This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

Practical large-scale coordinated scheduling in LTE-Advanced networks[1]

Giovanni Nardini(1), Giovanni Stea(1), Antonio Virdis(1), Dario Sabella(2), Marco Caretti(2)

(1) Dipartimento di Ingegneria dell'Informazione, University of Pisa, Italy
g.nardini@ing.unipi.it, giovanni.stea@unipi.it, a.virdis@iet.unipi.it

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

***Abstract*—In LTE-Advanced, the same spectrum can be re-used in neighboring cells, hence coordinated scheduling is employed to improve the overall network performance (cell throughput, fairness, and energy efficiency) by reducing inter-cell interference. In this paper, we advocate that large-scale coordination can be obtained through a layered solution: a *cluster* of few (i.e., three) cells is coordinated at the first level, and clusters of coordinated cells are then coordinated at a larger scale (e.g., tens of cells). We model both small-scale coordination and large-scale coordination as optimization problems, show that solving them at optimality is prohibitive, and propose two efficient heuristics that achieve good results, and yet are simple enough to be run at every Transmission Time Interval (TTI). Detailed packet-level simulations show that our layered approach outperforms the existing ones, both static and dynamic.**

***Index Terms*—LTE-A, Coordinated Scheduling, CoMP, Optimization**

1. Introduction

THE ever-increasing trend towards higher user bandwidth in LTE-Advanced (LTE-A) cellular networks [1] finds a natural opponent in inter-cell interference. Coordinating neighboring cells, so as to reduce the interference suffered by each User Equipment (UE, e.g. a mobile phone), is also the key to achieve higher Signal-to-Interference-and-Noise Ratios (SINRs), hence higher throughput, energy efficiency for the same throughput, and fairness for cell-edge UEs. Coordinated Scheduling (CS) is a CoMP (Coordinated Multi-Point Transmission and Reception) technique that allows several eNodeBs (eNBs) to coordinate service to a set of UEs: by deciding who addresses whom and using which Resource Blocks (RB), pairs of cell-UEs transmissions can be scheduled concurrently with a tolerable increase in interference, thus maximizing the benefits of spatial spectrum reuse [2],[3].

Cells can be coordinated in both a distributed and a centralized architecture. The former relies on eNBs running independent algorithms and sharing information through peer-to-peer inter-eNB connections. This approach may suffer from limited state visibility (i.e., each eNB only possesses partial information on the state of the network, and especially of neighboring cells, hence makes suboptimal decisions) and may entail higher inter-eNB communication latencies. Centralized coordination, instead, can leverage cloud-based architectures, such as Cloud Radio Access Network (C-RAN) [4]. This makes it possible

[1] Some of the material included in the present paper also appeared, in a preliminary form, in [24] and [25].

to exploit network-wide information to make better decisions, provided that the computational overhead does not become a bottleneck itself.

Coordinating a (possibly large) number of cells entails deciding which cells are active on which RB, to target which UE, so as to minimize the interference and increase the overall throughput. In order to do this, the system needs to be able to assess the effect of interference of subsets of cells on single UEs. The main problem with this approach is *scale*: UE channel reporting is limited in practice, and an eNB can only be expected to be made aware of the interference of but a few (e.g., one or two) neighboring cells by each UE **Errore. L'origine riferimento non è stata trovata.**,**Errore. L'origine riferimento non è stata trovata.**[2]. Even though increasing the coordination scale is likely to yield diminishing returns in the long run, the scale at which coordinated scheduling is beneficial goes beyond these figures. Therefore, in this paper we advocate a *layered* approach: we decompose the problem into *small-scale* and *large-scale* coordination (SSC and LSC, respectively): we first endeavor to coordinate a relatively small *cluster* of three neighboring cells, using a *level*-1 *master node* that arbitrates the provisional schedules of the coordinated cells. Then, we scale up by coordinating *clusters* through a level-2 master node, which capitalizes on the underlying SSC work. We formulate both the SSC and the LSC problems as optimization problems and discuss their complexity, showing that solving each of them at optimality is impractical. Then we propose fast, yet effective heuristics for both problems which can be run at short timescales.

The strengths of layered coordination are several: first, it improves the performance of the network as for throughput, Quality of Service (QoS), fairness and energy efficiency, as we show using detailed multi-cell packet-level simulations. The improvements are *progressive*: SSC alone brings significant benefits (notably, a remarkable increase in cell throughput). Adding LSC further improves the performance, particularly in terms of fairness, QoS and energy efficiency. Second, thanks to the efficiency of our heuristics, layered coordination can be run *dynamically*, and at fast timescales, possibly at each TTI, thus reaping the benefits of fresh channel quality information (CQI) and better coping with bursty and/or intermittent traffic sources (e.g., video). Third, it is *flexible*: it can be implemented in both a

[2] One method to deal with interference measurement is reported in [31], Chapter 15.2: a set of neighboring cells can be configured to transmit either a non-zero- or a zero-power Reference Signal (RS), hence one can measure the interference with/without transmission from that set of cells. RSs are transmitted using Resource Elements in the Physical Downlink Shared Channel. As more cells are added to the set, more RSs are required, which increases the overhead.

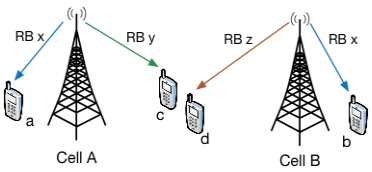

Figure 1 – Coordinated Scheduling.

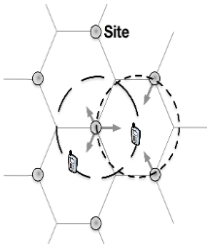

Figure 2 – Clustering architecture.

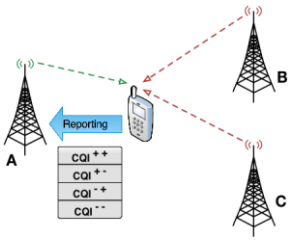

Figure 3 – CQI reporting.

centralized architecture, such as C-RAN, and a distributed-RAN one, and may accommodate any eNB scheduler, e.g., Maximum Carrier over Interference (Max C/I), Proportional Fair (PF), etc..

CS-CoMP has attracted some research lately (see, e.g., [7] and the references therein). *Static* approaches (e.g. [12],[13]) have been proposed: each cell has a statically reserved subset of RBs, where it transmits only exclusively or together with low-power interferers. Among the *dynamic* approaches [8] bears some apparent similarity to ours, in that it requires a central controller which arbitrates provisional schedules made by the cells. However, it performs considerably worse in practice, because the controller – by arbitrating single RBs – fails to find work-arounds to conflicting requests by the eNBs and is thus prone to long-term unfairness.

The rest of the paper is organized as follows: Section 2 reports background on LTE-Advanced. We describe the system model in Section 3. Our layered approach is explained in Section 4. Section 5 reviews the related work. In Section 6, we evaluate our framework and compare it to the existing ones. Finally, Section 7 concludes the paper.

2. Background on LTE-Advanced

In this section we describe those features of the LTE-A system that are more relevant to the problem at hand, i.e. downlink scheduling at the MAC layer.

In an LTE-A network, resource allocation takes place at the level of *cells*. Cells are implemented at an eNodeB (eNB), which may be physically realized either as a compact entity, possessing the intelligence to compose cell transmission schedules (called *subframes*) at every TTI, or in a split architecture, with a Remote Radio Head (RRH) connected to a baseband (BB) unit. In the latter case, BB resources of several cells can be pooled in a centralized entity, as in a C-RAN architecture. Since our problems and solutions can be mapped on both frameworks via straightforward architectural modifications, we henceforth make reference to the first deployment for the sake of consistency and readability. The radiation pattern of a cell may or may not be isotropic. In this last case, cells are usually co-located.

Scheduling takes place every Transmission Time Intervals, (TTIs), whose duration is 1ms, and consists

in allocating a vector of (*Virtual*) *Resource Blocks* (RBs) to UEs (one RB goes to one UE only[3]). Each RB carries a different amount of bits depending on the Channel Quality Indicator (CQI) reported by the UE it is addressed to. The CQI increases with the measured Signal to Noise and Interference Ratio (SINR), and it can be either *wideband*, i.e. covering the entire subframe, or *frequency-selective*. In the latter case, a number of per-subband CQI are reported by a UE. However, when assembling a Transmission Block (TB) in a TTI, the eNB maps it on the relevant number of RBs and chooses *one* Modulation and Coding Scheme (MCS), typically, the one corresponding to the minimum CQI reported on the allocated RBs.

A single UE is associated to a *cell*, whose signal it receives and decodes[4]. Transmissions of neighboring cells on the same RBs count as interference, which can be mitigated through *coordinated scheduling* (CS), a CoMP (Coordinated Multi-Point Transmission and Reception) technique [2]. CS can be exemplified with reference to Figure 1: cells A and B can target UEs *a* and *b* on the same RB *x*, since the interference that each will perceive from the neighboring cell will be small, whereas they should use different RBs, e.g., *y* and *z*, to target UEs *c* and *d*, and refrain to transmit on *z* and *y*, respectively, to avoid excessive interference. Interference-prone transmissions imply lower SINR, hence more RBs are required to transmit the same payload. On one hand, this obviously reduces the capacity of the network, allowing fewer UEs to be served simultaneously. On the other, it negatively affects the energy efficiency, which also depends on the number of bits per RBs.

3. System Model

This section details the hypotheses and goals of this work.

For ease of representation, we picture the network as a tessellation of hexagons, as in Figure 2. Each hexagon represents an area covered by three overlapping cells. We assume that cells are anisotropic, radiating at 120° angles, hence each second vertex of a hexagon hosts three co-located cells. A number of UEs is deployed in the area: each of them is associated to one cell, and it reports wideband CQIs to it. However, the serving cell is made aware of the level of interference received by each UE from *two other cells*, This information is stored by the cell scheduler in the form of *four* different CQIs, corresponding to the case when either or both the two interferers are *muted*.

[3] Multi-user Multiple-Input/Multiple-Output (MIMO) techniques are outside the scope of this paper.

[4] We leave out techniques such as *joint processing*, whereby two cells target the same UE simultaneously, reinforcing the useful signal.

Furthermore, we assume that the network can be configured so that cells can be *clustered by three*, and all the UEs associated to a cell report the interference from the other two cells in the same cluster. Two ways to cluster cells, shown in Figure 2, are considered:

- *intra-site clustering*: the three co-located cells at a vertex form a cluster;
- *inter-site clustering*: the three cells radiating in the same hexagon form a cluster.

Clustering will be used as a basis for SSC. The notation, models and algorithms reported in the rest of the paper are independent of how we cluster cells, although the resulting performance will of course vary, as we show in Section 6. For the sake of concreteness, but without any loss of generality, we will refer to *intra-site* clustering hereafter.

We denote with A, B, C the three cells in a cluster, each one of which can allocate M RBs. To make notation consistent, if *x* denotes a generic cell, then $x+1$ and $x-1$, denote the next and the previous ones in the above order (with wrap-around, i.e., $x=A \Rightarrow (x+1)=B$, $(x-1)=C$). Let $N(x)$ be the number of UEs associated to cell *x*. UEs can thus be identified by couple (*x*,*j*), where $1 \le j \le N(x)$.

