CLEEN 2013, Las Vegas, USA, September 2 2013 – full paper here

Power-aware allocation of MBSFN subframes using Discontinuous Cell Transmission in LTE systems

Daniele Migliorini(1)(*), Giovanni Stea(2), Marco Caretti(3), Dario Sabella(3)

(1) Lepida S.p.A., Bologna, Italy – daniele.migliorini@lepida.it

(2) Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy – g.stea@iet.unipi.it

(3) Telecom Italia, Turin, Italy – {dario.sabella, marco.caretti}@telecomitalia.it

***Abstract*—In LTE and its evolutions, energy efficiency is a critical aspect, also in view of the dramatic traffic growth foreseen for the next years. Cell Discontinuous Transmission (DTX) techniques can be important tools to achieve the needed efficiency in the networks, and one possibility is to implement the DTX by switching off the eNB at some subframes (MBSFN subframes) and not in others (where reference signals are also transmitted). Switching schedules in LTE are made for larger periods (e.g., 40/80ms or even more). We present an algorithm that i) estimates how many resources will be needed in a period, and ii) decides how many resource blocks to activate in each subframe so as to maximize the power efficiency. We show that the power saving is significant, close to the theoretical minimum at low loads. This comes with no reduction in cell throughput and without impacting the QoS (a tolerable extra delay is added only at low loads).**

***Index Terms*—LTE, DTX, MBSFN, energy efficiency**

I. INTRODUCTION

THE Long Term Evolution (LTE) of the Universal Mobile Telecommunication System (UMTS) [1] is gaining progressive hold as an access network for Internet services, thanks to its foreseen near-ubiquitous coverage and high bandwidth. Transmissions are point-to-multipoint, with a base station (or enhanced Node-B, eNB) scheduling several Resource Blocks (RBs) to User Equipment (UEs), at Time Transmission Intervals (TTIs, or subframes) of 1 ms.

Energy efficiency in LTE networks has attracted significant amount of research in the last few years (see, e.g. [6]). Several solutions have been devised to reduce the eNB power consumption, which represents more than half of the power consumption of a mobile operator [6]. Energy savings techniques work in *space*, *frequency* and *time* dimensions. In space, they mainly consist in switching off antennas for long periods of time (i.e., during off-peak hours), covering up by increasing the radius of neighboring cells. In frequency, they may advocate carrier aggregation, using one power amplifier (PA) for a larger spectrum. In time, they rely on switching off the PA for time intervals when the conditions allow it. This last technique has been explored in the EARTH project [7],[9], where Discontinuous Cell Transmission (DTX) was studied at different levels of time granularity: in particular, the switching at subframe (SF) level can be conveniently combined with the usage of Multicast and Broadcast Single-Frequency Network (MBSFN), a feature originally devised for multicast transmission in LTE [1]. Since only *some* SFs in a superframe can be switched off, and since control signals eat up some capacity in some SFs, the problem exists of how to achieve optimal power efficiency, without sacrificing the system QoS. Moreover, switching off SFs increases the delay, hence the benefits of power saving should also be weighed against delay increases.
Power saving policies will certainly benefit from the enhanced processing capabilities brought about by cloud-based architectures, such as Cloud-RAN (C-RAN, [8]). In the latter, all baseband processing is centralized in a cloud, which connects to many Remote Units via wired high-bandwidth links. On one hand, this reduces the cost of endowing single cells with enhanced algorithms (which run in the cloud, instead of at each eNB). On the other, this enables inter-cell coordination, which is expected to reap further efficiency benefits.

In this paper we present an algorithm that allows significant savings, while keeping the packet delay under the QoS constraints, in an LTE cell. The algorithm forecasts the traffic that the cell will need to serve in the next superframe (e.g., 40ms), computes the amount of resources required to serve that traffic and distributes them among the SFs so as to minimize the power consumption, capitalizing on cell DTX. This algorithm constitutes a first building block for a larger-scale one, which may coordinate clusters of (possibly many) neighboring cells, interleaving switch-off periods so as to minimize inter-cell interference and further improve the power efficiency of a whole network.

The rest of the paper is organized as follows: in Section II we describe MBSFN subframes. Section III describes the system model. Section IV describes our resource allocation framework, whose performance is evaluated in Section V. Conclusions are reported in Section VI.

