Data Driven Selection of DRX for Energy Efficient 5G RAN

Diarmuid Corcoran*‡, Loghman Andimeh*, Andreas Ermedahl*, Per Kreuger†, Christian Schulte‡
*Ericsson AB, Sweden, email: firstname.lastname@ericsson.com
†RISE SICS AB, Sweden, email: per.kreuger@ri.se
‡KTH, Sweden, email: cschulte@kth.se

***Abstract*—The number of connected mobile devices is increasing rapidly with more than 10 billion expected by 2022. Their total aggregate energy consumption poses a significant concern to society. The current 3gpp (3rd Generation Partnership Project) LTE/LTE-Advanced standard incorporates an energy saving technique called discontinuous reception (DRX). It is expected that 5G will use an evolved variant of this scheme. In general, the single selection of DRX parameters per device is non trivial. This paper describes how to improve energy efficiency of mobile devices by selecting DRX based on the traffic profile per device. Our particular approach uses a two phase data-driven strategy which tunes the selection of DRX parameters based on a smart fast energy model. The first phase involves the off-line selection of viable DRX combinations for a particular traffic mix. The second phase involves an on-line selection of DRX from this viable list. The method attempts to guarantee that latency is not worse than a chosen threshold. Alternatively, longer battery life for a device can be traded against increased latency. We built a lab prototype of the system to verify that the technique works and scales on a real LTE system. We also designed a sophisticated traffic generator based on actual user data traces. Complementary method verification has been made by exhaustive off-line simulations on recorded LTE network data. Our approach shows significant device energy savings, which has the aggregated potential over billions of devices to make a real contribution to green, energy efficient networks.**

Keywords*—*Software architecture, 5G mobile communication, Adaptive systems, Energy efficiency, Green computing.

I. Introduction

The LTE (Long Term Evolution) and LTE-A (LTE Advanced) wireless access systems [1] represent the forth generation of mobile access technologies. In contrast to early systems such as GSM or UMTS [2], LTE is designed as a fully packet-based system. As such it is suitable for handling a wide variety of device and traffic types. LTE can seamlessly handle streamed video, http-style web browsing and voice over LTE (VoLTE) [3]. Devices such as smart phones, touch-screen tablets and pervasive small low-power machine-to-machine compute devices are some of the various device categories that routinely connect to LTE radio networks. Figure 1 shows an overview of a typical mobile system, with the main components being a core network, a RAN (Radio Access Network) including RBS (Radio Base Station) of various generations, e.g LTE or 5G RBS and end-user devices. The type and number of connected devices are forecasted to increase significantly in the coming years [4] and one growing concern is their total aggregate energy consumption. LTE currently includes

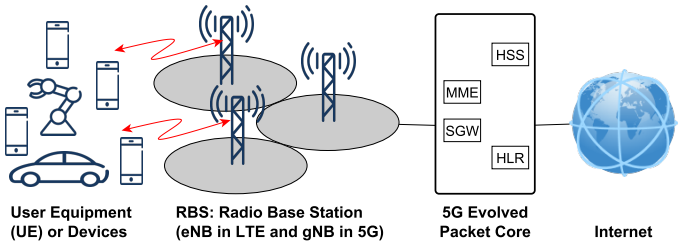

Fig. 1: Radio Access Network Overview

a mechanism to regulate the sleep mode of devices attached to a network. Using this mechanism, called Connected Mode DRX (Discontinuous Reception) or CDRX, the network can set a suitable sleep and wake cycle for each network connected device individually.

According to the LTE standard for radio resource control (RRC) [5], a device attached to the network can be in either of the states RRC_IDLE or RRC_CONNECTED. As illustrated in Figure 2, when the device is connected to the network in state RRC_CONNECTED, there is an opportunity to configure a set of parameters with the RRC-Connection-Reconfiguration message [5] that control a DRX sleep cycle. By adapting these parameters to the particular traffic type and pattern, a device can achieve considerable battery savings. Across many devices, this can lead to significant system energy reduction and make a real contribution to greener mobile networks. The key challenge, however, is finding a good set of DRX parameters for each specific device and dynamically changing them as the traffic density and pattern changes. A common practice in industry is to set a fixed system wide DRX per traffic type, e.g. voice (VoLTE) or mobile broadband. The key motivation for this approach is typically simplicity.

There has been considerable modeling and simulation work carried out on the DRX problem, see Section III. Our work differs from previous work by being based on a practical data-driven approach, allowing for, per user device, real-time DRX parameter adaption due to changing patterns in traffic. We achieve this by: monitoring the traffic profiles of devices; extracting data samples from the user plane [2] packets; applying the samples to an energy model which derives a good set of operating DRX parameters per device; and reconfigure the devices' DRX settings accordingly.

More broadly, telecommunication networks such as LTE, depend on a vast array of parameters to tune network performance and operational efficiency. E.g., the radio resource

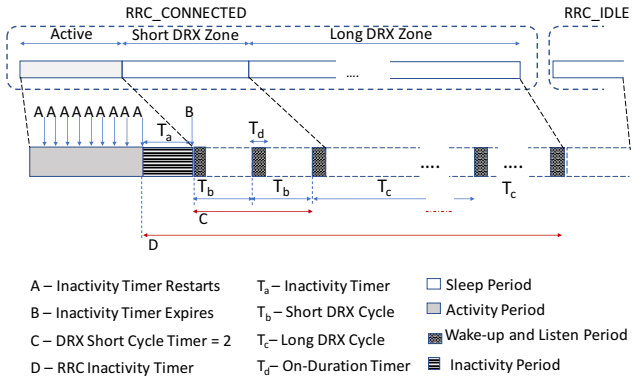

Fig. 2: Overview of DRX Mechanism

configuration protocol [5] exposes several hundred parameters that are vital to the correct and efficient operation of an LTE network. In conjunction with these, parameters from the transport network, the hardware and software resource configuration domain creates a very large space of configurable parameters. As a consequence, we stand before an immensely complex task of configuring and tuning networks for operational efficiency. In this paper, we propose a data driven architecture for future mobile networks, such as 5G. This new approach facilitates the broader uses of sophisticated closed loop automation. The key contributions are:

- A novel offline scheme for estimating a good working set of DRX combinations from a much larger set for a given traffic profile.
- A fast energy model algorithm to evaluate a good DRX parameter combination for a device.
- An online scheme using either the fast energy model or a derived neural network model to determine a good DRX parameter set for the observed traffic profile.
- An implementation in a real LTE system to evaluate the technique.
- A software architecture for applying data driven and machine learning performance improvements in telecommunication networks.

