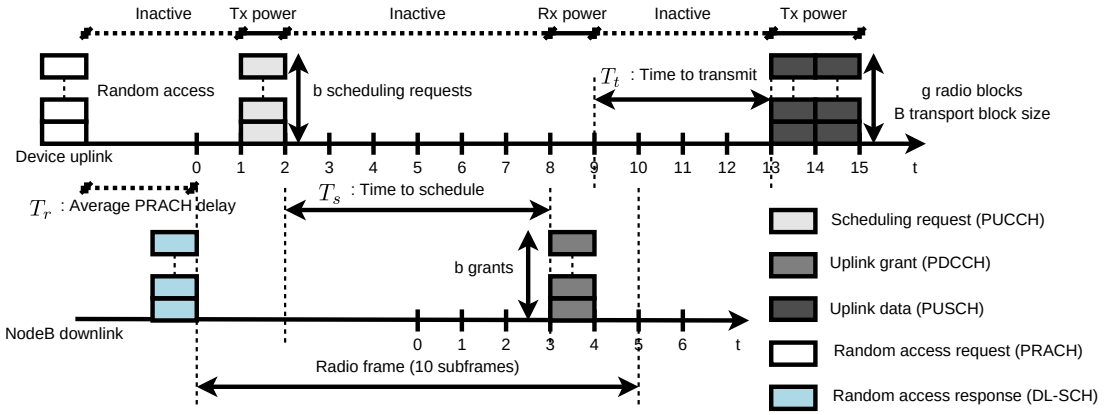

Fig. 6: LTE time division duplex frame timing and main channels.

Symbol	Definition	Default Value / Range
n	total devices per *small* cell	1000 / Up to 50000
a	coverage area of eNB	60 meter^2 / 1 Km^2, $n = 50000$
k	active devices per cell	300 / 30 to 300
m	REPLISOM pulled devices	$m = 2k\log(n/k)$
d	D2D neighbors	$d = 2\log(n/k)$
b	scheduling requests per subframe	18 / 1 to 18
g	radio blocks per subframe	32 / $\{4, 8, 16, 32\}$
B	transport block size per radio block	408 bits / 16 to 584 (see 3GPP)
l	memory replica size	512 bytes / 16 to 2048 bytes
s	memory replica sparsity	$0.1l$ / $0.1l$ to $0.3l$
w	compressed replica size	$w = 2s\log(l/s)$
P_{rx}	consumed power during receive	100 mW
P_{tx}	consumed power during transmit	200 mW
P_{inact}	consumed power during inactivity	10 mW
P_{d2d}	consumed power during D2D	0.1 mW
T_{r}	average PRACH delay (collision free)	50 ms / 47 to 55
T_{s}	time to schedule	10 ms
T_{t}	time to transmit	4 ms

TABLE I: Parameters summary and values for numerical evaluations.

monitors the control channels in an optimized Discontinuous Reception (DRX) mode. Table I summarizes all the used parameters, and the notation definitions with their numerical values that are used in our following evaluation. Table II summarizes the three benchmarks, based on the timing and the channels definitions in Fig. 6, for our proposed memory replication protocol, REPLISOM , with and without applying compressed sampling on the memory deltas and compared to the *idle to active* and *dormant to active* scenarios (Fig. 2 and Fig. 3).

1) The total allocated control channels: Since, REPLISOM requires only m device to be pulled, while each device consumes one uplink grant message, the eNB allocates a less number of control channels (m) compared to using conventional LTE procedures for the uplink data transfer (see Fig. 7). Unlike the conventional LTE procedures which require a number of control channels that scales linearly in n, the number of channels in REPLISOM scales logarithmically in n. In the *dormant to active* scenario, the eNB allocates scheduling occasions to all n devices whether these devices will use them or not in addition to k uplink grants to the active devices. Although this behavior, minimizes the delay and power consumption, it significantly wastes the network resources as it requires the allocation of $n+k$ control channels. In the *idle to active* scenario, the eNB allocates four control channels per device (a random access request, random access response, scheduling request, and uplink grant) requiring a total $4k$ control channels. Although, this scales linearly with k, as k increases this scenario requires a greater number of channels than those required by REPLISOM . Moreover, if the random access occasions, L, are not sufficient compared to the random access intensity γ, the number of control channels used in the *idle to active* scenario increases significantly due to collisions.