Consider UE *j* associated to cell x. Its SINR, hence its CQI, will be different based on whether cells $x+1$ and $x-1$ are active (thus increasing the interference) or not. This allows us to define *four Interference Logical Subbands (ILSs)*, corresponding to the four combinations of activity of $x+1$ and $x-1$, for that UE, and four different per-ILS CQIs accordingly (see Figure 3). We thus use two superscript symbols to denote the interference from the other two cells. The first symbol identifies cell $x-1$, whereas the second is for cell $x+1$. Symbol "+" means "active", and "$-$" means "inactive". This way, $CQI_{x,j}^{t}$, where $t \in T = \{++,-+,+-,--\}$, denote the four possible CQIs for a UE *j* associated to cell *x*: $CQI_{x,j}^{++}$ is the one achievable when both *x*-1 and *x*+1 are active, etc. Set *T* thus represents the four ILSs for a UE: "$++$" denotes the *no-muting* ILS, "$--$" is the *double-muting* one, and "$-+, +-$" are the *single-muting* ones. We use the name "subband", which is suggestive of multi-band CQIs, to exploit the inherent parallel between coordinated scheduling, on one hand, and multi-band scheduling at a single cell, on the other: in fact, in both cases, a UE may be served based on one of relatively few CQIs, depending on the (interference logical) subband(s) it is assigned to, hence scheduling decisions must take this into account. However, we recall that, in multi-band scheduling, subbands have fixed size, whereas in CS the size of ILSs must be decided.

Our goal is to coordinate a relatively high number of cells, those radiating in a cluster of up to seven

hexagons (i.e., 21 cells), so that a network-wide measure (e.g., the overall throughput) is maximized.

4. LAYERED APPROACH TO COORDINATION

The *logical* (i.e., functional) layout of our coordination scheme is shown in Figure 4: a *level*-1 *master* (L1M) coordinates a cluster of three cells, thus embodying SSC. L1Ms are further coordinated by a *level*-2 master (L2M), to achieve LSC. The job of SSC is to decide which subset of cells transmits on a given RB, with the aim of optimizing a cluster-wide measure (e.g., the overall throughput), given the CQIs of the associated UEs. During SSC, the L1M will compute the *size of the ILSs* for each cell in the cluster. The purpose of LSC is instead to *arrange* ILSs of neighboring clusters so as to minimize the overall interference, given that the ILS sizes have already been set in the previous phase. The output of the LSC is a set of associations {RB-ID, ILS-ID}, computed in such a way that the overall interference is reduced. Note that it is perfectly possible to run SSC only, and still reap some of the benefits of coordinated scheduling: in doing so, each cluster will arrange its ILSs autonomously, hence their placement will be suboptimal due to the absence of LSC (interference will not be minimized, although it will be considerably less than *without* SSC). In the following, we present SSC and LSC in order: we formalize both as optimization problems, show why solving them at optimality is infeasible, and devise fast heuristics to solve them.

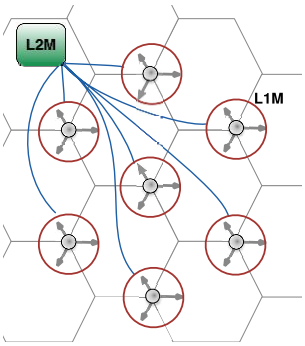

Figure 4 – Layered coordination.

a. Small-scale coordination

Small-scale coordination coordinates $K=3$ cells. Let $s_{x,j}^{t}$ be the number of RBs allocated to UE *x*,*j* within ILS *t*. Let $Q_{x,j}$ be that UE's backlog and let $r_{x,j}$ denote the number of bits per RB according to the (one and only) TB format that UE *x*,*j* will be scheduled with. Let us denote with $\pi(c)$ the number

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

of bits per RB achievable under CQI c. We denote with $b^t_{x,j}$ a binary variable that is set when UE *x,j* has a RB within ILS *t*. Finally, let R be a constant such that $R \geq \pi(CQI_{\max})$. A *cluster-wise max-throughput* problem can then be formulated as follows:

$$
\begin{aligned}
&\max \sum_{x\in\{A,B,C\}} \sum_{j=1}^{N(x)} r_{x,j} \cdot \left(\sum_{t\in T} s^t_{x,j} - p_{x,j} \right) \\
&s.t. \\
&r_{x,j} \cdot \sum_{t\in T} s^t_{x,j} \leq Q_{x,j} + p_{x,j} && \forall x, j && (i) \\
&r_{x,j} \leq \pi\left(CQI^t_{x,j}\right) + R \cdot \left(1 - b^t_{x,j}\right) && \forall x, j, t && (ii) \\
&b^t_{x,j} \leq s^t_{x,j} \leq M \cdot b^t_{x,j} && \forall x, j, t && (iii) \\
&p_{x,j} \leq \pi\left(CQI^t_{x,j}\right) - 1 + R \cdot \left(1 - b^t_{x,j}\right) && \forall x, j, t && (iv) \\
&\sum_{t\in T} \sum_{j=1}^{N(x)} s^t_{x,j} + \sum_{j=1}^{N(x+1)} s^{--}_{x+1,j} + \sum_{j=1}^{N(x-1)} s^{--}_{x-1,j} \\
&\quad + \max\left\{ \sum_{j=1}^{N(x+1)} s^{+-}_{x+1,j}, \sum_{j=1}^{N(x-1)} s^{-+}_{x-1,j} \right\} \leq M && \forall x && (v) \\
&\sum_{x} \left(\sum_{j=1}^{N(x)} s^{--}_{x,j} + \max\left\{ \sum_{j=1}^{N(x+1)} s^{+-}_{x+1,j}, \sum_{j=1}^{N(x-1)} s^{-+}_{x-1,j} \right\} \right) \\
&\quad + \max\left\{ \sum_{j=1}^{N(x-1)} s^{++}_{x-1,j}, \sum_{j=1}^{N(x)} s^{++}_{x,j}, \sum_{j=1}^{N(x+1)} s^{++}_{x+1,j} \right\} \leq M && && (vi) \\
&b^t_{x,j} \in \{0,1\}, \quad s^t_{x,j} \in \mathbb{Z}^+ && \forall x, j, t && (vii) \\
&r_{x,j}, p_{x,j} \in \mathbb{Z}^+ && \forall x, j && (viii)
\end{aligned}
\tag{1}
$$

The objective function states that the cluster throughput should be maximized. Note that other, alternative objectives can be easily substituted to this one in order to realize different CS-CoMP strategies. We will come back to this later on. Every UE *x,j* has a unique rate $r_{x,j}$, which is multiplied by all the RBs that are allocated to that UE, whatever the ILS they belong to. $p_{x,j}$ denotes the padding, not to be counted as useful bits.

Constraint (*i*) states that each UE cannot transmit more than its backlog's worth of traffic, including possible padding bits. Padding is necessary, because the number of RBs is an integer, and queues may never be emptied otherwise. Constraint (*ii*) states that the rate cannot exceed the minimum number of bits per RB among all the ILSs it is scheduled in. For instance, if a UE is allocated RBs with no interference ($b^{--}_{x,j} = 1$) and with interference from both cells ($b^{++}_{x,j} = 1$), it will use the smallest number of bits per RB, i.e. $r_{x,j} = \pi\left(CQI^{++}_{x,j}\right)$. Note that R is a large constant, hence constraint (*ii*) is inactive if $b^t_{x,j} = 0$ (meaning that ILS *t* does not contribute to the limit). Constraint (*iii*) states that $s^t_{x,j} = 0$ if $b^t_{x,j} = 0$, and $s^t_{x,j} \geq 1$ if $b^t_{x,j} = 1$, thus ensuring consistency. Constraint (*iv*) states that a UE always gets

less than one RB's worth of padding. Constraint (*v*) states that a subframe must include the RBs that a cell *x* allocates to its UEs *x*,*j*, whichever their ILSs *t* (i.e., those in the first double sum). However, cell *x* has to leave enough room in its frame to allow *other* cells $x+1$ and $x-1$ to allocate RBs without interference from cell *x*. Such room is in fact accounted for in the other three addenda, which can be further split into two: first, the ILSs where the other cells require exclusive transmission (i.e., those with a $--$ superscript). Second, the ILSs where other cells require *only x* to be muted (i.e., those in the *max* bracket). These last need not be disjoint. Figure 5 shows an example of coordinated subframe structure for three cells A, B, C, over which constraint (*v*) can be exemplified. Cell A's subframe (the leftmost one) must have room for all the RBs where:

- A transmits to its UEs: first addendum in (*v*), corresponding to ILSs 1, 4, 6, 7 in Figure 5;
- B requests *both* A *and* C not to transmit: second addendum in (*v*), ILS 2 in the figure;
- C requests *both* A *and* B not to transmit: third addendum in (*v*), ILS 3 in the figure;
- B requests A not to transmit, whereas C may transmit: first element of the *max* bracket in (*v*), ILS 5 in the figure;
- C requests A not to transmit, whereas B may transmit: second element of the *max* bracket in (*v*), again ILS 5 in the figure;

The last two terms can overlap, thus we take their maximum instead of their sum. Note that inequality (*v*) is verified with some slack at cell A, i.e. there are some unused RBs (bottom parts of ILSs 4 and 6). We will come back to this later on. Constraint (*vi*) describes the fact that the clusters of RBs where muting of one or two cells is required must occupy the same positions in the subframes of the three cells. Finally, constraints (*vii-viii*) define the domain of the problem variables.

The above problem is a mixed integer-nonlinear problem (MINLP), with a size of $O\left(K \cdot N \cdot |T|\right) = O\left(K \cdot N \cdot 2^K\right)$ variables and constraints, N being the overall number of UEs. Non-linearity comes from the product in both the objective function and constraint (*i*), whereas the *max* operator in constraints (*v-vi*) could easily be linearized[5]. MINLPs are NP-hard in general. As anticipated, the structure of this one is indeed similar to that of a multi-band-CQI scheduling (i.e. one where a MaxC/I allocation has to be made on per-subband CQIs), subbands being replaced by ILSs, with the added complication that the dimension of ILSs is not known in advance, but is obtained as a result, as

[5] This problem *can* be reformulated as a mixed-integer-*linear* problem (MILP), through a careful reformulation (omitted for the sake of conciseness), but only at the price of increasing the number of variables to $O\left(2 \wedge \left(2^K\right)\right)$.

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

per constraints (*v*-*vi*). Since the multi-band-CQI scheduling problem has been proven to be NP-hard in [28], this one can only be NP-hard as well. In any case, solving it in a TTI's time is out of question, even for a small number of UEs, i.e., 10-20), as shown in [24]. Furthermore, we observe that the reporting information required is proportional to the number of ILSs, which increases exponentially with the number of coordinated cells K. This clearly indicates that clustering cells at larger scales is impractical, and this is why our SSC scheme coordinates three cells only.

The solution to the SSC problem yields a set of $s^t_{x,j}$ values. From the latter, the size of each ILS *d* of a cell, call it Δ_d, can be easily obtained. However, ILSs can be arranged in several ways, provided that mutual exclusion constraints are met, without affecting optimality. For instance, the first three (double muting) ILSs in Figure 5 could be permuted. This degree of freedom will in fact be exploited later on to achieve larger-scale coordination.

As anticipated, we observe that the above problem formulation easily accommodates different objectives. For instance, a Coordinated Proportional Fair (CPF) could be achieved by simply substituting the objective with:

$$\max \sum_{x\in\{A,B,C\}} \sum_{j=1}^{N(x)} r_{x,j} \cdot \left(\sum_{t\in T} s^t_{x,j} - p_{x,j}\right) \Big/ \Phi_{x,j} \,,$$

where $\Phi_{x,j}$ is the long-term PF rate achieved by UE (*x*,*j*), which is available at each cell. Similarly, any other scheduling strategy that weighs UEs according to some constant (e.g., urgency-based, queue length-based, etc.) can be accommodated in the same way.

Figure 6 shows the information flow for optimal SSC. The cells play a very minor role, since they only juxtapose queue information (and, possibly, long-term PF rates or similar information) to the CQIs reported by the UEs. Moreover, the L1M composes schedules itself, hence each cell only has to place UE data within it.