II. The LTE DRX Mechanism

The details of the LTE DRX mechanism are specified in [5], [6], this section provides a brief overview. A device in LTE can be in either *connected* or *idle* mode. When a device is in idle mode (RRC_IDLE), the idle mode DRX mechanism (DRX paging mode), is used. We consider here only the use of the DRX mechanism in *connected* mode, (RRC_CONNECTED). In this mode, after periods of activity on the uplink (UL) or downlink (DL), the device can enter the short DRX zone followed by a long DRX zone after which the device will enter the idle mode. In the different zones the DRX parameters will determine how often, how many times and with what frequency the device will awake and check for data. If data is available the active zone will be re-entered again. Figure 2 shows an example sequence.

We use five parameters to tune the DRX device setting: 1) **Inactivity-Timer**, 2) **Short-DRX-Cycle**, 3) **Short-DRX–Cycle-Timer**, 4) **On-Duration-Timer** and 5) **Long-DRX-Cycle**. Note that 3gpp [6] names the parameters as follows: drx-InactivityTimer, shortDRX-Cycle, drxShortCycleTimer, onDurationTimer and longDRX-CycleStartOffset. During a period of activity indicated as **A** in Figure 2, the Inactivity-Timer, indicated as $\mathbf{T_a}$, is restarted on each receive/send event. On expiration of $\mathbf{T_a}$, the Short-DRX-Cycle is entered, indicated as $\mathbf{T_b}$. Each Short-DRX-Cycle consists of an On-Duration-Timer, indicated as $\mathbf{T_d}$, plus a sleep period. The length of the Short-DRX-Cycle sleep period is not given explicitly but derived as $\mathbf{T_b}$ - $\mathbf{T_d}$. During the On-Duration the device wakes up and listens to a special LTE downlink control channel called PDCCH (Physical Downlink Control Channel), where it will be notified of pending downlink data. If data is available the active zone **A** is entered and the Inactivity-Timer is restarted. If no data is pending the short cycle is repeated the number of times indicated by Short-DRX-Cycle-Timer, illustrated as **C**.

Once the Short-DRX-Cycle has been repeated Short-DRX-Cycle-Timer times, the Long-DRX-Cycle, indicated as $\mathbf{T_c}$, is entered. Each Long-DRX-Cycle consists of an On-Duration-Timer, indicated as $\mathbf{T_d}$, plus a sleep period. Again the sleep period is not explicitly given but derived as $\mathbf{T_c}$ - $\mathbf{T_d}$. There is no operating parameter to specify how many Long-DRX-Cycle events to repeat and so, assuming no pending data is detected during an On-Duration, the Long-DRX-Cycle will be repeated the number of times allowed before yet another timer called RRC-Inactivity-Timer expires, indicated as **D**. After that the device will enter the RRC_IDLE mode. While in this mode the DRX mechanism described here does not apply. There is one additional DRX timer called the Retransmission-Timer specified by 3gpp [5], [6] which is used when waiting for retransmissions. We currently do not consider this in our model, which is a minor restriction. There is, however, no obstacle to including it as a future refinement.

The short DRX mode is optional and in scenarios where it is not configured or not supported by a device the long DRX mode will be entered directly. One important aspect of connected mode DRX is that it trades latency for device energy efficiency. By reducing the On-Duration-Timer and increasing the Short-DRX-Cycle and/or Long-DRX-Cycle, for example, the device's potential to reduce power consumption is improved by allowing longer sleep periods. A consequence of this is that data will need to be buffered in the network RBS, leading to increased resource usage in the RBS and higher latency on the device side.

III. Related Work

Related work has established that DRX is an essential mechanism for saving energy of UEs and that the choice of parameters for DRX is challenging. Evaluations of the trade-off between latency and energy saving are based on simulation rather than real traffic data and real lab environments. This is in contrast to the approach presented in this paper which is based on real, recorded, traffic patterns run against a fully functional LTE test system. Zhou [7] models (using a semi-Markov process) and simulates the effect of adjustable short and long DRX cycles using a bursty traffic model. A fixed On-Duration-Timer is assumed, which is a limitation. Our approach, in contrast, is flexible enough to manage variation in DRX parameters. Zhou [7] also does a comparison with the older UMTS [2] DRX mechanism, showing that the LTE approach is more effective. Mihov [8] performs a very

similar analysis but with a slight refinement that claims to more accurately estimate device savings from adjustable long and short cycles. Fowler [9] has analyzed the effect of the LTE subframe size on the DRX mechanism, concluding that increasing the TTI (Transmission Time Interval) or subframe leads to better DRX related power saving. Fowler also notes that varying types of traffic, for example voice or web, may benefit differently from use of long or short DRX cycles, indicating the benefit of having an adjustable and dynamic DRX approach. Karthik et al. [10] attempt to derive a practical DRX algorithm which trades energy against latency. First an analytical framework is constructed and then used to find suitable DRX alternatives by comparing against a set network value. An evaluation is then carried out where battery-life on a UE (User Equipment) is compared using the network setting and that suggested by the framework. In this case there is a slight degradation in battery-life indicating the difficulty of choosing good values. Our algorithm differs as it targets live traffic patterns and updates DRX according to a chosen selection criteria such as energy, latency or both. Kolding et al. [11] show the importance of dynamically changing the DRX parameters to achieve power savings. Their model experiments with On-Duration-Timer and Inactivity-Timer where they show good power-gains for a single user but run into complexity issues when dealing with several UE's. They also note that an optimal On-Duration-Timer is difficult to preset as a priori knowledge of traffic patterns is needed. These issues are less of a limitation in our approach, again due to on-line data-driven nature of our algorithm. We also deal with many UE's per RBS without difficulty. Jha et al. [12] again explore the energy-latency trade-off using OPNET modeler simulations. They also propose an algorithm for trading energy against latency. Stea et al. [13] build a detailed simulation model using the OMNeT++ framework. This is then used to do simulation against various traffic models such as http, VoIP (Voice Over IP) and video on demand. The simulations indicate that it is possible to improve energy usage without substantially degrading QoS (Quality of Service) through increased latency. Lauridsen [14] presents a very useful smartphone power model which shows the importance of connected mode DRX in power saving as well a system model for smartphone modem power usage. We will apply aspects of this model to our results to estimate total device energy savings. Lauridsen also does a very detailed study of mobile terminal energy consumption for LTE and future 5G in his PhD thesis [15]. Tseng [16] et al. perform a theoretical analysis with long and short DRX cycles then verify their analysis through simulation. For simplicity the Inactivity-Timer was not used, which is a limitation given that previous work [11] indicates the Inactivity-Timer to be key to power saving. Bo et al. [17] propose an addition to the RBS scheduling algorithm which prioritizes UE's or devices that have the shortest return to DRX sleep DRX parameter settings. Tung et al. Ergul [18] proposes a new scheduling algorithm that considers DRX and claims to be QoS and energy efficiency aware. Fowler et al. propose extending the current three-state DRX mechanism (Active, Short-Sleep and Long-Sleep) to a four or five state mechanism. They use a semi-Markov process to generate analytical models which are used for simulations to compare the standard 3-state mechanism against a four or five-state extension. The simulation results show potential improvements but also conclude that real-traffic measurements are need for evaluation. The first theoretical