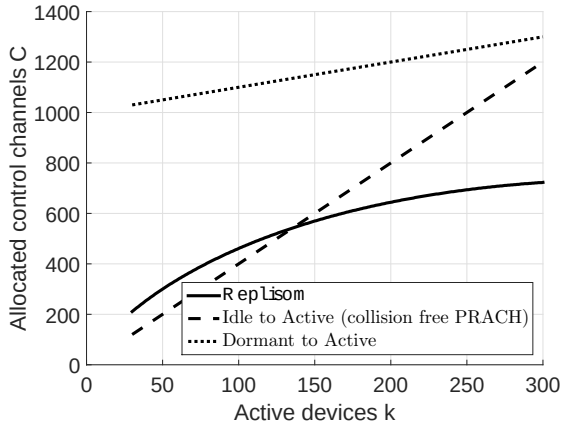

Fig. 7: Allocated control channels for the proposed protocol comparing conventional LTE scenarios.

2) The total replication delay: Given the LTE current procedures, the *dormant to active* scenario achieves the lowest possible uplink replication delay that one can hope for (on the expense of wasted control channels). Similarly, the *idle to active* (under the collision free assumption) scenario resembles our assumed LTE worst case performance. In both cases the total replica transfer time is given by $T^{\mathrm{data}} = \frac{k \times l}{g \times B}$ for all the

Memory replication	C	T [ms]	E [μJ]
REPLISOM	m	$2\frac{d\times l}{B} + \frac{m\times(T_\mathrm{t}+1)}{b} + \frac{m\times l}{(g\times B)}$	$P_\mathrm{d2d}\frac{k\times l}{B} + m\left(P_\mathrm{inact}\times T_\mathrm{t} + P_\mathrm{tx}\frac{l}{B} + P_\mathrm{rx}\right)$
REPLISOM (with compressed replicas)	m	$2\frac{d\times w}{B} + \frac{m\times(T_\mathrm{t}+1)}{b} + \frac{m\times w}{(g\times B)}$	$P_\mathrm{d2d}\frac{k\times w}{B} + m\left(P_\mathrm{inact}\times T_\mathrm{t} + P_\mathrm{tx}\frac{w}{B} + P_\mathrm{rx}\right)$
Idle to Active (collision free random access)	$4k$	$\frac{k\times(T_\mathrm{r}+T_\mathrm{s}+T_\mathrm{t}+2)}{b} + \frac{k\times l}{g\times B}$	$k\left(P_\mathrm{inact}\left(T_\mathrm{r}+T_\mathrm{s}+T_\mathrm{t}-1\right) + P_\mathrm{tx}\left(\frac{l}{B}+2\right) + 2P_\mathrm{rx}\right)$
Dormant to Active (ideal with no random access)	$n+k$	$\frac{n}{k\times b} + \frac{k\times(T_\mathrm{s}+T_\mathrm{t}+2)}{b} + \frac{k\times l}{g\times B}$	$k\left(P_\mathrm{inact}\left(\frac{n}{k\times b}+T_\mathrm{s}+T_\mathrm{t}\right) + P_\mathrm{tx}\left(1+\frac{l}{B}\right) + P_\mathrm{rx}\right)$

TABLE II: Benchmarks of proposed memory replication protocols comparing current LTE generic uplink transmission procedures.

k active devices since the number of radio blocks needed to transmit an l-bits replica is l/B, and the network can transmit at most g blocks simultaneously. The total access delay in the *idle to active* scenario is given by $T^\mathrm{acc} = \frac{k\times(T_\mathrm{r}+T_\mathrm{s}+T_\mathrm{t}+2)}{b}$, which is dominated by the average random access delay, T_r, per device. While in the *dormant to active* scenario the access delay is significantly reduced as it only takes: two subframes for sending a scheduling request and receiving an uplink grant, T_s subframes to schedule the uplink grant, T_t subframes to start the uplink transmission, and $\frac{n}{k}$ subframes between successive scheduling occasions for up to b devices to access the network simultaneously (i.e. $T^\mathrm{acc} = \frac{n}{k\times b} + \frac{k\times(T_\mathrm{s}+T_\mathrm{t}+2)}{b}$).