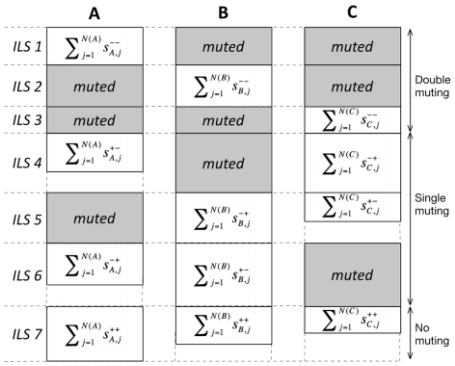

Figure 5 – Subframe structure and ILSs for three coordinated cells.

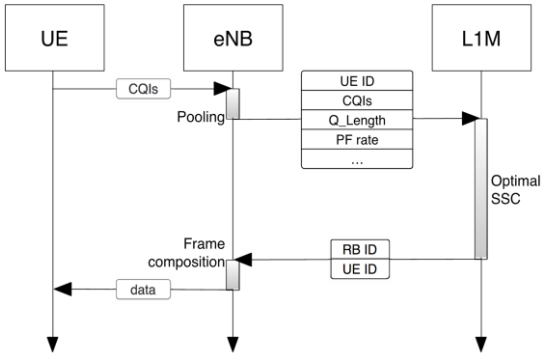

Figure 6 – Optimal SSC.

Heuristic solution for small-scale coordination

The key observation for our heuristic is that the job of the L1M can be made considerably easier (hence faster) by having the cells *pre-process* information first, and *participate in the scheduling* later on. As said before, the SSC problem presents similarities with the multi-band allocation problem, whose main difficulty is determining the size of each ILS based on the traffic demand.

At a high level, our SSC heuristic can be split in three steps. First, cells make a *provisional resource allocation*, deciding which UEs should be served in which ILS. Each cell then communicates its requirements to the L1M. By doing this, it makes a *bid* on how large each ILS should be to meet its needs. Second, the L1M computes the *actual* size of each ILS, by composing the cell bids, and curbing requests that cannot be accommodated. Then, it sends the results back to the cells in its cluster. As a third and last step, cells perform the *actual resource allocation*, in a subframe where the position and size of the ILSs are consistent for the whole cluster. We now explain each step in more detail.

Step 1: For each UE *j* under its control, cell *x* computes $\gamma_{x,j}^{t} = \pi\left(CQI_{x,j}^{t}\right) / \pi\left(CQI_{x,j}^{++}\right)$. Values $\gamma_{x,j}^{t}$ indicate the throughput gain per RB of UE *j*, with respect to the *no-muting* ILS, when that UE is scheduled in ILS *t*. We partition UEs into four sets Γ_x^t. Each set Γ_x^t groups UEs that should be served in ILS *t*. The association is made by testing the following rules in cascade:

- If $\gamma_{x,j}^{--} \geq th^{DM}$, add *j* to Γ_x^{--}; otherwise,
- If $\max\left\{\gamma_{x,j}^{-+}, \gamma_{x,j}^{+-}\right\} \geq th^{SM}$, add *j* to the set with the higher gain (i.e., Γ_x^{+-} if $\gamma_{x,j}^{+-} \geq \gamma_{x,j}^{-+}$); otherwise,
- Add *j* to Γ_x^{++}.

th^{DM} and th^{SM} are the *double-muting* and *single-muting* thresholds. This way, muting of neighboring cells is requested only when significant gains can be obtained. At this point, the cell performs a *provisional* schedule, according to its own policy (e.g., MaxC/I, PF etc.). The CQI used for this procedure depends on which Γ_x^t the UE has been assigned to. The provisional schedule provides that cells' bid for each ILS *t*, N_x^t, which is sent to the L1M.

Step 2: In this step, the L1M sets the size and position of the ILSs. To do so, first it composes all the bids of the coordinated cells, which may not be simultaneously feasible. Then, if there is room to do so, it reduces the level of interference of the RBs, e.g. moves them from the no-muting ILS to the single-muting and double-muting ILSs as much as possible. The exact algorithms for composing the bids and

upgrading the RBs are intuitively simple, but cumbersome to describe formally. For this reason, we provide an intuitive description here, using examples when required, and refer the interested reader to the Appendix for an algorithmic description.

Since the bids are made independently by the cells in the cluster, the L1M must ensure that they are mutually compatible, by checking the following inequalities – homologous to (*v*-*vi*) in (1):

$$\begin{aligned} &\sum_{t\in T} N_x^t + \sum_{\substack{y,\\ y\neq x}} N_y^{--} + \max\left\{N_{x-1}^{-+}, N_{x+1}^{+-}\right\} \le M \quad \forall x \quad (i) \\ &\sum_x \left(N_x^{--} + \max\left\{N_{x-1}^{-+}, N_{x+1}^{+-}\right\}\right) + \\ &\qquad + \max\left\{N_{x-1}^{++}, N_x^{++}, N_{x+1}^{++}\right\} \le M \qquad (ii) \end{aligned} \tag{2}$$

Note that inequalities (*i*) are one per cell. If (2) holds, then all the bids are feasible, and the L1M can partition the subframe into ILSs. Otherwise, some of them must be reduced. The first part then consists in decreasing by one all the bids of the violated inequalities in a round-robin fashion, until all inequalities are made to hold again.

Once the bids have been composed, we can optimize the ILSs, by "upgrading" RBs to more protected ILSs whenever possible. Assume that the situation is the one depicted in Figure 7: in ILS 4, i.e. the one where B only is muted, it is $N_C^{-+} > N_A^{+-}$, thus there are $N_C^{-+} - N_A^{+-}$ RBs where C would transmit *alone* in any case. This means that these RBs (shown as shaded in the figure) in fact belong to C's double-muting ILS, to which they must be added. The same happens for the $N_B^{-+} - N_C^{+-}$ RBs in ILS 5 and the $N_B^{+-} - N_A^{-+}$ RBs of ILS 6, both to be added to cell B's double-muting ILS. Finally, with reference to the *no-muting* ILS 7, where N_A^{++} is equal to the width of the ILS, there are $N_A^{++} - N_B^{++}$ RBs where A is transmitting alone, to be moved to its double-muting ILS, and $N_B^{++} - N_C^{++}$ RBs where A and B are in fact in single-muting, since C is not using them, and therefore they should be moved to ILS 6 instead. Figure 8 shows the final partitioning of the subframe for all the three cells, obtained by moving RBs as described above. Note that the double-muting ILSs end up being larger, and the no-muting ILS is considerably smaller.

As a final step, the allocation can be further enhanced if there are still empty RBs: for example, we can rearrange the same *no-muting* RB so as to obtain three *double-muting* RBs, one per cell, as shown in Figure 9. By optimizing the subframe partitioning as specified above, the number of RBs allocated to each cell stays the same: rather, we modify their interference conditions, allowing higher CQIs to be exploited. Finally, the L1M builds a list of tuples {RB_ID, ILS_ID} and sends it to the cells.

Step 3: This is where cells perform the actual scheduling. As ILSs have a well-defined size, any multi-band scheduling algorithm can be adapted to this purpose. As already stated, solving *optimally* the multi-band version of the most common scheduling algorithms (e.g., MaxC/I and PF) is NP-hard in general (see [28]). We therefore use a commonplace heuristic, which consists in filling up one ILS at a time, starting from the double-muting one. Considering ILS *t*, the cell assigns N_x^t RBs to UEs that were inserted in Γ_x^t at step 1, using the *same* algorithm as in step 1. If some RBs remain unallocated, then we "upgrade" UEs from one of the less protected sets (i.e., corresponding to ILSs in which more cells transmit simultaneously). UEs are upgraded in order of decreasing gain $\gamma_{x,j}^t$. Vice-versa, if not all UEs belonging to Γ_x^t can be served in ILS *t* (e.g., because its width was reduced during step 2), we move the remaining ones to the set of a less protected ILS, i.e., the one where they have the highest CQI. Once a ISL is full, the next one is considered.

The complexity of the SSC heuristic is affordable: cells are required to do no more than their scheduling job, and the task of L1M is computationally trivial. Moreover, the information being exchanged between the cells and the L1M (shown in Figure 10) is limited, and *independent* of the number of UEs or the traffic load. For this reason, SSC can be run *dynamically, at a TTI timescale*.

Moreover, the heuristic SSC can accommodate different (sub-band aware) scheduling algorithms at the cell, e.g., Proportional Fair, Max C/I, time-based priority, etc. It is also worth noting that the SSC heuristic allows a last word (i.e., step 3) to the cells, which is done on purpose to achieve scheduling consistency: suppose, in fact, that the PF criterion would select UE *x,j* to be scheduled in the double muting ILS, since it is just above the th^{DM} threshold. If the L1M reduces the double-muting ILS for *x*, we do want to allow cell *x* to choose whether to schedule *j* in some other ILS (e.g., either of the single-muting ones) rather than having to drop it altogether. Similarly, if the double-muting ILS is larger than expected, we still want *x* to be in control of which additional UEs will be promoted to it (e.g., starting from those nearer to the th^{DM} threshold).

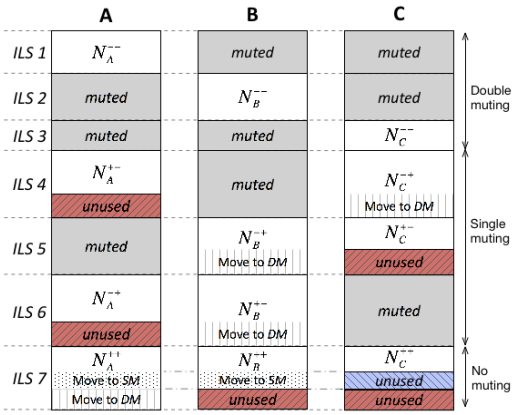

Figure 7 – Frame composition.

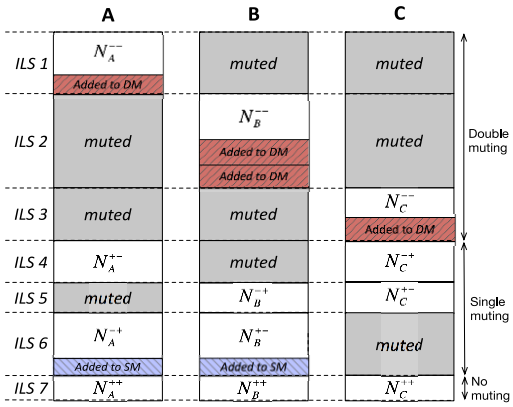

Figure 8 – Resulting frame partitioning.

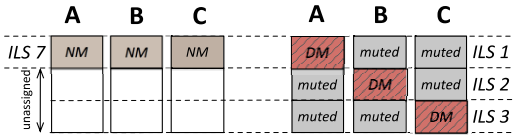

Figure 9 – Further frame optimization.

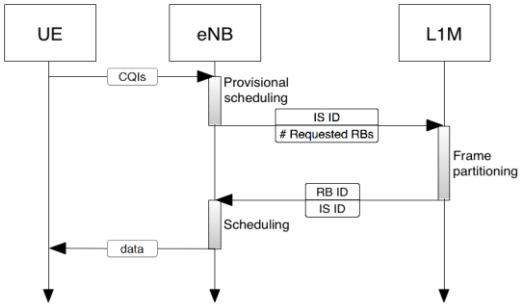

Figure 10 – SSC heuristic.