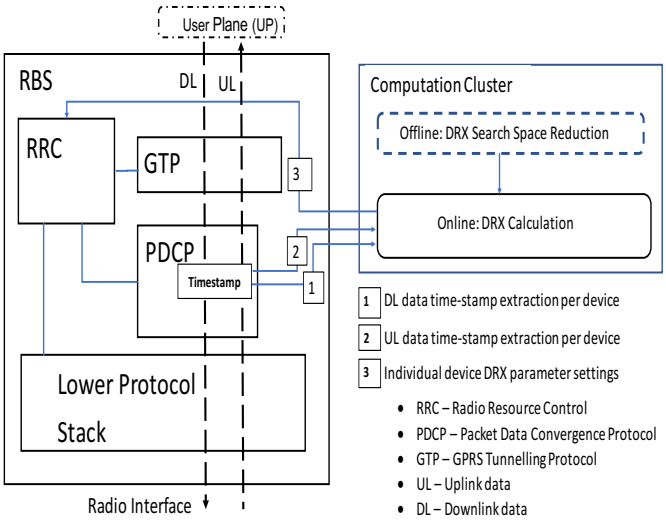

Fig. 3: Overview of System Architecture

studies of applying DRX to 5G with multi-beams have also started to appear. Agiwal [19] et al. do a study of applying a directional DRX mechanisms to a device with several radio beams. They conclude that DRX will be an important energy saving technique in 5G.

Our work differs from all the above in that we apply a practical data driven approach using real traffic. We develop a model and algorithm for estimating device energy efficiency. We evaluate this in a real, lab based LTE network. To our knowledge this is the first study to apply such an approach.

IV. System Architecture

To support data driven DRX selection, we changed the RBS protocol stack to lift out the arrival times of uplink (UL) and downlink (DL) user plane packets. The packet arrival time extraction is done at the PDCP (Packet Data Convergence Protocol) [20] layer and transported to an RBS external compute cluster. This is motivated by the fact that RBSs are, in general, resource-constrained embedded devices. There is also a trend towards advanced machine learning assisted resource management and an external compute cluster is seen as most suitable for this.

Figure 3 sketches the overall architecture. Time-stamped events are streamed per UL and DL PDCP data package through interfaces 1 and 2 for each connected device. The external compute cluster collects the events into separate UL and DL time-boxed structures per connected device. The compute cluster has two distinct parts: • **On-line**: This part analyses the time-boxed DRX structures per device and suggests a good working set of DRX parameters. This computation is performed in real-time. Details are covered in Section V. • **Off-line**: This part supports the on-line part by reducing the DRX search space. It is considered off-line as it is not part of the real-time traffic chain.

Our original system architecture had only an on-line computation component. The off-line part has been added later to reduce the working set of DRX combinations considered by the on-line algorithm. At the end of each time-boxed analysis period the on-line algorithm may suggest a better set of DRX

TABLE I: Specified Values for DRX Parameters

DRX Parameter Space	
Parameter	Possible Values of {i}
IT: Inactivity-Timer (ms)	1,2,3,4,5,6,8,10,20,30,40,50,60,80, 100,750,1280,1920,2560
SC: Short-DRX-Cycle (ms)	2,5,8,10,16,20,32,40,64,80,128,160, 256,320,512,640
SCT: Short-DRX-Cycle-Timer (iterations)	1,2,3...
ODT: On-Duration-Timer (ms)	1,2,3,4,5,6,8,10,20,30,40,50,60,80, 100,200
LC: Long-DRX-Cycle (ms)	10,20,32,40,64,80,128,160,256,320, 512,640,1024,1280,2048,2560

parameters for the traffic just observed. This suggestion is communicated to the RRC (Radio Resource Control) protocol element through a new control interface [3]. RRC has the responsibility to distribute new DRX settings to other RBS protocol elements as well as the device. The effect of changing DRX settings for a device has complex consequences within the RBS. The mechanism which schedules air interface access for each device must be updated to ensure synchronization between the RBS and device. Overall scalability of the system is limited by interfaces [1] and [2]. To prevent network overload by many high frequency events techniques like event aggregation or limiting number of active UE DRX calculations per RBS can be used. The on-line component is designed to scale across all available cores and multiple compute blades.

V. Data Driven DRX Parameter Selection

As indicated by Figures 1 and 3, the RBS configures capable devices with DRX parameters. The possible combined space of parameters, shown in Table I, is large. The parameters IT,SC,ODT,LC are specified in terms of TTI (Transmission Time Interval), which in LTE translates to one millisecond. The parameter SCT specifies the number of iterations of the short cycle. These values are specified by 3gpp [5] and are subject to change between protocol revisions. A common approach among mobile system operators is to choose a common, system-wide, DRX setting $\{IT_i,SC_i,SCT_i,ODT_i,LC_i\}$, per traffic type (i.e. VoLTE, web based) where i represents one of the possible values for each parameter in Table I. This simplifies system management but ignores the opportunity to reduce device power consumption.