II. DISCONTINUOUS TRANSMISSION IN LTE

According to current LTE specifications, at most six MBSFN SFs can be configured per radio frame for FDD (five for TDD systems). Moreover, MBSFN SFs have fewer common reference signals (RS) than normal SFs (i.e., the data region is RS-free): for this reason sleep modes within the eNB can also be triggered during data regions of these SFs, thus enabling more significant energy savings through DTX. Some SFs are ineligible to be MBSFN for backward compatibility with Rel-8 UEs. In particular, as shown in Figure 1, SF #0 and #5 are used for SCH (synchronization ch.) and BCH (broadcast ch.) transmission, whereas SF #4 and #9 are used for pag-

* Daniele Migliorini carried out this research while at the University of Pisa.

ing transmission in Rel-8 FDD. Thus, the transmitter of the eNB must be active in those frames. The other SFs (eligible for MBSFN operations) are those where the transmission logic may or may not be activated. MBSFN SFs can be signaled via RRC at intervals of few frames, typically four (i.e., 40 ms).

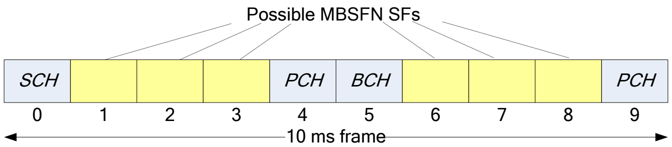

Figure 1: Allowed MBSFN subframes in FDD mode.

III. System Model and Problem Formulation

A downlink scheduler allocates up to N Resource Blocks (RBs) to a set of UEs. *MBSFN superframes* are represented as strings of T bits, $a_0,...,a_{T-1}$ (it is usually $T=40$), which repeat in a pattern unless changed. The eNB activates/deactivates the *transmission* logic at instants i when $a_i=1/a_j=0$ respectively, whereas the *reception* logic is kept always on, independently of the transmission logic. At multiples of T, a new string can be enforced and signaled to the UEs. Some of the a_i s are stuck to 1 (e.g., corresponding to SFs where the some control channels must be transmitted), hence we call them *pinned*. The other SFs, where a decision must be made to activate them or not, are called *free*.

The power consumed by an eNodeB in transmission at a TTI (taken from [9]) depends on whether it is active, and, if it is, on how many RBs are transmitted. The power consumed in *inactive* SFs is constant and equal to P_{off}. The power consumed in *active* SFs, although depending on the type of SF, is an *affine* function of number of allocated RBs, i.e., $P=P_{base}+\rho\cdot n$, where $P_{base}\geq P_{off}$ is the *baseline* power, and $n\leq N$ is the number of allocated RBs.

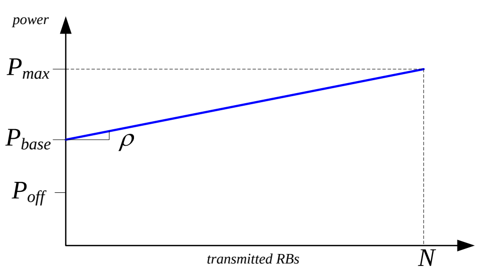

Figure 2: Power model

We assume that in pinned SFs the *baseline* power is higher than in free SFs, since it takes into account the power consumed in transmitting control channels. On the other hand, the *maximum* power in an active frame (when N RBs are allocated) does *not* depend on the presence of control channels, and it is equal to $P_{\max}$. Hence, the *per-RB power consumption rate* ρ depends on the baseline power, i.e., $\rho=\left(P_{\max}-P_{base}\right)/N$. On the other hand, the *exploitable per-RB capacity* depends on the type of SF. In fact, in pinned SFs, control channels eat out some bits that would otherwise be available for UE traffic. We denote with x the exploitable per-RB capacity of an active SF, $0\leq x\leq 1$. For free SFs, it is $x=1$, whereas for pinned SFs it will be $x<1$. Thus, a type i of SF is completely described by the tuple $\{P_{base,i},\rho_i,x_i\}$.