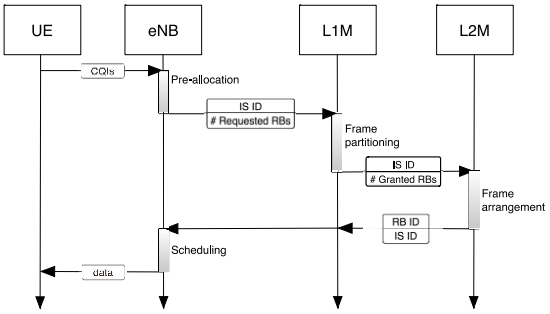

Figure 11 – LSC information flow.

b. Large-scale coordination

It is often the case that cells of neighboring clusters exert a considerable interference on the UEs of a cell. These are however subject to independent, uncoordinated instances of SSC, which may result in interference-prone schedules if interfering antennas use the same RBs. We have already ascertained that it is impractical to handle more than few interferers per UE (two, in our algorithm), hence we cannot use the SSC approach at a larger scale. We can instead exploit the fact that the *average* interference that cell *x* exerts on the area covered by cell *y* can be measured statically. Furthermore, the SSC algorithm described in the previous section exhibits a degree of freedom, i.e., the *position* of the RBs in each frame: the output of the L1M at step 2, in fact, can be permuted. More to the point, we do not even need ILSs to be contiguous in the subframe. This can be exploited to mitigate the inter-cell interference on a larger scale. More specifically, the output of the SSCs of neighboring clusters can be ar-

ranged so as to minimize the interference perceived by their UEs. We first formulate the LSC problem as an optimization problem, whose objective is to minimize the overlap of interfering cells, and then propose a heuristic solution.

We consider *C* clusters, deployed as in Figure 4. For each couple of cells *x* and *y* belonging to clusters *i* and *j* respectively, we define an Interference Coefficient (IC) $\alpha_{x,y}$, which measures the *average* interference that *x*'s UEs will suffer from cell *y*. In general, $\alpha_{x,y} \neq \alpha_{y,x}$, since cells are anisotropic. We call T_n the set of all ILSs of a cluster *n*, S_x the set of ILSs where cell *x* is active (so that $S_x \subseteq T_n$ if $x \in n$), and Δ_s the size of ILS *s* (given by the SSC). We define the following variables:

- $b_{i,s} \in \{0,1\}$, where $b_{i,s} = 1$ means that ILS *s* is allowed to use RB *i*, $1 \le i \le M$;
- $o_{i,s,t} \in \{0,1\}$, where $o_{i,s,t} = 1$ means that both ILSs *s* and *t* are allocated in RB *i*. It is, of course, $o_{i,s,t} = b_{i,s}$ AND $b_{i,t}$.

The LSC problem can be formulated as follows:

$$
\begin{aligned}
&\min \sum_{x,y} \left[\alpha_{x,y} \cdot \sum_{i=1}^{M} \sum_{(s,t) \in S_x \times S_y} o_{i,s,t} \right] \\
&s.t. \\
&o_{i,s,t} \le b_{i,s} && \forall i, \forall (s,t) \in S_x \times S_y && (i) \\
&o_{i,s,t} \le b_{i,t} && \forall i, \forall (s,t) \in S_x \times S_y && (ii) \\
&o_{i,s,t} \ge 1 - M \cdot (2 - b_{i,s} - b_{i,t}) && \forall i, \forall (s,t) \in S_x \times S_y && (iii) \\
&o_{i,s,t} \le 0 && \forall i, \forall s,t \in T_n && (iv) \\
&\sum_{i=1}^{M} b_{i,s} \ge \Delta_s && \forall s && (v) \\
&b_{i,s} \in \{0,1\} && \forall i, \forall s && (vi) \\
&o_{i,s,t} \in \{0,1\} && \forall i, \forall s,t && (vii)
\end{aligned}
\qquad (3)
$$

The objective function is to minimize the amount of overlapping RBs. Constraints (*i-iii*) are the linear version of the logical AND between variables $b_{i,s}$ and $b_{i,t}$. Constraint (*iv*) states that ILSs belonging to the same cluster cannot overlap, coherently with the SSC approach described in the previous section. Constraint (*v*) states that the sum of RBs allocated to a ILS *s* is no less than its size Δ_s. Note that equality will hold in (*v*) at the optimum in any case. Finally, constraints (*vi-vii*) show that variables are binary. This problem is a MILP with $O\left(M \cdot C^2 \cdot |T_n|^2\right) = O\left(M \cdot C^2 \cdot 2^{2K}\right)$ variables (i.e., around 120000 for seven clusters of three cells and frames of 50 RBs), thus it is infeasible to solve it at fast time scales. Figure 11 shows the information flow for the LSC. The output of the L2M is a map that

matches RBs to ILSs, which the L1M forwards to the cells in its cluster.

Heuristic solution for large-scale coordination

In order to solve the above problem fast enough, we use a divide-and-conquer approach, i.e., we split the LSC in several smaller problems. Our heuristic sorts cluster according to some order (e.g., starting from the innermost cluster in Figure 4 and going towards the outer ones), and adds one cluster at a time: at step $n \geq 2$, ILSs belonging to T_n are arranged so as to minimize the overall mutual interference with clusters T_j, $1 \leq j \leq n-1$. Clearly, ILSs of the first cluster can be placed arbitrarily in the frame. Then, the following procedure is repeated for each of the remaining *C*-1 clusters. For each couple of ILSs *s*, *t*, we define $\beta_{s,t} = \sum_{s \in x, t \in y} \alpha_{x,y}$ as the interference that cells active in *s* produce on users served by cells active in *t*. Recall that we defined $\alpha_{x,y}$ as a static coefficient measuring the average interference that UEs under cell *x* suffer from *y*. Then, we solve the following optimization problem:

$$\begin{aligned} &\min \sum_{i=1}^{M} \sum_{s \in T_n} b_{i,s} \cdot \sum_{t \text{ active in } i} \beta_{s,t} \\ &s.t. \qquad\qquad (4) \\ &\sum_{i=1}^{M} b_{i,s} \geq \Delta_s \quad \forall s \in T_n \quad (i) \\ &\sum_{s \in T_n} b_{i,s} \leq 1 \quad \forall i \quad (ii) \\ &b_{i,s} \in \{0,1\} \quad \forall i, \forall s \in T_n \quad (iii) \end{aligned}$$

The sum $\sum_{t \text{ active in } i} \beta_{s,t}$ in the objective is an estimate of the overall interference that cells active in *s* would produce if *s* included RB *i*, knowing which ILSs have been already allocated in that RB at earlier steps. Constraint (*i*) states that each ILS consists of Δ_s RBs, whereas (*ii*) states that two ILSs of the same cluster cannot overlap. The output of (4) is the frame allocation for cluster *n*, which will be taken into account for the allocation of cluster $n+1$. The heuristic requires solving *C*-1 instances of the above MILP. Note that the above problem is a Linear Assignment Problem, hence can be solved in polynomial time using the Hungarian algorithm (via minor modifications). Therefore, $O\left(C \cdot M^3\right)$ is an upper bound on the complexity of this heuristic, again independent of the number of UEs or the load. Moreover, (4) can be solved at optimality using continuous relaxation, since its coefficient matrix is *totally unimodular*. This means that (4) is in fact no harder than a LP.

Before evaluating the performance of SSC and LSC (both jointly and in isolation), we discuss the related work.

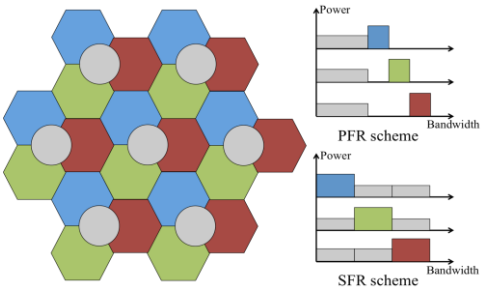

Figure 12 – PFR and SFR resource partitioning.

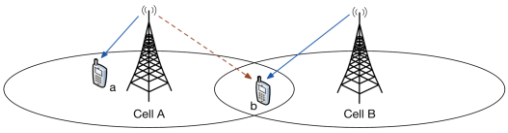

Figure 13 – Example scenario.

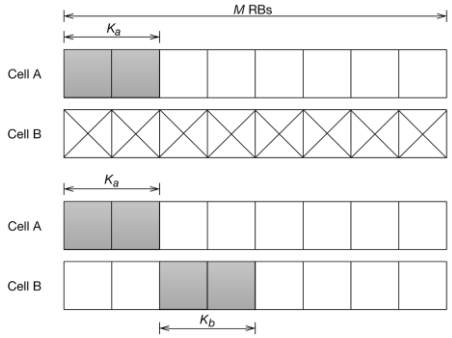

Figure 14 – Allocation using D-ICIC (top) and SSC (bottom).

5. Related Work

The simplest form of inter-cell interference mitigation are traditional frequency reuse schemes, e.g. a reuse of three. Although these techniques do reduce the interference, partitioning the overall bandwidth among cells impairs the overall throughput. Enhanced frequency reuse schemes, such as Partial Frequency Reuse (PFR) [12] and Soft Frequency Reuse (SFR) [13], have thus been introduced. The idea behind PFR is to partition the bandwidth so that only a limited amount of RBs can be used by all cells, while others are used with higher reuse factor. Cell-edge UEs can take advantage of lower interference in these sub-bands. In the SFR scheme, a cell can allocate the entire subframe, but different power levels are employed in cell-center and cell-edge RBs. Two examples of PFR and SFR with reuse 3 are shown in Figure 12. Their drawback is that the partitioning is *static*, being part of the network planning phase, hence does not take into account the dynamic UE and traffic distribution.

Semi-static Inter-Cell Interference Coordination (ICIC) schemes were then proposed, based on the above frequency reuse schemes. In [14], UEs are classified into four interference conditions according to the achievable spectral efficiency with different reuse patterns, which are similar to the four muting configurations used by our SSC. Bandwidth is partitioned according to the *number* of UE in each configuration, without taking into account their requirements. The scheme proposed in [15] achieves a semi-static PFR via a coordination algorithm – run by a central controller – that takes into account average UE rates (on all RBs) and their minimum data rate requirements. The output is a network-wide PFR scheme that each cell can then enforce. Authors of [16] also propose an optimal partitioning of the

resources based on user rate requirements. The problem with such schemes is that each RB is assigned a *fixed* reuse factor, which hampers large-scale coordination. Furthermore, such cell planning is ineffective when UEs are deployed non-uniformly, because it inherently assumes that the number of reused RBs is symmetric among the coordinate cells.

More flexible solutions have been proposed in [17] and [18]. In [17], the algorithm at the Radio Network Controller (RNC) gathers the achievable rates of all users on each RB with and without the highest ("dominant") interferer among neighboring cells. The RNC loops on each RB and selects which cell is allowed to transmit, based on the achievable gain in the overall system throughput. The RNC communicates its decisions to the cell, together with the recommended UE for each RB. If the recommended UEs have no traffic, the cell selects the backlogged UE that yields the maximum gain. This solution is opportunistic, hence unfair, at both the RNC and the cell level. Reference [18] employs the same RNC/cell framework as [17]. The RNC iteratively solves L MILPs, one per coordinated cell. At the l-th iteration, the algorithm updates the interference condition on each RB, based on the results of the previous iterations, and uses the results as coefficients for the objective function. This approach is similar to our LSC heuristic. However, it is worth noting that our LSC heuristic coordinates *triples* of cells, hence solves fewer problems (roughly one third), and much easier ones besides. For example, consider a scenario with $C = 7$ triples of cells, $M = 50$ RBs and $N = 50$ UEs per cell. The algorithm in [18] solves 21 MILPs with $M \cdot N = 2500$ binary variables each, whereas our LSC heuristic solves six *LPs* with $M \cdot \left(2^K - 1\right) = 350$ variables, each of which is polynomial. Furthermore, solutions in [17], [18] require per-UE, per-RB feedback to be conveyed *to the RNC*, exactly *because they lack pre-scheduling*. This makes it impossible, in their very authors' opinion, to run RNC coordination at timescales comparable to the TTI, hence they adapt worse to variable traffic patterns. For example, with bursty traffic sources the load may vary greatly from one TTIs to the next. The coordination opportunities of such situations can hardly be exploited if algorithms are run at coarse time granularities.