In our approach we allow the system to select a good working DRX for each device. Good in this context involves selecting the DRX combination that improves device energy usage most, without degrading latency substantially. It is also possible to select based on improved energy only while allowing latency to degrade or to use a combination of both strategies. We consider mainly mobile broadband traffic and not VoLTE [3] which has its special needs on DRX settings. As described in Section IV, packet data arrival events are streamed from the PDCP layer to our external compute cluster. The events are tagged as UL or DL for each device and a high-resolution UTC (Coordinated Universal Time) time stamp is included. An observation period **N** is defined in terms of TTI, which in the LTE system is equivalent to one millisecond (ms). In our experimental set-up we used N = 1000 and 10000 ms, but other periods are also possible. A high-resolution arrival time is used to box each UL and DL packet into one

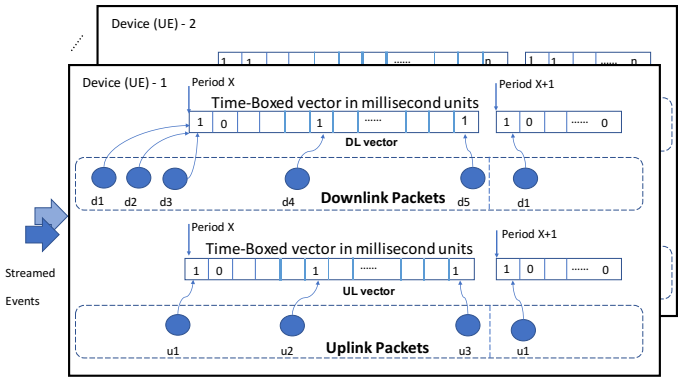

Fig. 4: Time-Boxing UL and DL Packets

millisecond buckets within this observational period. Figure 4 illustrates the mechanism. Time-boxes are realized as vectors of **N** bits where activity in a TTI (or one millisecond period) is represented as ON (1) while no activity is represented as OFF (0). At the end of each period the UL and DL activity vectors per active device are passed to a device energy model after which the observational period restarts. In our current implementation the boundary of each period is determined by the time-stamp on the incoming messages and thus the shift from observational period **x** to **x+1** is event driven.

An outline of the energy model is made clear in Algorithm 1. The input is uplink **U** and downlink **D** vectors as described above, and (implicitly) a DRX parameter combination search space D_{rx}. The output is a list W of: an energy value E and a latency vector **L** per input DRX combination d_i, in D_{rx}. The calculation of E and **L** can be parallelised and this is also how we implemented the algorithm.

To calculate E and **L** we build an activity vector **A** from **U** and **D**. **A** is first initialized to **U** (line 2). This is important as **U** represents a good fingerprint of very recent activities in the device. Next the inactivity timer (IT) is applied to **A** for each ON TTI (one millisecond time-box), giving an updated vector **A** (line 3). After the IT insertion we find the active TTIs in **D** that intersects with **A**. The Merge operation (line 11) performs this, returning an updated **D** with intersecting TTI in **D** now set to OFF. At the intersecting TTI position in **A**, the inactivity timer is again inserted (line 13). This procedure is iterated until there are no further intersecting active (ON) TTI in the updated **A** and **D** (line 12 and 15). In this way overlapping TTI in **D** are successively moved into **A**, and so that on termination, remaining ON TTIs in **D** are those where at least one downlink packet has been delayed.

While possible, short and long cycles plus associated on-duration-timer (ODT) durations are inserted into the silent (OFF) TTI in **A**. The SilentPeriods operation (line 16) takes **A** and **D** and returns **A'** and **D'** as a list of non-overlapping sub-vectors of **A** and **D**, corresponding to the silent (OFF) TTI in **A**. Changes to **A'** and **D'** will also affect **A** and **D**. Short and long cycles are then applied to each of these silent sub-vectors. The insertion of these cycles can result in a recursive call to the process described above, while there is still room in each sub-vector. Otherwise the insertion process completes for the current sub-vector, once the vector end has been reached. The construction of **A** terminates when there are finally no more overlapping regions in **A** and **D**.

Input: U (uplink vector), **D** (downlink vector) of size N
Output: W is list of: Energy value E, latency vector $\mathbf{L}$ per DRX parameter combination d_i

1 **ParallelForEach** d_i *in* D_{rx}
2 $\quad \mathbf{A} \leftarrow \mathbf{U}$
3 $\quad \mathbf{A} \leftarrow InsertInactivityTimer(\mathbf{A}, d_i.IT)$
4 $\quad (\mathbf{A}, \mathbf{D}) \leftarrow ConstructActivityVector(\mathbf{A}, \mathbf{D})$
5 $\quad E_i \leftarrow (\sum_{n=0}^{N-1} \mathbf{A}[n])/N$
6 $\quad \mathbf{L}_i \leftarrow Delays(\mathbf{A}, \mathbf{D})$
7 $\quad$ [W|$(E_i, \mathbf{L}_i, \mathbf{d}_i)$]
8 **return** W
9 **def** *ConstructActivityVector(**A**, **D**)*
10 $\quad$ **do**
11 $\quad\quad (\mathbf{A}, \mathbf{D}) \leftarrow Merge(\mathbf{A}, \mathbf{D})$
12 $\quad\quad$ **if** $\boldsymbol{A} \cap \boldsymbol{D} \neq \varnothing$ **then**
13 $\quad\quad\quad \mathbf{A} \leftarrow InsertInactivityTimer(\mathbf{A}, d_i.IT)$
14 $\quad\quad$ **end**
15 $\quad$ **while** $\boldsymbol{A} \cap \boldsymbol{D} \neq \varnothing$
16 $\quad S \leftarrow [(\mathbf{A'}, \mathbf{D'})] \leftarrow SilentPeriods(\mathbf{A}, \mathbf{D})$
17 $\quad$ **for** $(\boldsymbol{A'}, \boldsymbol{D'})$ *in S* **do**
18 $\quad\quad (\mathbf{A'}, \mathbf{D'}) \leftarrow InsertShortCycle(\mathbf{A'}, \mathbf{D'})$
19 $\quad\quad (\mathbf{A'}, \mathbf{D'}) \leftarrow InsertLongCycle(\mathbf{A'}, \mathbf{D'})$
20 $\quad$ **end**
21 $\quad$ **return** (**A**,**D**)
22 **def** *InsertShortCycle(**A'**,**D'**)*
23 $\quad \mathbf{A'} \leftarrow InsertONDuration(\mathbf{A'}, d_i.SC, d_i.ODT)$
24 $\quad (\mathbf{A'}, \mathbf{D'}) \leftarrow ConstructActivityVector(\mathbf{A'}, \mathbf{D'})$
25 **def** *InsertLongCycle(**A'**,**D'**)*
26 $\quad \mathbf{A'} \leftarrow InsertONDuration(\mathbf{A'}, d_i.LC, d_i.ODT)$
27 $\quad (\mathbf{A'}, \mathbf{D'}) \leftarrow ConstructActivityVector(\mathbf{A'}, \mathbf{D'})$

Algorithm 1: ENERGY MODEL

Building **A** successively moves OFF TTI to ON, according to Algorithm 1. The upper-bound on termination would then be **A** with all TTI active (ON). At this point the energy value E (where $0 < E \leq 1$) can be calculated (line 5), while all remaining active (ON) TTI in **D** represent delayed downlink packets. E is a measure of energy usage, where a minimized value is preferable when trying to reduce power consumption. A latency vector **L** can be calculated as the distance from each ON TTI in **D** to the next period of activity(ON) in **A** (line 6). This gives a distribution of delays which can be used in service criteria selection. A basic selection technique would be to use the average of this distribution. A more sophisticated criteria might be to select the best E where e.g. the 90:th latency percentile are better than a given value.