REPLISOM reduces the total delay required to start replica transmissions in two ways. First, as active devices send replicas to their neighbors in parallel it takes no more than $2\frac{d\times l}{B}$ subframes for the devices to construct replica samples regardless the number of active devices k (step 2.2). Second, as it only takes one uplink grant message to pull the replica samples from the devices, it requires $T_\mathrm{t}+1$ subframes before b devices start to transmit their replica samples (step 4). On the other hand, the total time that is required to transmit all replicas is governed by the values of m and l as $T^\mathrm{data} = \frac{m\times l}{g\times B}$. As the replica size increases, the total delay of the proposed protocol becomes strongly dependent on the uplink data transmission phase and is observably greater than the total delay in the *dormant to active* scenario (see Fig. 8). Fortunately, the sparse structure of memory page deltas improves this undesired behavior where the total size of data transmission improves by a w/l factor. In general, the smaller the replica-size, the improved total delay we observe compared to the conventional LTE scenarios. This restricts the applicability of REPLISOM to replication of tiny sized memory pages (see Fig. 9). Fig. 8 shows that for the same replica size l, the proposed protocol (with compressed replicas) exhibits a total replication delay as the *ideal* dormant to active scenario and is slightly better as k approaches $0.3n$.

3) The total consumed energy: The total consumed energy of REPLISOM is generally worse than the conventional LTE scenarios because it requires m devices to become active compared to k devices in the conventional LTE scenarios (see Fig. 10) although the energy consumed per device during a single replica transmission is significantly less. To see this, consider the inactive, transmit, and receive duration of a single device in the total consumed energy of Table II. In REPLISOM , a device first transmits its replica to its neighbors with a low power for a duration of l/B subframes. Then, the device consumes P_inact power for a duration of T_t compared to $T_\mathrm{r}+T_\mathrm{s}+T_\mathrm{t}-1$ subframes in the *idle to active* scenario and to $\frac{n}{k\times b}+T_\mathrm{s}+T_\mathrm{t}$ subframes in the *dormant to active* scenario. For a replica transmission, a device consumes P_tx power for a duration of l/B subframes, that is reduced to w/B if the sparsity at the memory level is utilized, compared to $l/B+2$ and $l/b+1$ in the *idle to active* and the *dormant to active* scenarios respectively. A pulled device, in REPLISOM , consumes less energy at each pulling occasion, but since a device becomes active more often than in the conventional LTE scenarios it consumes more energy on a longer term.

Fortunately, the energy consumption disadvantage of REPLISOM does not hold true for small enough memory replicas

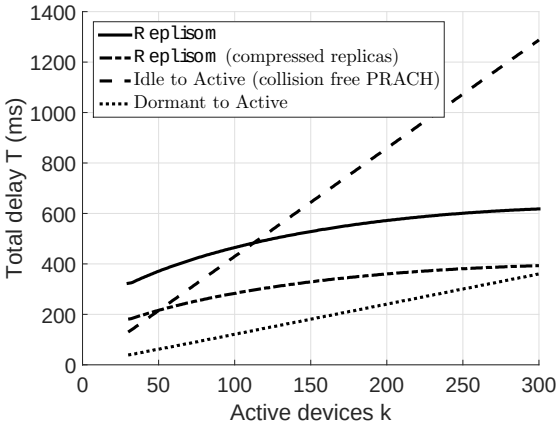

Fig. 8: Total replication delay.

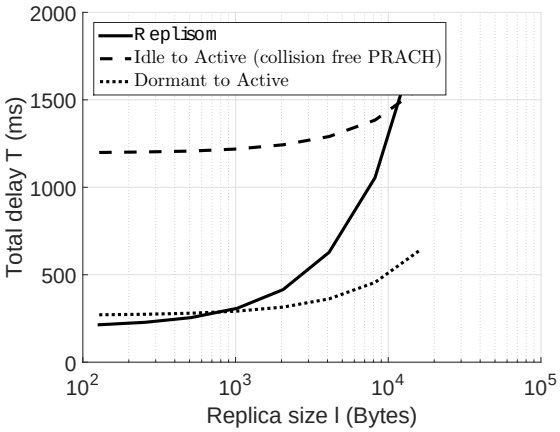

Fig. 9: Replica size impact on delay.