Authors of [19] proposed a resource allocation scheme that splits the frame into a *reuse* zone and a *resource isolation* zone. All cells can transmit simultaneously in the former, whereas in the latter different RBs (or *subchannels*, since the paper is based on WiMAX) are allocated to different cells, with muting requirements. UEs are scheduled in either of the two zones according to the perceived interference from neighboring cells. A central controller establishes the partitioning of the resource isolation zone based on a conflict graph, which determines which cells cannot use the same RBs. However, the

decision on the amount of RBs to allocate to a cell is only made on the number of UEs that falls in the isolation zone (the actual traffic is not considered) and is overly conservative (it is sufficient that only one UE of a given cell perceives high interference from a neighboring one in order to constrain the two cells to use mutually disjoint resources). Our algorithm, instead, allows a cell to request a number of RBs for each ILS depending on the channel quality and the buffer status of its UEs. Moreover, the output of the central controller is a long-term resource sharing plan (to be updated every hundreds of TTI), thus it is less responsive than our scheme.

Work [8] is again a two-stage scheduling mechanism. Each UE can report its two dominant interferers among a set of non-serving eNBs. UEs report the CQI *on each RB* in three possible configurations (both interferers active, both inactive, dominant interferer inactive), similarly to what we do (we also account for non-dominant interferer inactive). Using a threshold mechanism, the cell decides the optimal muting pattern for each pair UE-RB, and runs the Hungarian algorithm [20] iteratively to preassign all the RBs to UEs and to create a wish list of muting of interferers. This is sent to the central controller, which arbitrates conflicting muting requests by solving *one* MILP per RB. Then, it sends back the resulting muting pattern for each cell to enforce it.

This scheme can be regarded as a possible competitor for both our SSC and LSC. For simplicity, we refer to [8] as D-ICIC from now on. As far as LSC is concerned, we observe that D-ICIC suffers from scalability problems: on one hand, a lot more information has to be conveyed to the central controller (per-RB pairs of {UE, muting} sent from each coordinated cell, as opposed to K pairs {muting, ISL size}), similar to what is done in [17], [18]. Moreover, the number of MILPs to be solved is large (a subframe consists of 50-100 RBs), and their dimension scales with the number of coordinated cells, to such an extent that it is impossible to run it at timescales comparable to the TTI. Last, D-ICIC solves *independent* MILPs *for each RB* (capitalizing on per-RB CQIs), while our algorithm solves the allocation problem considering all RBs in a subframe *simultaneously*. While the approach with per-RB CQIs is more fine-grained, it also requires *another* algorithm to select the *one and only* MCS to be used in the presence of differing CQIs (see Section 2). As a trivial example, if D-ICIC allocates RBs 1 and 2 to the same UE, with a CQI 15 and 1 respectively, a decision is due on whether it will send traffic on: a) RB 1 only, transmitting with a CQI of 15; b) RB 2 only, with CQI 1, or c) RBs 1 and 2, this time with (possibly) CQI 1 to guarantee correct reception. While b) is obviously to be avoided, it is not immediately clear that option a) is preferable to c). Obtaining the optimal configuration becomes exponentially

hard as the number of allocated RBs increases, and even finding good *suboptimal* configurations is non-trivial. None of the above works assuming per-RB CQIs seems to take this aspect into account.

As far as SSC is concerned, we observe that the D-ICIC scheme exhibits a pathological behavior. Consider the simple scenario in Figure 13 with two cells, A and B. Call *a* and *b* the two UEs served by A and B, respectively and assume that *b* perceives high interference from A on all the spectrum, hence requests A to be muted on each and every RB. Now *b*, being cell-edge, has a smaller *utility value* (which is a function of its current and long-term rate in [8]) than *a*, which is instead cell-center. Since the central entity assigns each RB *separately* and based on the utility values, the results will be that A wins each per-RB contest, hence gets the whole frame, and B is muted on all RBs. This leads to underutilizing the resources, since *a* may not even have enough backlog to fill the frame, and B would still be prevented to use leftover RBs to address *b*. This would not happen with our SSC, which instead strives to *compose* conflicting requests from the cells, possibly by reducing them proportionally. A comparison between the two resulting allocations is shown in Figure 14, where K_A, K_B are the RBs exploited by *a* and *b* respectively.

6. Performance Evaluation

In this section we evaluate the performance of our heuristics. First, we describe the simulation models, then we provide insight on both the SSC and LSC, and finally we compare our solution to other schemes reviewed in Section 5.

a. Simulation model

Our evaluation is carried out using SimuLTE [9]-[10], a system-level simulator, comprising more than 40k lines of object-oriented C++ code. SimuLTE is developed for the OMNeT++ simulation framework [11], which includes a considerable amount of network simulation models, including the INET framework [26], with all the TCP/IP stack, mobility, wireless technologies, etc. SimuLTE simulates the data plane of the LTE/LTE-A radio access network. It allows simulation of LTE/LTE-A in Frequency Division Duplexing (FDD) mode, with heterogeneous eNBs (macro, micro, pico etc.), using omnidirectional and/or anisotropic antennas, possibly communicating via the X2 interface [27]. The SimuLTE protocol stack includes:

- A Packet Data Convergence Protocol – Radio Resource Control (PDCP-RRC) module, which performs encapsulation and decapsulation and Robust Header Compression (ROHC).

- A Radio Link Control (RLC) module, that performs fragmentation and reassembly and implements the three RLC modes, namely *Transparent Mode* (TM), *Unacknowledged Mode* (UM) and *Acknowledged Mode* (AM).
- A MAC module, where most of the intelligence of each node resides. Its main tasks are encapsulation of MAC SDUs into a TB and vice-versa, channel-feedback management, H-ARQ, adaptive modulation and coding (AMC).
- A Physical-Layer (PHY) module, that implements channel feedback computation and reporting, data transmission and reception, air channel simulation and control messages handling. It also stores the physical parameters of the node, such as transmission power and antenna profile, which allows *macro-*, *micro-*, *pico-*eNBs to be instantiated, with different radiation profiles.
- eNB scheduling in the downlink and uplink directions.

Only downlink traffic is simulated. The simulation scenario is depicted in Figure 15. We assume that the traffic is generated by a server and forwarded by a router to the serving cell of the receiver. The X2 interface is considered to be ideal (null latency and infinite bandwidth).

We consider seven sites. Each site consists of three cells radiating toward the center of neighboring hexagons. Each hexagon has three sites located on three vertices. The distance between different sites is 500m. We assume 10 MHz bandwidth, resulting in 50 RBs per frame. Path loss, shadowing and fading models are taken from [21]. Cells radiation patterns are anisotropic and attenuation is $A(\theta) = \min\left\{12 \cdot \left(\theta/70^{\circ}\right)^2, 25\right\}$, where θ is the relative angle between the cell and the receiver. Transmission power is the same over the whole bandwidth. UEs are static and randomly deployed, but equally distributed among cells. System parameters are summarized in Table 1. Statistics are gathered only in the central cluster. In order to evaluate the power consumption of the system, we use the model in [30], which assumes that the consumed power of an active eNB is an affine function of the number of transmitted RBs, i.e. $P = P_{base} + \rho \cdot n$, where P_{base} is the baseline power, n is the number of allocated RBs and ρ is the power per RB. The power model parameters are listed in Table 2.

We simulate both application-specific and synthetic traffics. More specifically, Voice over IP (VoIP) and Video on Demand (VoD) applications are simulated. VoIP is modeled according to [23]. The employed codec is the GSM AMR Narrow Band (12.2 kbit/s) with Voice Activity Detection (VAD), i.e. no packets are sent during silences. The durations of talkspurts and silence periods are distributed according to Weibull functions, coherently with a one-to-one conversation model. Header compression is employed. The set of parameters is summarized in Table 3. VoD traffic is taken from a pre-encoded H.263 trace file ([22]) whose parameters are summarized in Table 4. As far as synthetic traffic is concerned, we use Constant-bit-rate (CBR) sources generating 100-byte packets each 10 ms. The latter are used to reach saturation with a smaller number of UEs.

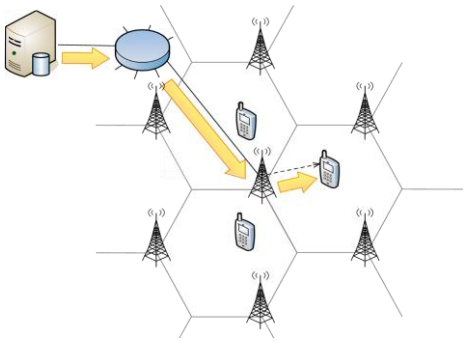

Figure 15 – Simulation scenario.

Table 1 – System parameters.

Parameter	**Value**
Cellular layout	Hexagonal grid
Inter-site distance	500 m
Carrier frequency	2 GHz
Bandwidth	10 MHz
Number of RBs	50
Path loss model	Urban Macro
Fading model	Jakes (6 tap channels)
eNB Tx Power	46 dBm

Table 2 - eNB power model.

P_{base}	260 W
ρ	3.76 W/RB

Table 3 – VoIP model parameters.

Talkspurt duration (Weibull distribution)	Scale Shape	1.423 0.824
Silence duration (Weibull distribution)	Scale Shape	0.899 1.089
Codec Type	GSM AMR Narrow Band (12.2 kbps) w. VAD	
VAD Model	One-to-one conversation	
Header Compression	Active (RTP+UDP+IP headers = 6 bytes)	
Packet length	32 bytes/frame + 6 bytes Hdr + 1 byte RLC	

Table 4 – VoD trace statistics.

Min frame size	27 Bytes
Max frame size	6806 Bytes
Mean frame size	560.703 Bytes
Mean bit rate	15.598 kbps
Peak bit rate	136.133 kbps
Frames per second	25

b. Results

First, we present performance results for our scheme: we show how to tune the thresholds of the SSC heuristic, we demonstrate the added value of employing large-scale coordination, we highlight the differences between intra-site and inter-site clustering, and we discuss the time cost and the optimality of our approach. Then, we compare our scheme to some of those reviewed in Section 5.

Small-scale coordination

We assume intra-site clustering. In this scenario, every cluster runs SSC independently, and LSC is not employed. In Figure 16, we report the average MAC-layer cell throughput achieved in the cells of the

central cluster in different load conditions, using several values for the thresholds, with CBR traffic. The figure shows that the system behaves similarly for a relatively wide range of thresholds. This is because the optimization step at the L1M increases the number of RBs in the protected ILS as much as possible, thus mitigating the effects of possible suboptimal choices of threshold values and making the algorithm more robust. Obviously, there *is* a limit to what the L1M can do to counterbalance misconfigurations: at high loads, if the thresholds are so high that most UEs end up in the no-muting ILS, which in turn fills most of the subframe, there is no room for improving the situation. In the following, we use values 5 and 2 for the double- and single-muting thresholds, respectively.

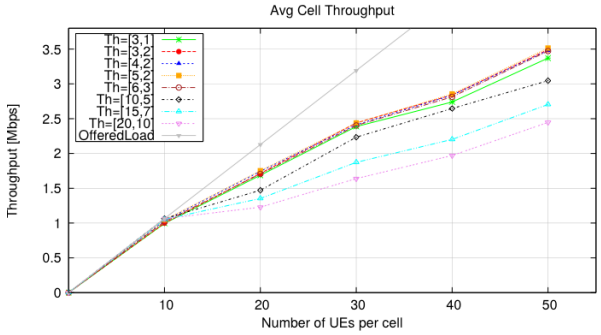

Figure 16 – Average cell MAC throughput with several thresholds

Large-scale coordination

LSC is independent of SSC and may run at different time scales. In Figure 17 we show the cumulative distribution functions (CDFs) of the UE MAC-level throughput with 50 and 75 UEs per cell, with VoIP traffic. Several periods have been tested, expressed as multiples of TTIs in the captions. Our results show that the benefits of a fast-paced LSC show up at higher loads, in terms of cell-edge throughput, identified by the 5th percentile of the CDF. Protecting cell-edge UEs is in fact a key operator requirement. Recall that the LSC takes the subframe partitioning generated by the SSC as an input. Thus, when the period is a multiple of the TTI, in between two iterations of the LSC algorithm it may happen that the subframe generated by one L1M does not fit well in the arrangement provided by the L2M earlier on. Thus, it is likely that some UEs are scheduled in inadequate RBs. Clearly, this is not the case when LSC is run on each TTI. On the other hand, protection of cell-edge UEs is not achieved at the expenses of a reduction in cell throughput, as testified by both graphs. We use a period of one for the LSC from now on.