We assume an arbitrary number of types of pinned SFs (thus making the framework more general), and one type of free SFs, which we denote with subscript *f*. Call T_i the number of type-*i* SFs, with $\sum_i T_i=T$.

The problem that we address in this paper is to minimize the overall power consumption over a superframe, i.e. $\sum_{t=0}^{T-1}P(t)$, given the (unknown) arrivals at the eNB. This problem can be addressed at two levels: one is the *scheduling* level, where a more efficient scheduler will use fewer RBs to transmit the same quantity of traffic. The other, complementary one is the *activation* level, which is what we focus on. At the activation level, minimizing the power consumption means deciding which free SFs are active, and how many RBs the scheduler can be made to allocate in each active SF (whether free or pinned). More specifically, given a *traffic demand* of K *fully exploitable* RBs, we show how to allocate them within a superframe so as to minimize the power consumption.

IV. Power-aware Allocation of MBSFN Sub-frames

We first assume that the demand K is known, and then show how to estimate it using quantities that are available at the eNB. Let us lay down some intuitive statements. Being able to *exploit* K RBs is a throughput constraint, hence it is irrelevant *how* pinned/free active SFs are interleaved in a superframe, as long as they cover the required demand (the delay may change by changing the pattern, but the throughput will not). Thus, only the following variables are relevant:

i) how many RBs are allocated for each *type* of pinned SFs;

ii) how many RBs are allocated to *free* SFs, and how many free SFs need to be activated.

It is obvious that, if a total of B RBs is to be shared among free SFs, the minimum-power solution is when $\lceil B/N\rceil$ free SFs are activated, and any solution that shares the B RBs among $\lceil B/N\rceil$ SFs has the same power consumption, since power curves are affine. Call b_i the number of RBs allocated to type-i SFs, a the number of active free SFs. The overall power consumption in a superframe is:

$$\begin{aligned}P&=\sum_{i\neq f}P_{base,i}\cdot T_i+P_{off}\cdot\left(T-a-\sum_{i\neq f}T_i\right)\\&\quad+P_{base,f}\cdot a+\sum_i\rho_i\cdot b_i\\&=\left[\sum_{i\neq f}P_{base,i}\cdot T_i+P_{off}\cdot\left(T-\sum_{i\neq f}T_i\right)\right]\\&\quad+\left(P_{base,f}-P_{off}\right)\cdot a+\sum_i\rho_i\cdot b_i\end{aligned}\tag{1}$$

The addenda between square brackets in (1) are a constant power offset, hence uninteresting from a decision standpoint. Thus, the minimum-power solution is the optimum of the following integer-linear optimization problem:

$$\begin{aligned}&\min\left\{\left(P_{base,f}-P_{off}\right)\cdot a+\sum_i\rho_i\cdot b_i\right\}\\&s.t.\\&\quad b_i\leq T_i\cdot N\qquad i\neq f\qquad(i)\\&\quad b_f\leq a\cdot N\qquad\qquad\quad(ii)\\&\quad a\leq T_f\qquad\qquad\qquad(iii)\\&\quad\sum_i x_i\cdot b_i\geq K\qquad\quad(iv)\\&\quad b_i,a\in\mathbf{Z}_+\qquad\qquad(v)\end{aligned}\tag{2}$$

The function to be minimized is the variable component of the power, and a and b_i are the decision variables. Constraints (*i-iii*) state that the number of RBs made available to the scheduler is limited by the number of type-*i* pinned SFs and active free SFs, respectively. Constraint (*iv*) states that the requested capacity must be fulfilled, taking into account the exploitable per-RB capacity of each type of SF. The cost of the optimal solution is exponential in the number of variables.

We present a linear-complexity heuristic algorithm to solve (2), which gives more insight into the system behavior. The key parameter is the *power-capacity ratio (PCR)* ρ_i / x_i, which measures the power cost per *exploitable RB capacity* in type-i SFs. We proceed to allocate RBs *by increasing PCR*. We therefore sort SF types by PCR, and split pinned SFs in two sets, L and H. The first set includes those types whose PCR is *smaller than (or equal to)* ρ_f / x_f, whereas H includes those whose PCR is *strictly higher* than ρ_f / x_f. The first thing to do is to fill up the SFs in L, in order of increasing PCR. Only afterwards free SFs may be considered for allocation. For these, in fact, activating a single SF also has a *fixed* cost equal to $P_{base,f} - P_{off}$, independent of the number of allocated RBs. That fixed cost has to be weighted in as well when comparing against allocating the same number of RBs to SFs in H. We first show how the algorithm works through a simple example, which however covers all the relevant decisions, and then report the pseudocode.