(tiny replicas). This is illustrated in Fig. 11 where energy consumption improves by reducing the replica size, which we also attain by utilizing the sparsity at the memory level. As the memory replicas become smaller, the less energy consumption per a single device activity becomes the dominant energy factor.

Generally, REPLISOM has delay and energy advantages over the conventional LTE scenarios if: the replica size is sufficiently small, or an LTE operator is limited in the number of resource blocks for control channels. Both conditions are of significant practical importance. Although there can be a large number of IoT devices, an individual device generally replicates a small sized data objects (see example applications in [5]). Additionally, the number of radio blocks that are allocated for control channels is limited by the maximum LTE bandwidth (20 MHz) and it is generally in an operator interest to allocate most of the radio blocks as data blocks for conventional network users that have tough quality of service requirements.

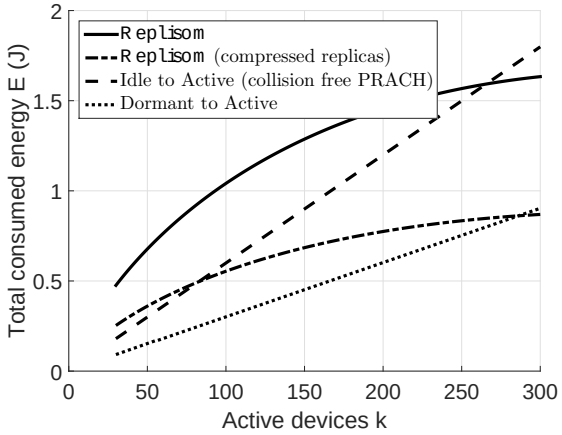

Fig. 10: Total consumed energy.

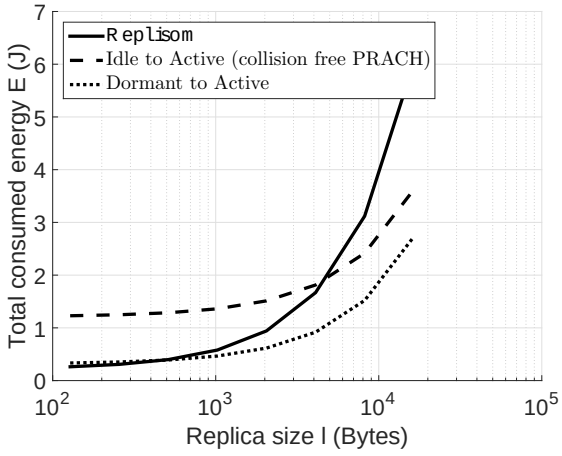

Fig. 11: Replica size impact on power.

4) Impact of Different Parameters: The earlier discussion influenced the impact of the transport block size, B on the delay and energy consumption. It is obvious that as the radio conditions improve and B increases the total delay shall decrease and the device shall consume less energy (as it transmits for a shorter duration); but how the radio condition, hence the transport block size, does influence the delay of REPLISOM compared to the conventional LTE procedures? The poor radio conditions significantly reduce the transport block size, B, and render the proposed protocol to exhibit greater delay than the *idle to active* scenario. However, for moderate and good radio conditions and for the same memory replica size, l, the delay improves rapidly, so as the energy consumption, to approach the *dormant to active* performance as shown in Fig. 12.

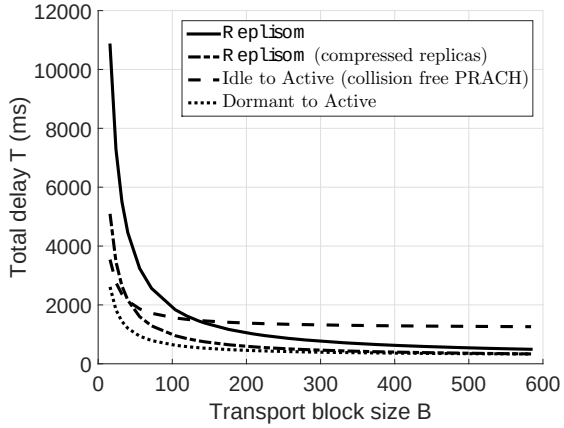

Fig. 12: Transport block size impact on delay.