SSC vs. LSC

We now show that using the second layer of coordination on top of the first one brings additional benefits. We report the CDFs of the UE MAC-level throughput in Figure 18. VoIP traffic is used. The green

line refers to the case with both SSC and LSC enabled, whereas the red line represents the use of SSC without LSC. In the latter case, cluster run independent, uncoordinated instances of SSC. The brown line represents the case with no coordination at all. The blue line reports an *ideal* baseline, obtained by sending the same VoIP traffic on a wired Gb-speed link. The CDF obtained with LSC practically overlaps the latter, leaving the other progressively behind as the load increases. Figure 19 reports the average MAC-layer cell throughput in the same scenario with a varying number of UEs.

LSC also impacts the user QoS. Figure 20 shows the CDF of the Mean Opinion Score (MOS) of VoIP flows in the same scenarios. The MOS measures the quality experienced by human users taking into account mouth-to-ear delays and losses (therein including those at the playout buffer) [23], and ranges from one (unintelligible) to five (perfect). A MOS above three is considered satisfactory. We observe that LSC achieves both a higher average MOS, even at low loads, and a smaller variation (which implies higher inter-user fairness) than the other two. At high loads uncoordinated resource allocation leaves about 20% of the UEs with a MOS of 1, i.e. an unintelligible conversation.

Finally, the benefits of LSC also show up when energy efficiency is considered. Figure 21 reports the average number of RBs allocated by each cell in the two cases and the resulting consumed power, showing that LSC achieves a higher throughput with roughly one third of the RBs with respect to the SSC case.

Looking at the average number of allocated RBs, one may think that the network is underloaded. For instance, in the scenario with 75 UEs per cell with SSC only (second bar from the right in Figure 21, left), a cell uses an average 12 RBs in a subframe of 50. However, muting has to be taken into account: in the same scenario, the number of *requested* RBs per cell is 22.7, and their muting requests are such that on average 40.5 RBs are requested in total (i.e., the subframe is 80% full).

Intra-site vs. Inter-site SSC

Clustering is used as a basis for the SSC. In this subsection, we show how the choice of a cluster affects the performance. Figure 22 reports the CDF of the MAC-level UE throughput with 75 UEs per cell. Using intra- or inter-site clustering does not affect fairness among UEs, for both SSC and LSC, as their CDFs are almost overlapped. Figure 23 shows the number of allocated RBs and the corresponding power consumption. Inter-site clustering allocates fewer RBs when only SSC is employed, as UEs are more protected from their major interferers. Instead, the number of allocated RBs is essentially the same when the LSC is also run. In a traditional RAN deployment, SSC can be performed with short

latency if intra-site clustering is adopted, since the clustered cells are at the same site. Inter-site clustering, instead, requires cells to be connected through additional wiring, thus it may require higher CAPEX and latency. In a C-RAN deployment, all cells are connected to a central processing unit, which performs resource allocation. In this case, the two clustering schemes are equivalent from the CAPEX and latency viewpoints.

Time cost and optimality

We now investigate the time cost of our heuristics and the optimality ratio of the SSC. We have run the SSC and LSC heuristics on a (rather low-end) PC with 4 Intel Core I7 CPUs at 2.80 GHz, 8 GB of memory and Ubuntu 14.04 OS, and the results are shown in Figure 24: the average running time of the SSC is in the order of few tens of microseconds, even at higher loads. On the other hand, the solving time of the whole LSC heuristic ranges between 1.2 ms and 1.6 ms using CPLPEX, and between 0.9 ms and 1.2 ms using the Hungarian algorithm. Although the latter times are actually above the TTI threshold, these figures confirm that solving the LSC heuristic at TTI timescales is within reach of today's technology, e.g. by employing more powerful hardware.

As for SSC optimality, Figure 25 shows a scatterplot of the optimality ratio against the whole cluster backlog in some snapshots of a simulation run with 20 UEs per cell. A fully-fledged, per-TTI comparison is made impossible by the fact that the time to solve the SSC optimization problem at optimality is in the order of minutes per TTI. The optimization problem is formulated as a MILP and solved with an optimality gap of 5%, hence optimality ratios are rescaled by 0.95 to play on the safe side. Figure 25, left, shows an average optimality ratio larger than 0.72, when *no* LSC is run. At lower loads, i.e., when optimality is probably less of a concern, more variability can be observed. Figure 25, right, shows the same result when the LSC is enabled as well. In this case, even though the offered load is the same, the average backlog is considerably smaller, thanks to the reduced inter-cluster interference. Moreover, the optimality ratio becomes considerably higher (each marker in the scatterplot encompasses a large number of overlapping points). We are unfortunately unable to provide figures for the LSC heuristic, since CPLEX refuses to solve optimization problem (3) when $C = 7$ and $M = 50$. When scaling down to a smaller dimension (i.e., $C = 3$), 40 minutes of CPLEX computation are not enough to solve problem (3) at optimality (gaps are still high, i.e., around 20%-30%). However the best results found thus far by CPLEX practically overlap those of our heuristic.

Comparison with other schemes

We compare our scheme against three reference ones, namely the non-coordinated case, the PFR scheme and the D-ICIC algorithm presented in [8]. We use the MaxC/I scheduler. In the non-coordinated scheme, each cell performs its own scheduling independently and can exploit the whole frame. This scheme maximizes the utilization of the frequency resources by each cell. However, it is severely affected by inter-cell interference, since neighboring cells can transmit on the same RBs.

We use the following static partitioning of the bandwidth for PFR: the first 20 RBs are shared among all cells, whereas the remaining RBs are employed with a reuse-3 pattern, i.e. 10 RBs per cell. We call these two partitions "cell-center subband" and "cell-edge subband", respectively. A UE i will be scheduled in the cell-center or in the cell-edge subband according to the power received from its serving cell. If $P_R^i \geq P_{th}$, then i is a cell-center UE, otherwise i is a cell-edge UE. We set P_{th} equal to -40 and -50 dBm. We also assume that the cell scheduler possesses two CQIs for each UE, one for the cell-center subband and one for the cell-edge subband, and schedules the UE in either subband using the correct CQI.

In D-ICIC, cells periodically do a pre-assignment phase using the Hungarian algorithm and send muting requests to a central controller. The latter replies to each cell indicating which RBs can be used for transmission during the next period and which cannot. Since [8] does not specify how the actual scheduling is carried out, i.e., what cells actually do once the central controller has terminated its job, we use an algorithm that first schedules UEs in the RBs they were assigned during the pre-assignment phase. If there are still backlogged UEs, we schedule them in leftover RBs using a MaxC/I scheduler. The central-level algorithm is run every 10 ms.

Figures 26-27 show the CDFs of the frame delay and loss ratio of VoD traffic with a varying number of UEs. Our scheme achieves lower delay and frame losses than the others. This is because our scheme, by improving coordination, allows higher CQIs, hence *fewer* RBs for the same transmission, and makes interference more predictable. This is important with bursty traffic, such as VoD. In fact, a low-CQIs UE may end up transmitting a (portion of a) large video frame in a single, large TB. This, coupled with unpredictable interference, considerably increases the error probability, to a point where four H-ARQ retransmissions are not enough to decode the PDU at the destination. This effect was already observed in [29], in different conditions. The behavior of the PFR scheme depends heavily on the threshold value P_{th}. With -40 dBm, more UEs will fall into the cell-edge subband than with -50 dBm. On one hand, this

makes UEs more protected from interference. On the other hand, this would overload the cell-edge subband faster, since it is only 10 RB wide. Overloading the cell-edge subband increases transmission delays, as shown in Figure 27. Our results suggest that D-ICIC causes unfairness: in fact, its delay and frame loss ratio are comparable to those obtained with PFR for about 60% of UEs, but are much higher for the others. At low loads, e.g. with 10 UEs per cell, D-ICIC is worse than no coordination at all, since it prevents cells to exploit the whole bandwidth unnecessarily (see the example of Figure 14), and the preassignment allocates the same number of RBs to each UE, regardless of their demand, which is harmful with highly variable traffic. On the contrary, the pre-allocation phase of our SSC algorithm reserves RBs according to the users' demand and CQIs. Requests may be reduced by the L1M only if they cannot be accommodated, and the muting pattern of the RBs is "upgraded" by the L1M as long as there is space to do so. Figure 28 shows the average number of allocated RBs and the average consumed power, in the scenario with 10 UEs per cell. The latter shows that, when LSC is used, fewer RBs are allocated than with the other schemes, which results in a more energy-efficient allocation.

7. CONCLUSIONS

This paper presented a resource allocation framework for dynamic CS-CoMP in LTE-Advanced networks. Coordinated scheduling addresses the problem of selecting which cells transmit in which RBs so as to mitigate the interference suffered by UEs. We showed that in general this problem cannot scale to large dimensions, in terms of number of coordinated cells, due to the amount of UE channel reporting required and the complexity involved in manipulating it. We have then proposed a layered approach, which splits the problem into *small-scale* and *large-scale* coordination. Small-scale coordination (SSC) arbitrates a small cluster of three cells, by partitioning the frame in interference logical subbands (ILSs). Each ILS defines the subset of cells that can transmit in the same RBs. SSC has been used as a basis for large-scale coordination (LSC), which was accomplished by defining the position of the ILSs in the frame, so as to minimize the interference among neighboring clusters. We modeled both SSC and LSC as optimization problems, and showed them to be too complex to be solved at optimality. Thus, we designed fast heuristics that can be run at TTI timescale. System-level simulations showed that our scheme achieves significant benefits in terms of throughput, QoS and fairness among UEs, and outperforms static and dynamic schemes proposed in the literature. Moreover, it keeps the number of allocated RBs low, thus increasing the energy efficiency.

8. ACKNOWLEDGEMENTS

The subject matter of this paper includes description of results of a joint research project carried out by Telecom Italia and 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. Authors would like to thank Prof. Antonio Frangioni and Dr. Laura Galli of the University of Pisa for their useful suggestions.

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] 3GPP - TR 36.819 v11.2.0, Coordinated multi-point operation for LTE physical layer aspects (Release 11), Sept 2013

[3] Lee, D., Seo, H., Clerckx, B., Hardouin, E., Mazzarese, D., Nagata, S., Sayana, K., Coordinated Multipoint Transmission and Reception in LTE-Advanced: Deployment Scenarios and Operational Challenges, *IEEE Communications Magazine*, February 2012, pp.148-155

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

[5] 3GPP - TS 36.211 v12.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical channels and modulation", Dec 2014

[6] 3GPP - TS 36.213 v12.4.0, "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures", Dec 2014

[7] Pateromichelakis, E., Shariat, M., ul Quddus, A., Tafazolli, R., (2012) On the Evolution of Multi-Cell Scheduling in 3GPP LTE / LTE-A, *IEEE Communications Surveys and Tutorials*, 2012

[8] Rahman M., and Yanikomeroglu, H. (2010) Enhancing cell-edge performance: a downlink dynamic interference avoidance scheme with inter-cell coordination, *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1414-1425, April 2010

[9] Virdis, A., Stea, G., Nardini, G. (2014) SimuLTE: A Modular System-level Simulator for LTE/LTE-A Networks based on OMNeT++, *proc. of SimulTech 2014*, Vienna, AT, August 28-30, 2014

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

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

[12] Sternad, M., Ottosson, T., Ahlen, A., Svensson, A. (2003), Attaining both coverage and high spectral efficiency with adaptive OFDM downlinks, *Proc. of VTC 2003-Fall*, pp.2486-2490 6-9 Oct. 2003.