Example: assume $P_{base,f} - P_{off} = 20$, $T = 40$, $N = 10$, $K = 100$, and three types of pinned SFs, as in Table I:

TABLE I. POWER AND CAPACITY FOR THE SF TYPES

SF type	1	2	3	f
ρ	12	7	11	10
x	.92	.85	.90	1
ρ_i / x_i	13.04	8.23	12.22	10
T_i	4	4	8	24

It is $L = \{2\}$ and $H = \{3,1\}$. Hence, the first step is to allocate all the RBs in type-2 SFs, i.e. $4 \cdot 10 = 40$. This covers a capacity equivalent to $0.85 \cdot 40 = 34$ fully exploitable RBs, hence we still need 66 more. We thus move to considering *free* SFs. Allocating one free SF has a power cost of $P_{base,f} - P_{off} + N \cdot \rho_f = 120$, for an equivalent capacity of 10 RBs. The same equivalent capacity can be obtained using RBs in H if we allocate 12 RBs from type-3 frames (a total of 10.8 units), at a higher cost of $12 \cdot \rho_3 = 132$. Thus, activating one free SF is more efficient. Hence we can safely activate 6 free SFs, thus covering 60 more RBs, and we are left with 6 RBs, which we have to carve from type-3 frames (these having a lower PCR than type-1). To cover 6 *full* RBs worth of capacity, we need $\lceil 6/0.9 \rceil = 7$ more RBs, hence our solution is: $a = 6$, $b_f = 60$, $b_1 = 0$, $b_2 = 40$, $b_3 = 7$. Note that this is also the optimum (computed using CPLEX, [10]). □

With reference to the pseudocode in Figure 3, the types of pinned SFs in L get served first until depletion (lines 2-5). Then, the smallest-PCR SF type are *free* SFs, and we must check two conditions (6):

- if the capacity of *one* free SF is larger than the overall capacity of *all* SFs in H (unlikely as it may be), *or*
- if the power cost of one entire free SF is smaller than the cost of the equivalent capacity using pinned SFs in H, then the lowest-cost solution is to allocate one free SF. The second condition is equivalent to $f(N) \geq 0$, with:

$$f(z) = \sum_{i \in H} \rho_i \cdot \min\left(T_i \cdot z, \left\lceil \left(z - \sum_{j=0}^{i-1} \beta_j \cdot x_j\right) \Big/ x_i \right\rceil\right) - \left(P_{base,f} - P_{off}\right) + \rho_f \cdot z \qquad (3)$$

Furthermore, the above condition holds while there is enough demand to fill one free SF entirely. Hence, we activate $\min\{\lfloor K/N \rfloor, T_f\}$ entire free SFs at once, allocating N RBs to each (7-9).

After all free SFs have been allocated, either $a = T_f$, hence we can only use pinned SFs in H, or $a < T_f$, in which case there still is a free SF to give away, which will however not be a full one (being the remainder of ratio k/N). In this last case (11-15), it remains to be seen if the cost of allocating a *partial* free SF is justified, which requires a similar check as the previous one, this time using $f(k)$ instead of $f(N)$.

After free SFs have been considered (and possibly allocated), there may still be some unsatisfied demand. In order to clear it, we must first allocate RBs to the types in H (16-19). If at the end there is still some residual demand, we need to allocate available free SFs, no matter what the cost (20-23).

```
1.  a=0, bi=0, k=K
2.  for i = 1 to |L| do
3.     bi=min{ceiling(k/xi),Ti*N}
4.     k-=bi*xi
5.     if k<=0 then stop
6.  if sumi∈H{xi*Ti}<N or f(N)>=0 then
7.     a=min{floor(k/N), Tf}
8.     bf=a*N
9.     k-=a*N
10.    if a<Tf then
11.      if sumi∈H{xi*Ti}<k or f(k)>=0 then
12.        bf+=k
13.        a++
14.        k=0
15.        stop
16. for i = 1 to  H  do
17.    bi+=min{ceil(k/xi),Ti*N}
18.    k-=bi*xi
19.    if k<=0 then stop
20. d=min{ceiling(k/N), Tf}
21. a+=d
22. bf+=d*N
23. k-=d*N
```