The total number of radio blocks, g, that are available per subframe also improves the total delay of REPLISOM , but has no impact on the energy consumption. As the radio resources available to the eNB increase (e.g. increase bandwidth), the delay decreases rapidly. However, if the radio resources are limited as in the scenarios where human communication consumes most of the available radio resources, the total delay of REPLISOM becomes worse than the *idle to active* scenario.

VI. CONCLUSION

We propose REPLISOM , a cloud resources augmented eNB architecture with an LTE-optimized memory replication protocol for the Internet of Things applications. REPLISOM works with the in-place LTE technologies and the emerging D2D technologies to efficiently replicate tiny-sized memory pages from a massive number of devices as fast as possible with the minimal control channel requirements via the sparse reconstruction in compressed sampling theory. REPLISOM also utilizes the sparsity at the memory level to further improve the delay and energy consumption. With extensive numerical evaluations of the delay and energy consumption benchmarks, we demonstrate the benefits of REPLISOM to overcome the LTE bottlenecks that arise from simultaneous access of devices for memory replication.

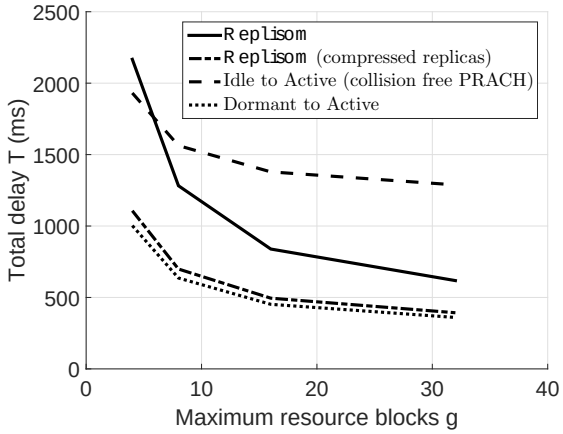

Fig. 13: Radio blocks per subframe impact on delay.

REFERENCES

[1] 3GPP, "Service requirements for Machine-Type Communications (MTC); (Release 13)," 3rd Generation Partnership Project (3GPP), TS TS 22.368, Dec. 2014.

[2] S. Abdelwahab, B. Hamdaoui, M. Guizani, and A. Rayes, "Enabling smart cloud services through remote sensing: An internet of everything enabler," *Internet of Things Journal, IEEE*, vol. 1, no. 3, pp. 276–288, June 2014.

[3] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *Pervasive Computing, IEEE*, vol. 8, no. 4, pp. 14–23, 2009.

[4] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[5] 3GPP, "Study on RAN Improvements for Machine-type Communications; (Release 11)," 3rd Generation Partnership Project (3GPP), TR TR 37.868, Feb. 2014.

[6] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.

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

[8] M. Hasan, E. Hossain, and D. Niyato, "Random access for machine-to-machine communication in lte-advanced networks: issues and approaches," *Communications Magazine, IEEE*, vol. 51, no. 6, pp. 86–93, 2013.

[9] M. Gerasimenko, V. Petrov, O. Galinina, S. Andreev, and Y. Koucheryavy, "Energy and delay analysis of lte-advanced rach performance under mtc overload," in *Globecom Workshops (GC Wkshps), 2012 IEEE*. IEEE, 2012, pp. 1632–1637.

[10] ETSI, "Mobile-Edge Computing," European Telecommunications Standards Institute (ETSI), Technical White Paper, Sep. 2014.

[11] A. Asadi, Q. Wang, and V. Mancuso, "A survey on device-to-device communication in cellular networks," pp. 1801–1819, 2014.

[12] D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.

[13] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "Maui: making smartphones last longer with code offload," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 49–62.