[13] 3GPP, "Soft Frequency Reuse Scheme for UTRAN LTE," 3rd Generation Partnership Project (3GPP), R1-050507, May 2005.

[14] Fang, L., Zhang, X., (2008), Optimal Fractional Frequency Reuse in OFDMA Based Wireless Networks, *Proc. WiCOM '08*, pp.1-4, 12-14 Oct. 2008.

[15] Ali, S.H., Leung, V. C. M., (2009) Dynamic frequency allocation in fractional frequency reused OFDMA networks, *IEEE Transactions on Wireless Communications*, vol.8, no.8, pp. 4286-4295, August 2009.

[16] Hoon, K., Youngnam, H., Jayong, K., (2004) Optimal subchannel allocation scheme in multicell OFDMA systems, *Proc. of VTC Spring '04* pp.1821-1825 Vol.3, 17-19 May 2004.

[17] Li, G., Liu, H., (2006) Downlink Radio Resource Allocation for Multi-Cell OFDMA System, *IEEE Transactions on Wireless Communications*, vol.5, no.12, pp.3451-3459, December 2006.

[18] Koutsimanis, C., Fodor, G., (2008) A Dynamic Resource Allocation Scheme for Guaranteed Bit Rate Services in OFDMA Networks, *Proc. ICC '08*, pp.2524-2530, 19-23 May 2008.

[19] Arslan, M. Y., Yoon, J., Sundaresan, K., Krishnamurthy, S. V., and Banerjee, S., (2013) A Resource Management System for Interference Mitigation in Enterprise OFDMA Femtocells, *IEEE/ACM Transactions on Networking*, vol.21, no.5, pp.1447-1460, Oct. 2013

[20] Khun, H.W., (1955) The Hungarian method for the assignment problem, *Naval Research Logistic Quarterly*, vol.2, pp. 83-97, 1955.

[21] 3GPP TR 36.814 v9.0.0, "Further advancements for E-UTRA physical layer aspects (Release 9)," March 2010.

[22] http://www-tkn.ee.tu-berlin.de/research/trace/pics/FrameTrace/mp4/index6e27.html

[23] Bacioccola, A., Cicconetti, C., Stea, G., (2007) User level performance evaluation of VoIP using ns-2, *Proc. NSTOOLS'07*, Nantes, France, Oct. 22, 2007.

[24] Nardini, G., Stea, G., Virdis, A., Caretti, M., Sabella, D., (2014) Improving network performance via optimization-based centralized coordination of LTE-A cells", *Proc. of CLEEN 2014*, Istanbul, TK, April 6, 2014

[25] Nardini, G., Stea, G., Virdis, A., Sabella, D., Caretti, M., (2014) "Effective dynamic coordinated scheduling in LTE-Advanced networks", *Proc. of EuCNC 2014*, Bologna, Italy, June 23-26, 2014

[26] INET framework for OMNeT++: http://inet.omnetpp.org/ [Accessed July 2014]

[27] 3GPP TR 36.420 v11.0.0, "X2 general aspects and principles (Release 11)," September 2012.

[28] Accongiagioco, G., Andreozzi, M. M., Migliorini, D., Stea, G., (2013), Throughput-optimal Resource Allocation in LTE-Advanced with Distributed Antennas, *Computer Networks*, 57(2013), pp. 3997-4009, Dec. 2013.

[29] Stea, G., Virdis, A., (2014) A comprehensive simulation analysis of LTE Discontinuous Reception (DRX), *Computer Networks*, 73 (2014), pp.22-40, DOI 10.1016/j.comnet.2014.07.014, November 2014.

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

[31] Dahlman, E., Parkvall, S., Skold, J., 4G LTE/LTE-Advanced for Mobile Broadband, 2nd ed., 2014

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

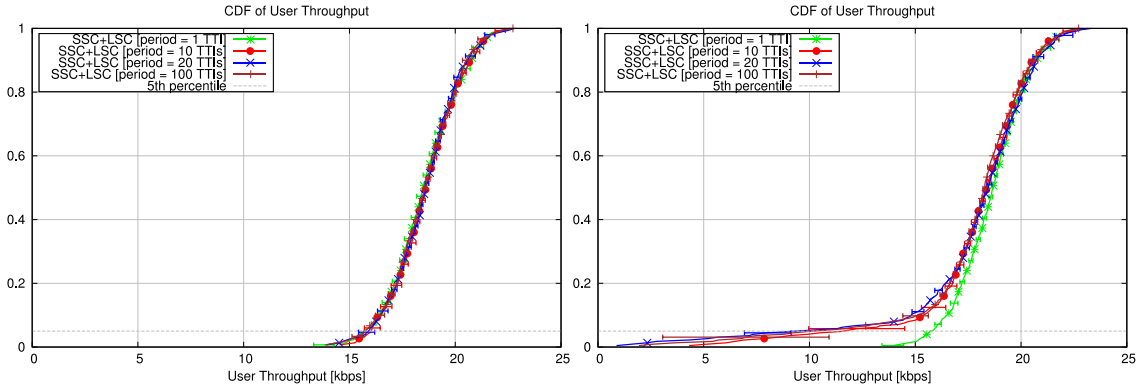

Figure 17 – Comparison of LSC timescales: CDF of user MAC throughput, 50 UEs per cell (left), 75 UEs per cell (right)

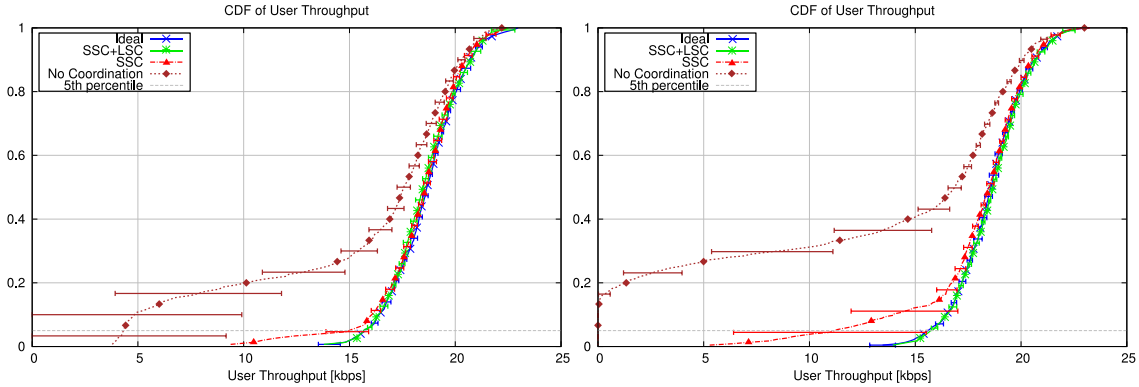

Figure 18 – Comparison of coordination schemes: CDF of user MAC throughput, 50 UEs per cell, (left), 75 UEs per cell (right).

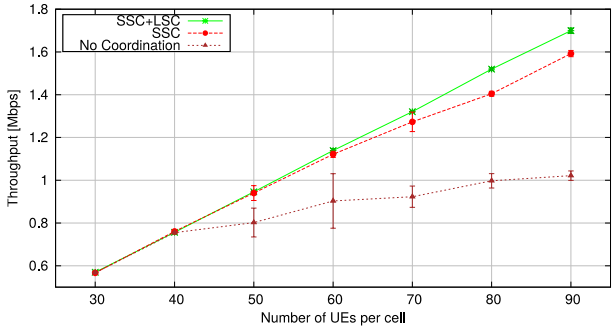

Figure 19 – Comparison of coordination schemes: cell throughput as a function of the number of users.

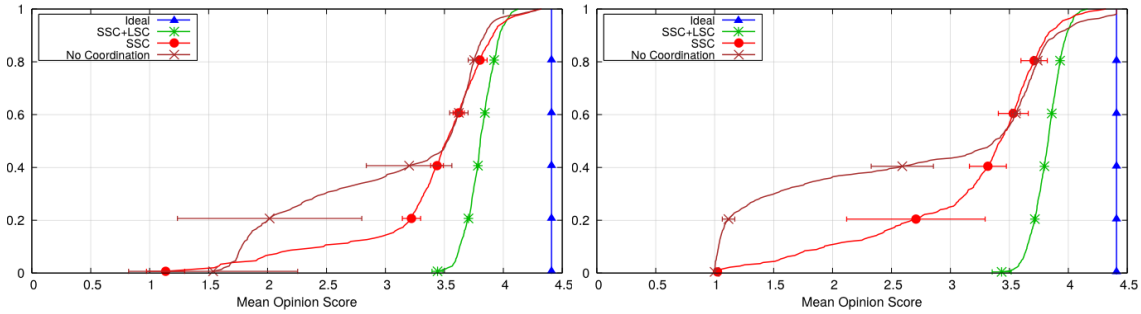

Figure 20 – Comparison of coordination schemes: MOS of VoIP flows, 50 UEs per cell (left), 75 UEs per cell (right).

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

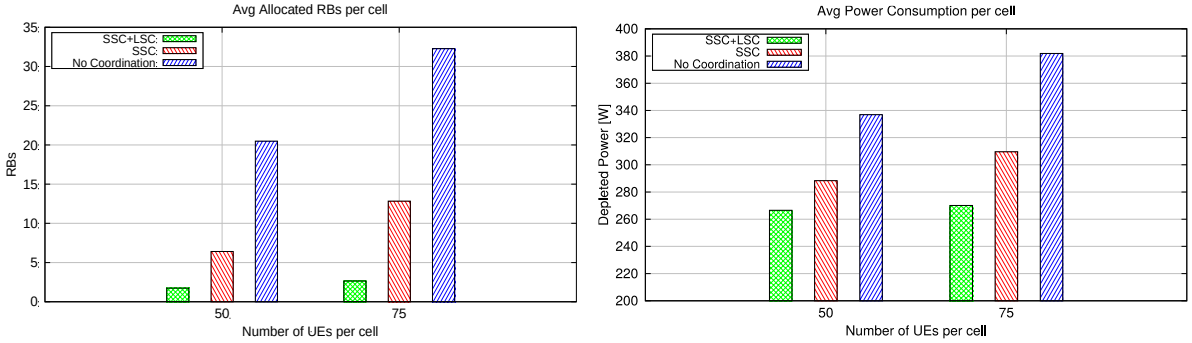

Figure 21 – Comparison of coordination schemes: number of allocated RBs per cell (left) and average consumed power (right).

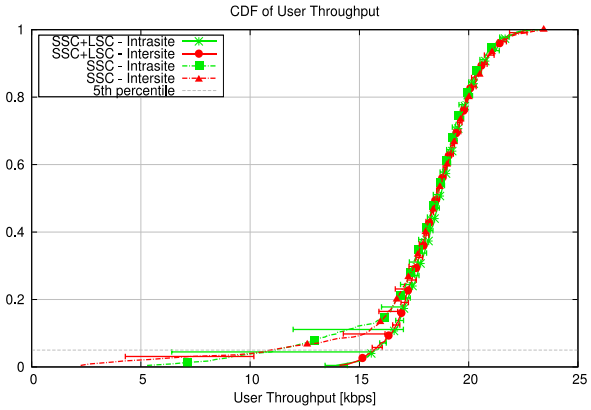

Figure 22 – CDF of user MAC throughput, 75 UEs per cell, intra-site vs. inter-site.