In our experiments we used a baseline DRX D_b that gives us $(E_b, \mathbf{L}_b)$ which can be compared, using the chosen criteria, against W. In our lab setup we selected a new working DRX by choosing a d_i from W where the value of E_i is minimized while the average of $\mathbf{L}_i$ is not worse than $\mathbf{L}_b$. We call this criteria **best selection**. Other possible criteria are **best latency**: minimize latency and **best power**: minimize energy use.

In our test and simulation environments, Algorithm 1 was implemented in Python using Numpy [21] for high performance vector operations. The implementation ran in order of ten's of milliseconds to ten's of seconds for N=10000, depending on the DRX search space size. Exact performance is dependent on the compute cluster HW, but the implementation is efficient enough to run in real-time.

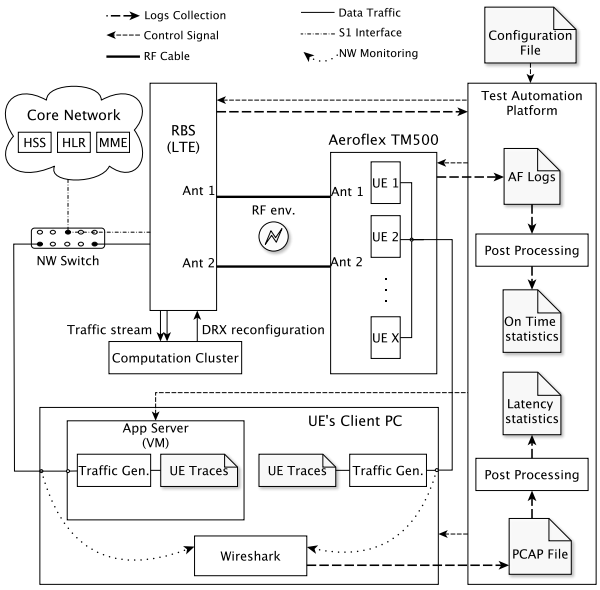

Fig. 5: Test Lab Configuration

VI. EXPERIMENTAL SETUP

Our test system setup is illustrated in Figure 5. It included a LTE Radio Base Station (RBS) [2], connected to a core network, and a multi-UE simulator (Aeroflex TM500) [22], emulating the user devices (UEs). To run the TM500 in multi-UE mode and to prevent radio interference from outside, the eNB radio antenna was connected to UE's radio antenna over two cables, where MIMO [2] configuration is used for all tests. The on-line DRX reconfiguration module, described in Section 3, ran on the compute cluster. Software updates of the RBS allow us to extract events including UL and DL traffic time stamps from the RBS PDCP module. The RBS RRC module was adapted to receive DRX reconfiguration messages from the compute cluster and to update the different UE's DRX settings accordingly (by sending 3GPP RRC-Connection-Reconfiguration [5] messages to the UEs). For the RBS to work correctly it was connected to a core network including HSS and MME functions [2]. The core network was also in charge of transferring IP packets from internet-connected servers to the RBS. To eliminate latency variations from other traffic occurring in the core network we used a network switch with special routing capabilities. This allowed us to bypass the core network and instead directly forward the application server packets to the RBS. As a practical matter we also disabled the RRC-Inactivity-Timer [5]. A UE only has a transient identity in the RAN, upon entering RRC_IDLE state this identity will be forgotten. Thus in order to ensure consistent tracking of UE under test the timer was disabled. In addition to this we are primarily concerned with experimenting with DRX in RRC_CONNECTED mode.

Traffic generation: We implemented traffic generators (called App Server and UE's client) in Erlang [23] for generating UL and DL traffic according to UE traces. For each UE we allocated two IP addresses, one in the App server and one in the UE's client. Each traffic generator then created two separate Erlang processes per UE, one for sending and one for listening to UDP traffic on the IP addresses dedicated to the particular UE. To make the App server and the UE's client

TABLE II: Key statistics of the UE traces studied

	trcs	avg. size & duration		avg. rates (pkts/s) - quartiles, average & median				set IAT (ms)		
set	#	# pkts	secs	25%	med	avg	75%	avg	95%	99%
A	848	110 802	4 852	7 020	13 548	24 782	28 093	44	10	75
B	277	193 836	4 793	12 776	30 876	44 119	58 378	25	8	34
C	50	6 185	132	34 488	53 377	74 344	80 780	21	67	160
D	4	15 964	462	207	2 124	26 959	28 876	29	40	100
E	8	9 062	1 146	4 082	8 515	7 680	10 725	127	27	231

traffic generators time synchronized we ran both programs on the same Linux PC. The UE's client program ran directly on the host machine while the App server ran in a VM. The App server also used the Client PC as Network Time Protocol[24] (NTP) server, since time synchronization was needed to make sure that two test run with the same trace file will result in the same UDP UL/DL traffic. Moreover, when deriving traffic latency, all time stamps of messages stored in the Wireshark PCAP files should based on the same hardware clock.

Statistics derivation: We used Wireshark [25] to derive traffic latency statistics for different UEs by monitoring the used network interfaces and capturing all incoming and outgoing data traffic on them. By running both the App server and the UE's client program on the same PC we could capture and save packets send- and arrival times in one common PCAP file. Aeroflex logs were used to derive on-time statistics for different UEs. The logs recorded the DRX state the UE was in during different TTIs and were then post-processed to derive the type of statistics found in Section VII-A.

UE traffic traces: Emulating user behavior in terms of traffic properties (and mobility) accurately enough to evaluate an on-line DRX is non-trivial. This is because the DRX decision mechanism reacts to packet inter-arrival times on a time scale ranging from a few milliseconds up to around a minute, and generating synthetic data with realistic inter-arrival time distributions on such a time scale is very challenging. Instead we extracted traffic traces from logs recorded by LTE core networks and used these for trace based simulations. Even though the accuracy of timing information differs from up- and down-stream traffic, since time stamps of up-stream traffic reflect the arrival of a packet at the measuring point which can differ somewhat from the time the UE actually tried to send it, such traces are rich in different types of behaviors, with instances of both sparse and intense traffic and with a range of more or less bursty traffic patterns that reflect realistic conditions to a reasonable degree.