[14] C. Shi, K. Habak, P. Pandurangan, M. Ammar, M. Naik, and E. Zegura, "Cosmos: computation offloading as a service for mobile devices," in *Proceedings of the 15th ACM international symposium on Mobile ad hoc networking and computing*. ACM, 2014, pp. 287–296.

[15] J. Haupt, W. U. Bajwa, M. Rabbat, and R. Nowak, "Compressed sensing for networked data," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 92–101, 2008.

[16] X. Wu, Y. Xiong, P. Yang, S. Wan, and W. Huang, "Sparsest random scheduling for compressive data gathering in wireless sensor networks," in *IEEE Transactions on Wireless Communications*, vol. 13, no. 10. IEEE, 2014, pp. 5867–5877.

[17] S. Andreev, A. Larmo, M. Gerasimenko, V. Petrov, O. Galinina, T. Tirronen, J. Torsner, and Y. Koucheryavy, "Efficient small data access for machine-type communications in lte," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3569–3574.

[18] X. Yang, A. Fapojuwo, and E. Egbogah, "Performance analysis and parameter optimization of random access backoff algorithm in lte," in *Vehicular Technology Conference (VTC Fall), 2012 IEEE*. IEEE, 2012, pp. 1–5.

[19] M. Weiser, "The computer for the 21st century," *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.

[20] J. Flinn, "Cyber foraging: Bridging mobile and cloud computing," *Synthesis Lectures on Mobile and Pervasive Computing*, vol. 7, no. 2, pp. 1–103, 2012.

[21] M. D. Kristensen, "Execution plans for cyber foraging," in *Proceedings of the 1st workshop on Mobile middleware: embracing the personal communication device*. ACM, 2008, p. 2.

[22] M.-R. Ra, A. Sheth, L. Mummert, P. Pillai, D. Wetherall, and R. Govindan, "Odessa: enabling interactive perception applications on mobile devices," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*. ACM, 2011, pp. 43–56.

[23] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *INFOCOM, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945–953.

[24] I. Zhang, A. Szekeres, D. Van Aken, I. Ackerman, S. D. Gribble, A. Krishnamurthy, and H. M. Levy, "Customizable and extensible deployment for mobile/cloud applications," in *Proceedings of the 11th USENIX conference on Operating Systems Design and Implementation*. USENIX Association, 2014, pp. 97–112.

[25] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "Lte-a access, core, and protocol architecture for d2d communication," in *Smart Device to Smart Device Communication*. Springer, 2014, pp. 23–40.

[26] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *Communications Magazine, IEEE*, vol. 52, no. 4, pp. 40–48, 2014.

[27] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *Communications Magazine, IEEE*, vol. 50, no. 3, pp. 170–177, 2012.

[28] R. Berinde, A. C. Gilbert, P. Indyk, H. Karloff, and M. J. Strauss, "Combining geometry and combinatorics: A unified approach to sparse signal recovery," in *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*. IEEE, 2008, pp. 798–805.

[29] N. Nguyen, D. L. Jones, and S. Krishnamurthy, "Netcompress: Coupling network coding and compressed sensing for efficient data communication in wireless sensor networks," in *Signal Processing Systems (SIPS), 2010 IEEE Workshop on*. IEEE, 2010, pp. 356–361.

[30] S. Katti, S. Shintre, S. Jaggi, D. Katabi, and M. Medard, "Real network codes," in *Proc. Forty-Fifth Annual Allerton Conference*, 2007, pp. 389–395.

[31] W. Wang, M. Garofalakis, and K. Ramchandran, "Distributed sparse random projections for refinable approximation," in *Proceedings of the 6th international conference on Information processing in sensor networks*. ACM, 2007, pp. 331–339.

[32] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega,

"Spatially-localized compressed sensing and routing in multi-hop sensor networks," in *Geosensor Networks*. Springer, 2009, pp. 11–20.

[33] Y. Zhang, M. Roughan, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," in *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4. ACM, 2009, pp. 267–278.

[34] F. Fazel, M. Fazel, and M. Stojanovic, "Random access compressed sensing for energy-efficient underwater sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 8, pp. 1660–1670, 2011.