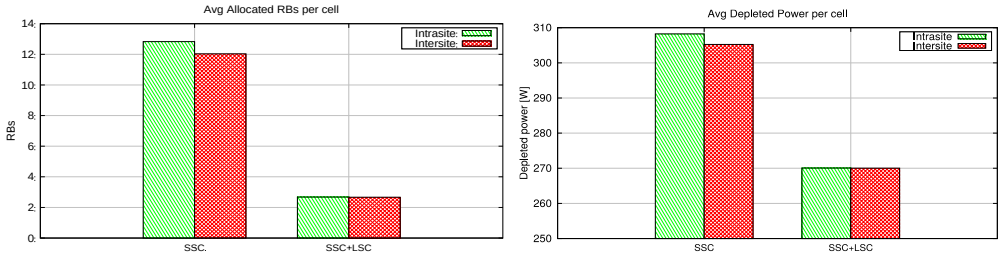

Figure 23 – Intra-site vs. inter-site, 75 UEs per cell: allocated RBs per cell (left) and average consumed power (right).

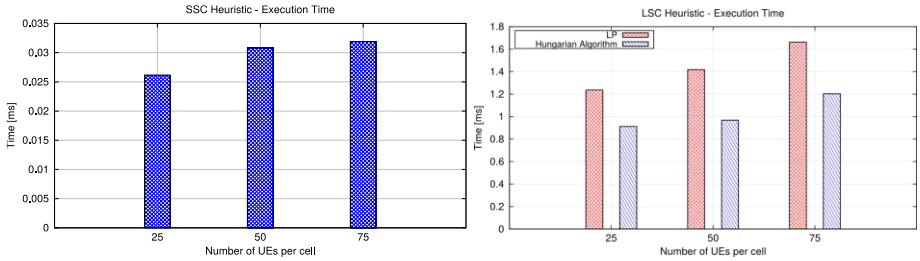

Figure 24 – Running time of the heuristics: SSC (left), LSC (right).

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

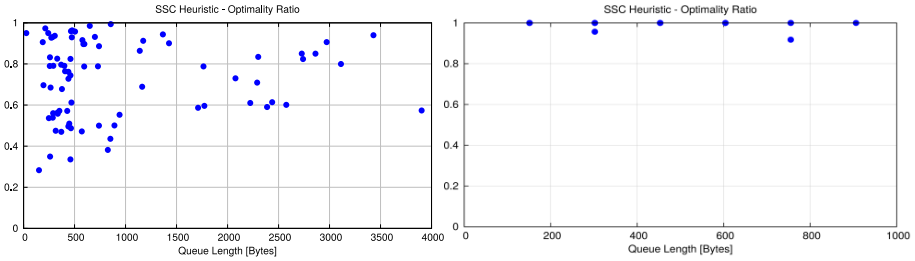

Figure 25 – Optimality ratio of the SSC heuristic as a function of the backlog, without (left) and with (right) LSC activated.

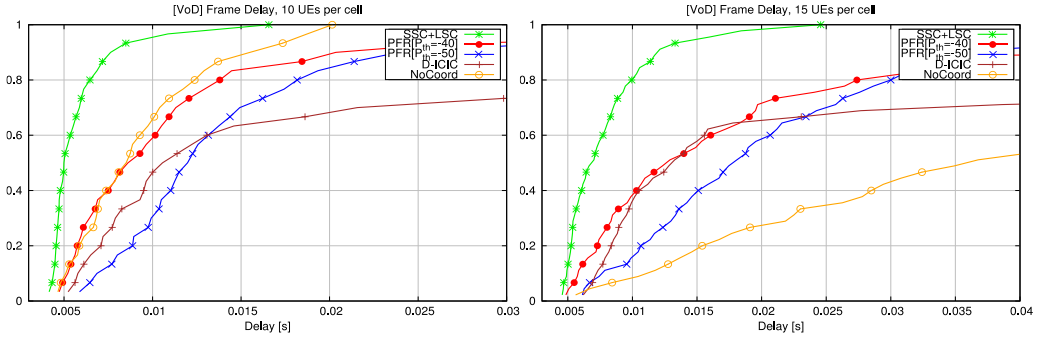

Figure 26 – VoD, Frame Delay, 10 UEs per cell (left), 15 UEs per cell (right).

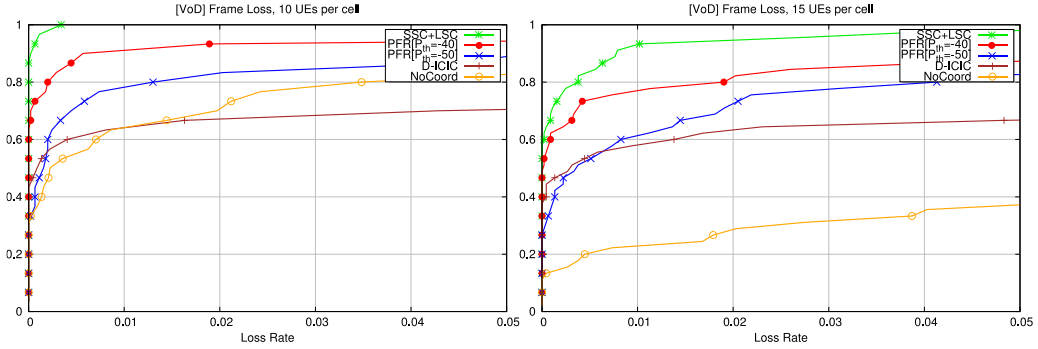

Figure 27 – VoD, Frame Loss, 10 UEs per cell (left), 15 UEs per cell (right).

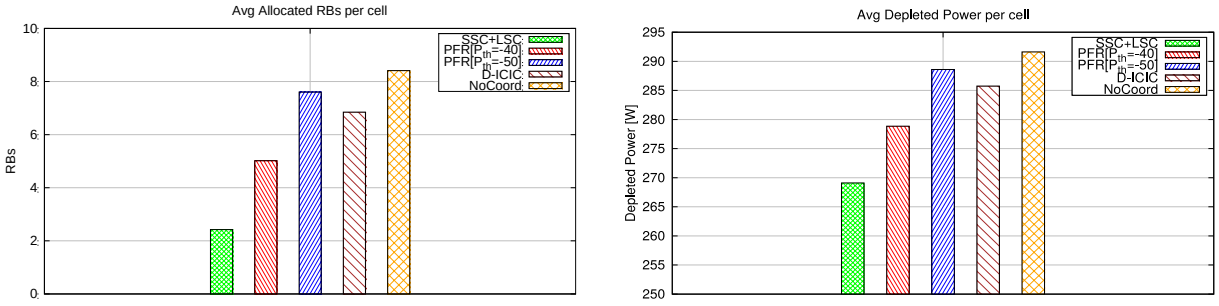

Figure 28 – VoD, 10 UEs per cell, average number of allocated RBs (left) and average consumed power (right).

This is a post-peer-review, pre-copyedit version of an article published in Wireless Networks. The final authenticated version is available online at: https://doi.org/10.1007/s11276-015-0948-6

9. APPENDIX

We provide here a formal description of the algorithms run in Step 2 of the SSC. The bid composition algorithm is reported in Figure A1. For each inequality in (2), we compute the exceeding RBs (lines 1-3). We then scan the inequalities and decrease the bids that appear in each of them (line 9), until the excess of the inequality is nullified. Since a bid appears in more than one inequality, the excesses must be updated (lines 15-16). By scanning the inequalities by decreasing order of their excess (line 4), the number of required iteration is in general smaller, as it is more likely that fixing those with larger excesses *first* will make some other inequalities hold as well.

Once the bids have been composed, we can define the ILSs. With reference to the pseudo-code in Figure A2, we denote the size of an ILS as $\Delta(x)$, where x is the set of active cells in that ILS. Double-muting ILSs are easily defined (line 1). Since the size of single- and no-muting ILSs is defined as the *maximum* among the requests from neighboring cells, there may be unused RBs in each subframe. Starting from the single-muting ILSs, for each cell we compute the size of the (possibly two) unassigned areas (lines 3-4) and fill them with as many RBs as possible from the no-muting bid (lines 5-10). Then, the size of single-muting ILSs is defined as the minimum between the corresponding bids (line 14), while their difference is added to the double-muting ILS of the cell that requested more RBs (lines 15-17). Similarly, no-muting ILS is defined as the minimum among the no-muting bids (line 19) and the excesses are redistributed to single- and double-muting ILSs (lines 20-23). According to this procedure, some of the single-muting RBs will be upgraded to double-muting, and some of the no-muting RBs will be upgraded to either double-muting or single-muting.

Finally, if there are enough unallocated RBs, we can transform one no-muting RB into three double-muting RBs. With reference to the pseudo-code of Figure A3, given the number of unallocated RBs (line 1), we compute the amount of RBs that can be moved to double-muting, taking into account that one no-muting RBs will become three double-muting RBs (line 2). That amount of RBs is carved from the no-muting ILS (line 3) and added to double-muting ILSs of the three cells proportionally, allowing for some integer rounding which preserves the original amount (lines 4-10).

```
1.  for each inequality i in (2)
2.     excess(i)=left_member-right_member;
3.  end for
4.  sort inequalities in (2) by decreasing order of excess(i);
5.  while list is not empty
6.     extract top inequality i;
7.     while excess(i)>0
8.        for each bid N(x,t) in i
9.           N(x,t)--; excess(i)--;
10.          if excess(i)==0
11.             break for;
12.          end if
13.       end for
14.    end while
15.    for each inequalities j in the list
16.       excess(j)=left_member-right_member;
17.    end for
18. end while
```

Figure A1 – Pseudo-code for the composition of the cells' bids.

```
1   Δ(A)=N(A,--); Δ(B)=N(B,--); Δ(C)=N(C,--);          // assign double-muting ILSs
2   for each x in {A,B,C}
3      u(+-)=max{N(x,+-),N(z,-+)}-N(x,+-);
4      u(-+)=max{N(x,-+),N(y,+-)}-N(x,-+);
5      D(x)=min{N(x,++),u(+-)+u(-+)};
6      if D>0
7         D(xy)=D(x)*u(+-)/(u(+-)+u(-+));
8         D(xz)=D(x)*u(-+)/(u(+-)+u(-+));
9         N(x)-=D(x);
10        N(x,+-)+=D(xy); N(x,-+)+=D(xz);
11     end if
12  end for
13  for each xy in {AB,AC,BC}
14     Δ(xy)=min{N(x,-+),N(y,+-)};                     // assign single-muting ILSs
15     D(xy)=max{N(x,-+),N(y,+-)}-min{N(x,-+),N(y,+-)};
16     w=argmax{N(x,-+),N(y,+-)};
17     Δ(w)+=D(xy);
18  end for
19  Δ(ABC)=min{N(x,++),N(y,++),N(z,++)};               // assign no-muting ILS
20  x=argmin{N(x,++),N(y,++),N(z,++)};
21  y=argmax{N(x,++),N(y,++),N(z,++)};
22  z=argmid{N(x,++),N(y,++),N(z,++)};
23  Δ(yz)+=N(z,++)-N(x,++); Δ(y)+=N(y,++)-N(z,++);
```

Figure A2 – Pseudo-code for the definition of the ILSs.

```
1   availableRBs=M-allocatedRBs;
2   D=min{Δ(ABC),floor(availableRBs/3)};
3   Δ(ABC)-=D;
4   for each x in {A,B,C}
5      D(x)=D*Δ(x)/(Δ(A)+ Δ(B)+ Δ(C));
6   end for
7   for each x in {A,B,C}
8      round D(x) so that sum(D(x))=D;
9      Δ(x)+=D(x);
10  end for
```

Figure A3 – Pseudo-code for upgrading RBs from the no-muting to double-muting ILSs.