Figure 3. Pseudo-code for the heuristic

So far we assumed that the traffic demand for a superframe is known and equal to K fully exploitable RBs. There are however three sources of uncertainties. First, arrivals vary over time, because flows come and go *and* their traffic varies from one TTI to another. Second, the UE channel quality varies as well, an effect which is exploited by most schedulers, which strive to serve UEs with the best channel conditions. Third, in a MU-MIMO-capable scheduler, two UEs may be paired and served in MU-MIMO in the same RB (or group thereof), hence the *actual* number of RBs required to carry a given traffic demand depends on whether, and to what extent, MU-MIMO is exploited.

While certainty on the number of RBs required in a future superframe is unattainable, we can make an estimate based on historical records of the relevant quantities, and react – by reducing or increasing the number of RBs/active frames – to changing conditions. At a superframe boundary, we factor in:

- the UE backlog (which counts as unsatisfied demand);
- the amount of allocated RBs in the past superframe;
- the last reported transmission rate of the UEs;
- the MU-MIMO pairings on the allocated RBs;
- the retransmission rate;

and we assume that the past condition will repeat in the next superframe, which makes sense on a cell-wise perspective. We limit our estimate to *the last superframe*, since the further you go back, the less meaningful these data are going to be. The formula for estimating the future demand is the following:

$$K = \left\lceil \left(\frac{1}{1+\mu} \cdot \sum_{j=1}^{N_{UE}} \left\lceil q_j / R_j \right\rceil + \sum_{i \in \{b,s,p,f\}} A_i \cdot x_i \right) \cdot (1+\eta) \right\rceil, \quad (4)$$

where A_i is the number of RBs *allocated by the scheduler* to type-*i* SFs of the past superframe, μ is the percentage of RBs where MU-MIMO pairing was activated, q_i and R_j are the *current* backlog and the most recent rate sample of UE j (we assume a default value if j has never been served), N_{UE} is the number of UEs, and η is the retransmission probability (we assume that the number of double, triple etc. retransmission is negligible). "Allocated by the scheduler" means those that the scheduler actually exploited, possibly fewer than those made available by the provisioning algorithm.

Finally, given a free SFs to be activated, we distribute them as evenly as possible in the superframe, so as to guarantee the maximum responsiveness, and we share the b_i RBs evenly among type-*i* SFs for the same reason.

Some final comments are in order:

- the provisioner is compatible with any scheduler. The only information it needs to know are included in (4). Obviously, a more efficient scheduler will require fewer RBs to accomplish the same job, hence improve the efficiency of the whole framework.
- Since the amount of information flowing between the scheduler and the provisioner is limited, the two may not be co-located. For instance, the scheduler may reside on a small cell, and the provisioner may reside on a central controller (e.g., in C-RAN, [8]) which manages several cells. This paves the way for inter-provisioner coordination, which would allow even greater savings.
- The provisioner is power-model neutral. As long as the power model is an affine function, it just adapts. This means that, as the state of the art on equipment power-efficiency progresses (e.g., yielding smaller P_{off}, P_{base}, $P_{\max}$ and/or ρ values), the algorithm need not be changed, *and* the savings will increase accordingly.

V. Performance Evaluation

The provisioner performance is evaluated via the open-source SimuLTE simulator [3]-[4], based on OMNeT++ [2]. Simulation parameters are reported in Table II. UEs receive Video on Demand (VoD) traffic consisting of a trace from a pre-encoded MPEG4 file (*Futurama* trace [5], medium quality, 0.28 MB/s mean rate), with randomized starting offset. Simulations are run in a 2×2 MU-MIMO-enabled environment. Three downlink schedulers are implemented, namely MaxC/I and Proportional Fair (PF), which always exploit Transmit Diversity (*TxD*), and an enhanced version of MaxC/I (called MaxC/I++). This one *first* selects a transmission mode among TxD, Spatial Multiplexing (*SM*) or MU-MIMO based on the UE reports, so as to maximize the throughput. Then it produces a list of *couples* of UEs that can be paired on the same RB (including UEs that can be paired *with themselves* via SM, and those that cannot be paired, hence use TxD), and sorts that list by increasing overall throughput.