The traces in the recorded network logs vary from very sparse to extremely dense. Since traces with or without very long silent periods are irrelevant for the study of DRX we selected traces from a range of trace sizes and average throughput Table II summarizes key statistics of the trace sets used in our evaluations. All trace sets were selections from one large recorded network traffic log. Sets A and B are selected based on average data density. Sets C, D and E are random selections from set B, which is subset of A. Set E furthermore has long silent periods removed and trace durations restricted to between 15 and 20 minutes. Due to limitations on the number of available TM500 UEs only sets D and E have been used in the lab tests. In the off-line simulations all trace sets have been analyzed.

VII. Evaluation

As explained in Section VI, we performed evaluations using both a radio network lab (using trace sets D and E) and an off-line simulator (using all trace sets from Table II).

A. Evaluation in the Radio Network Lab

For the radio network lab evaluations a set of 18 commonly used DRX combinations was chosen (collected from operational networks setup), as well as a commonly used, baseline DRX, D_b, again selected from network operation data. The real-time UE traffic was generated into the network using the App server and UE client's traffic generators. First UE traffic was run against the chosen baseline D_b and then against the on-line DRX suggestion algorithm. The on-line algorithm was run using a time-boxed period of one or ten seconds. After each of these periods a new improved DRX combination may be sent to the UE. Energy usage per UE in the test environment was calculated from the total active time (ON time) as measured in TM500 in multi-UE mode integrated over the total traffic play time. Overall latency per UE was derived from the measured send and arrival time of packets, see Section VI.

Analysis of Set D Lab Results: Set D was characterized by having traffic of very low (UE1-UE3) and very high intensity (UE4). Figure 6 captures the percentage ON time as measured by the TM500. UE1-UE3 show ON activity for less than 10% of the total measurement time. It was still possible to observe a useful reduction in ON time when using our dynamic algorithm. UE4 is particularly interesting. In the baseline case, the device is active 79.77% of the total analysis time. By applying our algorithm this is reduced by almost 20% (79.77-60.13). In [14], Lauridsen estimates that an LTE modem and related components can consume close to 50% of total smartphone power in a full-load scenario. Even using his more conservative estimate of 20% power consumption due to LTE modem components, translates into a significant energy saving. Exact empirical figures vary, from device to device, dependent on battery and electronic circuit implementation decisions. They are not further discussed in this paper as we focus on the on-time reduction metric. In the interest of space we restrict the discussion on latency to set E.

Analysis of Set E Lab Results: Set D illustrates possible extremes in device ON time during recorded traffic. In order to study the effects of DRX configuration in more detail we constructed set E. As discussed in Section VI, set E does not include any long silent periods.

Figure 7 shows the TM500 measured on-time (ON) per UE for one second and ten seconds compared against the

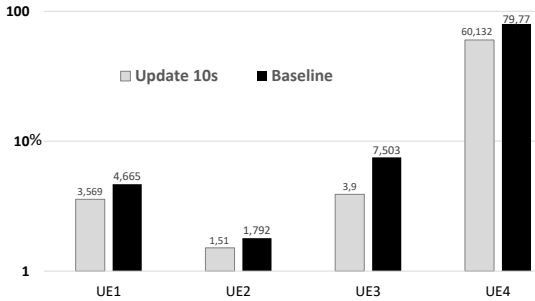

Fig. 6: Total % UE ON Time (Set D)

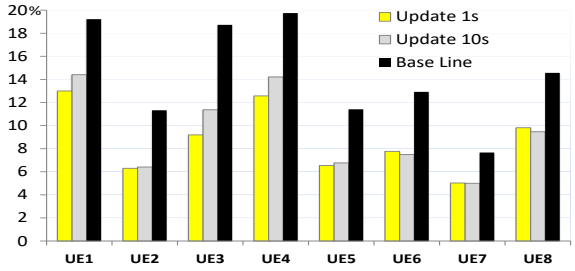

Fig. 7: Total % UE ON Time (Set E)

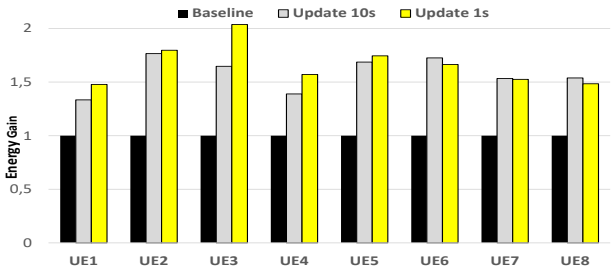

Fig. 8: Total UE Energy Gain Relative to Baseline (Set E)

baseline. Good energy savings in terms of reduced ON time was achieved in comparison to the baseline. We also note that increasing the update frequency shows only a modest gain and in some case is actually worse. There is a signaling cost towards the device associated with a change of DRX configuration, hence using a high update frequency such as one second may, in some cases, be counter productive.

Figure 8 shows the actual energy gain as measured in the lab TM500 infrastructure for update periods of one and ten seconds compared against the measured baseline. A (geometric) mean energy reduction, across all devices, of 1.57 is achieved. This means that using our algorithm device ON time is reduced by a geometric mean factor of 1.57 and thus an energy gain.

Figure 9 shows DL latency measurements for the baseline, one and ten seconds update scenarios. The algorithm attempts to give a better DRX setting without degrading latency as compared to the baseline. The average latency graph shows varied results with the baseline giving marginally better latency in three cases (UE2, UE4, UE5). UE1, UE3 and UE7 show significantly better latency for the average baseline case while UE6 and UE8 show significantly better latency for the system selected ten second update case. In all cases the maximum latency is marginally lower with ten second update while in most cases the standard deviation is significantly lower in the ten second update scenario. This variance in expected average latency illustrates one important limitation with our current approach: in our on-line algorithm we measure packet activity for period **x**, calculate an improved DRX using our algorithm and then apply this to period **x+1**. The results show this approach is effective but not always optimal. Prediction of future traffic patterns is a difficult problem in its own right and we leave a discussion of this to Section VIII. In addition to this the energy model algorithm does not currently have mechanisms to represent the delay due to the RBS scheduler and software stack. All results, however, lie well

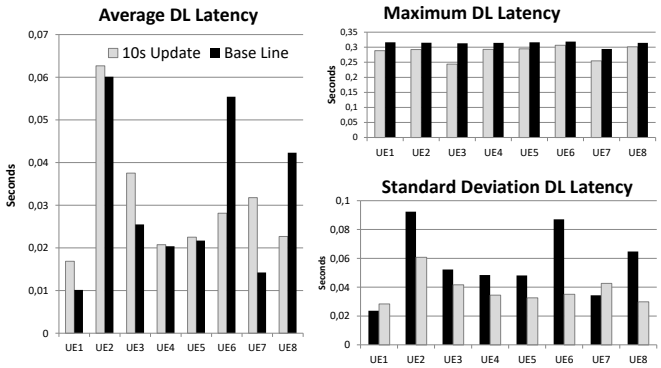

Fig. 9: Total % UE Latency Measurements (Set E)

within tolerance for mobile broadband data services. The DRX mechanism affects latency in downlink because packets will be buffered at the RBS until the next scheduled active period. This is not so with device uplink; uplink service requests (called SR [6] in LTE standards) can be sent at any time transmission resources are need. As such UL latency will not be shaped in a consistent way by the DRX mechanism and our test measurements indicate this is also the case.