[35] 3GPP, "Cellular System Support for Ultra Low Complexity and Low Throughput Internet of Things; (Release 13)," 3rd Generation Partnership Project (3GPP), TR Draft 45.8xx, Mar. 2015.

[36] ——, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures," 3rd Generation Partnership Project (3GPP), TS TS 36.213, Oct. 2010.

[37] ——, "EUTRA Medium Access Control (MAC) protocol specification," 3rd Generation Partnership Project (3GPP), TS TS 36.321, Sep. 2011.

[38] N. Lee, X. Lin, J. Andrews, and R. Heath, "Power control for d2d underlaid cellular networks: Modeling, algorithms and analysis," pp. 1 – 13, 2013.

[39] A. Gilbert and P. Indyk, "Sparse recovery using sparse matrices." Institute of Electrical and Electronics Engineers, 2010.

[40] 3GPP, "LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation," 3rd Generation Partnership Project (3GPP), TS TS 36.211, Jan. 2011.

[41] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing*. ACM, 2002, pp. 80–91.

APPENDIX A
PROOF OF THEOREM 4.1

In this appendix we prove that Φ satisifies the $RIP(1)$ property, hence recovers all memory replicas x accurately (Theorem4.1). If one shows that Φ relates to the adjacency matrix of an expander graph, then it satisfies the $RIP(1)$ property. Let $G = (U, V, E)$ be a left-d-regular bipartite graph, where U is its set of left vertices, V is its set of right vertices, and $E \subseteq U \times V$ is its set of edges, such that every left vertex in U has exactly d neighbors in V.

Definition A.1: A left-d-regular bipartite graph $G = (U, V, E)$ is an (k, d, ϵ)-expander if any set $S \subseteq U$ of at most k vertices has at least $(1 - \epsilon)d|S|$ neighbors.

ASSUME:

1. $G = (U, V, E)$ is the left-d-regular bipartite graph such that: U represents all the n IoT devices and V represents all the m selected devices by the edge cloud, a left node i is connected to a right node j if the later received the replica x_{i} (in step 1.2 or 2.1), and A is the adjacency matrix such that $A_{ij} = 1$ *iff* $(i, j) \in E$
2. Ψ is a random i.i.d $m \times n$ matrix such that $\Psi_{ij} \propto 1/d$ and $\Phi = A \circ \Psi$ ($\circ$ denote matrix element-wise multiplication)

PROVE: G is an (k, d, ϵ)-unbalanced expander

Proof:

1. The probability that a left vertex i has at least d neighbors for a network density $\eta = \frac{n}{a}$ is given by
$$Pr(|Neigh(i)| \geq d) = \left(1 - \sum_{i=0}^{d} \frac{(\eta\pi r^2)^i}{i!} e^{-\rho\pi r^2}\right)^n$$
(see Theorem 2 in [41] for details)
2. for a dense network (e.g. $n = 50000$, $a = 1$, $r = 0.2$), i has at least d neighbors almost surely
3. for $S \subset U$ such that $|S| \leq k$ and $M \subseteq V$ such that $|M| \leq m$, the neighborhood of S is completely contained in M with probability
$$Pr(Neigh(S) \subseteq M) \leq \left(\frac{|M|}{m}\right)^{d|S|}$$
4. G is not an expander if $|M| \leq (1 - \epsilon)d|S|$
5. Let Pr' denote the probability that G is not an expander and is bounded by
$$\begin{aligned} Pr' &\leq \sum_{i=1}^{k} \binom{n}{i} \binom{m}{(1-\epsilon)di} \left(\frac{1-\epsilon)di}{m}\right)^{di} \\ &\leq \sum_{i=1}^{k} \left[\underbrace{\left(\frac{ne}{i}\right) \left(\frac{me}{(1-\epsilon)di}\right)^{(1-\epsilon)d} \left(\frac{(1-\epsilon)di}{m}\right)^{d}}_{z} \right]^i \\ &\leq \sum_{i=1}^{\infty} z^i = \frac{z}{1-z} \end{aligned}$$
6. as $i \leq \rho n$ and $d = O(log(n/k)/\epsilon)$, then $z \leq \frac{1}{10}$ and $Pr' \leq \frac{1}{8}$

■