TABLE II. System Parameters

Parameter	Value
Carrier Frequency	2.0 GHz
Total Bandwidth	20 Mhz (10+10Mhz DL/UL)
Duplexing mode	FDD
Channel specifications	ITU URBAN MACRO
Fast Fading Model	Jakes Fading
Number of tap channel	6
MS distance	100 m, 300 m
MS speed	1 km/h ,120 km/h
eNB transmission power	46 dBm
Thermal noise level	-104 dBm
Cable loss	2 dB
Antenna Gain	18 dBi
Noise Figure UE/eNB	7 dB / 5dB
Shadowing std. deviation	8 dB
Mobility Model	Circular
CQI reporting interval	4 TTI
MCS reporting target BLER	0.1
No. of Resource Blocks	50
Max no. HARQ ReTx	4
RLC PDU size	40 Bytes
eNB power model (sleep, base, max)	(150, 260, 448) W

We initially assume a provisioning timescale $T = 80$ ms, and then vary the latter as well. Figure 4 reports the consumed power as a function of the cell load. Without the provisioner, the schedulers offer similar performance, with MU-MIMO-capable MaxC/I++ faring better for obvious reasons. When MaxC/I++ is used jointly with the proposed provisioner, the depleted power decreases by 30% at low loads. Note that the *lower bound* average consumed power is 194W, with the above parameter, corresponding to the case when no RB is ever transmitted. As Figure 4 shows, the average power consumed to support the lowest offered load (one UE) is 199.31W, i.e. quite close to the theoretical minimum. The savings given by the provisioner decrease with the offered load: in fact, in saturation the provisioner has no choice but to activate all SFs. Figure 5 shows the average delay in the same conditions. At low cell loads, the provisioner adds an extra delay of few tens of ms, since backlogged traffic has to wait for sparser active frames. As the load increases, the extra delay goes to zero, coherently with the explanation of the previous figure. Note that with MaxC/I and PF delays grow faster than MaxC/I++, and that adding the provisioner does not alter the comparison and preserves QoS constraints.

We then simulated the MaxC/I++ scheduler and the provisioner varying T. Figure 6 shows that the depleted power is only marginally affected by T (it does depend, instead, on the cell load). Figure 7 reports the average delay at different cell load and different values of T. The average delay increases proportionally with T (unless the cell is in saturation, i.e. at 8Mbps of offered load). This is due to the fact that the traffic prediction becomes less and less effective when T increases, since it relies on channel estimations which are progressively staler and fails to capture the errantly bursty behavior of compressed video. When T is as small as 10ms (and the cell is not saturated), the average delay is similar to the case when the provisioner is disabled. This shows that a faster RRC signaling, allowing for frame-aligned MBSFN provisioning, would yield optimal performance.

VI. Conclusions and Future Work

This paper presented an algorithm that selects which subframes should be set as MBSFN, enabling transmitter switch off periods, and how many RBs should the scheduler be allowed to exploit in each of the other subframes, based on a cell load forecast, so as to cover the forecasted demand at the minimum power. At low loads, the algorithm adds a tolerable amount of delay, which mainly depends on the window size, whereas it is practically transparent at high loads. On the other hand, the savings are close to the theoretical minimum at low loads. The provisioning algorithm works in conjunction with any kind of scheduler, and the amount of information exchanged by the two functional blocks is small enough to allow the two to be implemented in separate physical entities. Thus, the proposed framework lends itself to be employed in novel architectures that may centralize some of the RAN functions (e.g., by considering C-RAN paradigms) of a cluster of cells, notably the provisioning in this case.

Future work can include extending the proposed algorithm to a multi-cell environment, where a central entity schedules switch-offs of MBSFN frames in neighboring cells so as to minimize the inter-cell interference. This will also allow higher SINRs in the active cells all things being equal, thus further improving the power efficiency of the whole system.