We have shown that by dynamically selecting a DRX configuration from a set of 18 based on observed traffic we can show significant energy reduction gain per device. We also observe that while latency varies more than expected using the selection algorithm, the maximum and standard deviation for the analyzed UEs is improved.

B. Evaluation in Off-line Simulation

Executing experiments in the lab environment is time consuming as traffic is run in real-time. Additionally, the test equipment constrains the number of UEs which can run concurrently. To explore a much wider spectrum of DRX settings and their associated potential energy improvements we use an off-line simulation environment. Figure 10 shows the measured and simulated energy gains for traffic set E. The result seems to indicate that our algorithm predicts the real device energy usage well when used off-line. The deviation is primarily due to the same period **x, x+1** effect as described in the previous subsection. We used this result to explore a much larger space of DRX combinations and also to provide a reduced working DRX set recommendation.

In the first off-line simulation we use a random selection of 50 UEs from set B, called Set C, in table II. In this experiment each UE traffic pattern was analyzed as a whole (N=length of UE trace). We use three different groups of DRX settings: 1) **DRX-18**: using the same 18 DRX combinations as in our lab experiments; 2) **DRX-3780**: using a set of 3780 DRX configurations; 3) **DRX-3gpp**: using all combinations from Table I with SCT=$\{1, 2, 3, 4, 5\}$. The groups are proper subsets: DRX-18 $\subset$ DRX-3780 $\subset$ DRX-3gpp. Using a compute cluster and a parallel variant of Algorithm 1 we calculate the most energy efficient DRX, using **best selection** criteria, for each group. This allowed us to isolate energy efficient DRX settings for different UE traces. Figure 11 compares the E value (per UE) of all three groups described previously.

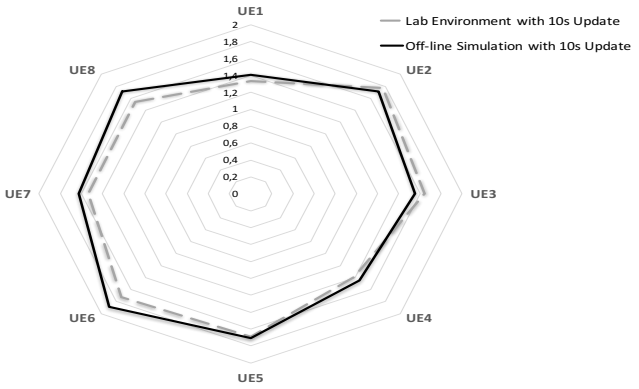

Fig. 10: Predicted Gain: Simulation vs Lab Tests (Set E)

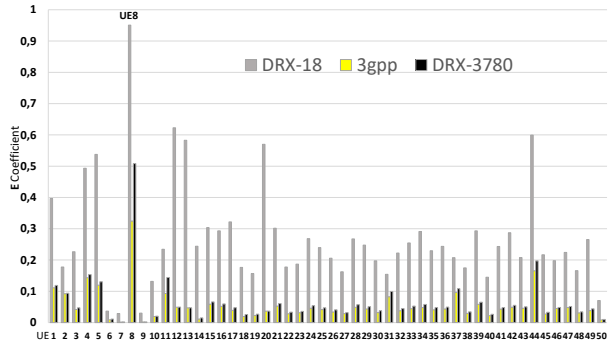

Fig. 11: Energy Reduction Potential from Wider DRX Space

The result illustrates that there is considerable energy saving potential to be harvested from exploring a wider DRX space versus the DRX-18 space used in the lab experiments. The figure also shows that in most case there is only a modest energy gain in exploring the complete DRX-3gpp space vs the DRX-3780 space. UE8 in Figure 11 is an exception to this where the DRX-18 E value (0.95) is reduced to 0.5 for DRX-3780 and then to 0.32 for DRX-3gpp.

Table III captures the count of unique DRX selection in each group when selected for **best latency**, **best power** and **best selection** (when compared against the baseline as described above). So, for example, when choosing **best selection** for group DRX-3gpp there are 17 unique DRX combinations chosen by our algorithm across the 50 devices. Numbers in parenthesis indicate the number of times the first most popular choice is selected followed by the second most popular choice. Following the example above this would be 28 and 6. The results show a high bias towards a few DRX.

We also applied set A from Table II to our simulator. Set A contained traces from 848 devices which was analyzed in ten second periods (N=10) by the simulator. Silent ten second periods were ignored as the they contain no uplink or downlink traffic and the algorithm will always recommend the same use of DRX short and long cycle. This gave a total of 136566, 10 second samples, across all devices. Table IV summarizes the results. When using the **best selection** criteria we got 1931 unique DRX which each gave the best energy gain for at least one of the analyzed periods. Six, 42 and 207 of the DRX accounted for 50%, 75% and 90% of the selections

TABLE III: Count of DRX Selections (Set C). Each UE trace analyzed as one period

DRX Group	**Best Latency**	**Best Power**	**Best Selection**
DRX-3gpp	18 (17,12)	5 (42,5)	17 (28,6)
DRX-3780	26 (12,5)	3 (45,3)	12 (33,4)
DRX-18	10 (12,10)	2 (47,3)	5 (40,5)

TABLE IV: DRX Selection Count and Accumulated Energy Value (Set A)

DRX Group	**Best Latency**	**Best Power**	**Best Selection**
DRX-3780 (50,75,90 %)	919 (4, 31, 112)	34 (2, 3, 5)	1931 (6, 42, 207)
Baseline E = 23712	E = 26660	E = 2490	E = 5672

respectively. We also show the accumulated E value for each group across all devices and samples. While not all of this will translate into energy saving in a device, due to limitations in circuit design, the result does indicated considerable energy saving potential by using a smarter DRX algorithm.

VIII. Discussion, Conclusions and Future Work

In this paper we present the development and implementation of a simple but effective model to estimate good operational DRX settings for a user device connected to an LTE network. We prove through lab experimentation and simulation that the model works effectively. We show a significant reduction in device modem active on-time using measurements from a lab based LTE network. We describe an off-line simulation environment and show a clear connection between this and our lab measurements. We also demonstrate that the off-line simulator is a very useful tool in building a good working DRX set to be used in the on-line algorithm. Execution of this work was challenging in many respects, key of which are; the development of an efficient real-time algorithm that can be used in simulation and deployment, the construction of a lab environment in which meaningful experimentation can be carried out and finally the collection and use of real-traffic data which can be injected into a lab test system. Our model forms a building block in enabling autonomous and self configuring networks with the objective of reducing total aggregate device energy consumption. The model can also be applied to future 5G mobile systems where it is expected that automation and self optimization will be key characteristics. We use a data-driven architecture where traffic data is collected in real-time and processed by a compute cluster. The cluster is capable of handling many connected RBS nodes.

Future work will explore the reduction of the DRX selection to a classification problem and apply machine learning. We intend to use our model to train a neural network and to produce DRX predictions based on a histogram of packet IAT (Inter-arrival time). We also foresee the opportunity to use our model together with advanced machine learning techniques to achieve an accurate DRX prediction of future traffic patterns.

Acknowledgements: We gratefully acknowledge the help from many colleagues at Ericsson in implementing a lab environment and running tests using live LTE equipment. We also thank the anonymous reviewers for their helpful comments. Corcoran and Schulte are partially funded by the WASP (Wallenberg Autonomous Systems and Software Program) research program in Sweden.

REFERENCES

[1] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (EUTRAN); Overall description; Stage 2," Tech. Rep. ETSI TS 136.300 V12.5.0, Apr. 2015.

[2] M. Sauter, Ed., *From GSM to LTE-Advanced: An Introduction to Mobile Networks and Mobile Broadband*, 2nd ed. Wiley, 2011.

[3] GSMA, "VoLTE Service Description and Implementation Guidelines," GSMA, Tech. Rep., 04 2014.

[4] Ericsson AB, "Ericsson Mobility Report," Tech. Rep. EAB-16:018498 Uen, Revision A, Jan. 2016.

[5] 3GPP, "Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC) protocol specification," Tech. Rep. ETSI TS 136.331 V12.12.0, Jan. 2017.

[6] ——, "Evolved Universal Terrestrial Radio Access (E-UTRA); Medium Access Control (MAC) protocol specification," Tech. Rep. ETSI TS 136.321 V12.5.0, Apr. 2015.

[7] L. Zhou, H. Xu, H. Tian, Y. Gao, L. Du, and L. Chen, "Performance analysis of power saving mechanism with adjustable DRX cycles in 3GPP LTE," in *IEEE Veh. Technol. Conf.*, 2008.

[8] Y. Y. Mihov, K. M. Kassev, and B. P. Tsankov, "Analysis and performance evaluation of the DRX mechanism for power saving in LTE," in *IEEE Conv. Electr. Electron. Eng. Isr.*, 2010, pp. 520–524.

[9] S. Fowler, R. S. Bhamber, and A. Mellouk, "Analysis of adjustable and fixed DRX mechanism for power saving in LTE/LTE-Advanced," in *IEEE International Conference on Communications (ICC)*, June 2012, pp. 1964–1969.

[10] R. M. Karthik and A. Chakrapani, "Practical algorithm for power efficient DRX configuration in next generation mobiles," in *Proc. - IEEE INFOCOM*, 2013, pp. 1106–1114.

[11] T. Kolding, J. Wigard, and L. Dalsgaard, "Balancing power saving and single user experience with discontinuous reception in LTE," in *IEEE Int. Symp. Wirel. Commun. Syst.* IEEE, 2008, pp. 713–717.

[12] S. C. Jha, A. T. Koç, and R. Vannithamby, "Optimization of discontinuous reception (DRX) for mobile internet applications over LTE," in *IEEE Veh. Technol. Conf.*, 2012.

[13] G. Stea and A. Virdis, "A comprehensive simulation analysis of LTE Discontinuous Reception (DRX)," *Comput. Networks*, vol. 73, pp. 22–40, 2014.

[14] M. Lauridsen, P. Mogensen, and L. Noël, "Empirical LTE smartphone power model with DRX operation for system level simulations," in *IEEE Veh. Technol. Conf.*, 2013.

[15] M. Lauridsen, "Studies on mobile terminal energy consumption for LTE and future 5G," Ph.D. dissertation, Aalborg University, Denmark, 2015.

[16] C. C. Tseng, H. C. Wang, F. C. Kuo, K. C. Ting, H. H. Chen, and G. Y. Chen, "Delay and Power Consumption in LTE/LTE-A DRX Mechanism with Mixed Short and Long Cycles," *IEEE Trans. Veh. Technol.*, vol. 65, no. 3, pp. 1721–1734, 2016.

[17] H. Bo, T. Hui, C. Lan, and Z. Jianchi, "DRX-aware scheduling method for delay-sensitive traffic," *IEEE Commun. Lett.*, vol. 14, no. 12, pp. 1113–1115, 2010.

[18] O. Ergul, A. T. Koc, and O. B. Akan, "DRX and QoS-aware Energy-efficient Uplink Scheduling for Long Term Evolution," in *IEEE Glob. Commun. Conf.*, 2013, pp. 4617–4622.

[19] M. Agiwal, M. K. Maheshwari, N. Saxena, and A. Roy, "Directional-DRX for 5G wireless communications," *Electron. Lett.*, vol. 52, no. 21, pp. 1816–1818, oct 2016. [Online]. Available: http://digital-library.theiet.org/content/journals/10.1049/el.2016.2850

[20] 3GPP, "LTE;Evolved Universal Terrestrial Radio Access (E-UTRA); Packet Data Convergence Protocol (PDCP) specification," 3GPP, Tech. Rep., 01 2015.

[21] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy array: A structure for efficient numerical computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, March 2011.

[22] Cohham. (2017) Tm500 datasheets. [Online]. Available: http://ats.aeroflex.com/component/edocman/tm500-data-sheets

[23] J. Armstrong, *Programming Erlang: Software for a Concurrent World*. Pragmatic Bookshelf, 2007.

[24] ntp.org. (2017) Network time protocol. [Online]. Available: http://ntp.org

[25] Wireshark. (2017) Wireshark documentation. [Online]. Available: https://www.wireshark.org/docs/