VII. Acknowledgements

The subject matter of this paper includes description of results of joint research project carried out by Telecom Italia and the Dipartimento di Ingegneria dell'Informazione of the University of Pisa. Telecom Italia reserves all proprietary rights in any process, procedure, algorithm, article of manufacture, or other result of said project herein described.

References

[1] 3GPP - TS 36.300 Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Overall description; Stage 2.

[2] OMNeT++, http://www.omnetpp.org

[3] SimuLTE webpage. http://www.simulte.com.

[4] G. Stea, A. Virdis, "SimuLTE – A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++", submitted

[5] TU Berlin, MPEG-4 and H.263 video traces for network performance evaluation: http://www2.tkn.tu-berlin.de/research/trace/trace.html

[6] T. Chen, et al. "Network Energy Saving Technologies for Green Wireless Access Networks", IEEE Wireless Communications, Oct. 2011

[7] P. Frenger, *et al.* "Reducing Energy Consumption in LTE with Cell DTX", proc. Vehicular Technology Conference (VTC Spring), 2011

[8] C-RAN – The Road Towards Green RAN, v. 2.5 white paper, China Mobile Research Institute, Oct. 2011

[9] EARTH EU project website, https://www.ict-earth.eu/

[10] ILOG CPLEX Software, http://www.ilog.com

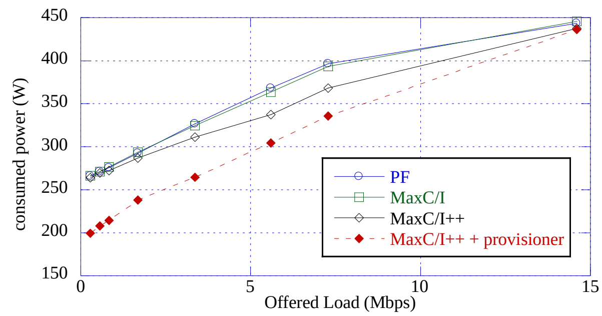

Figure 4: Consumed power as a function of the cell load

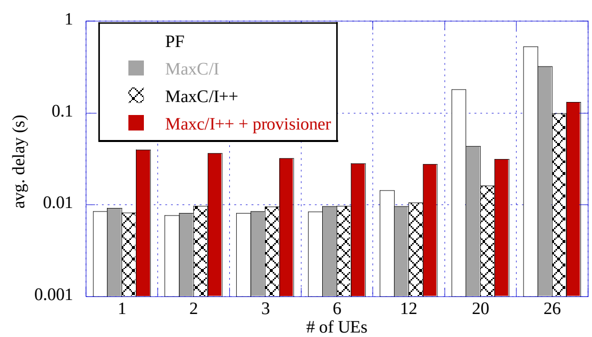

Figure 5: Average delay as a function of the cell load

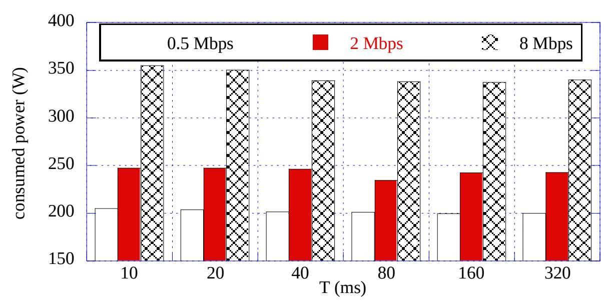

Figure 6: Consumed power as a function of the provisioning timescale

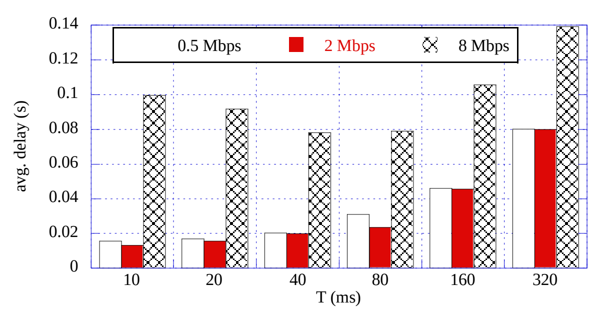

Figure 7: Average delay as a function of the provisioning timescale