Employing the Proactive Algorithms and the Design Structure Matrix Method for Load Balancing in UND Networks

Mohamad Salhani*
Aalto University, Espoo, Finland.

* Corresponding author. Email: mohamad.salhani@aalto.fi
Manuscript submitted May 3, 2019; accepted July 28, 2019.
doi: 10.17706/ijcce.2019.8.4.138-154

Abstract: In Ultra-Dense Networks (UDNs), the load across the small cells is not equally distributed due to the random deployment of small cells, the mobility of user equipments (UEs) and the preference of small cells during the selection and reselection. This results in performance degradation concerning the throughput and successful handovers. To address this problem, this paper proposes proactive algorithms for balancing the load across the small-cell clusters and compares their balancing results to the previous reactive algorithms. The proactive algorithms distribute the new UEs, one by one, to the small cells, while the reactive algorithms are only triggered when the load of the chosen cluster reaches a predefined threshold. In addition, this paper employs the design structure matrix (DSM) method in order to balance the load across the small cells and to reduce the inter-communications between the access points (APs) as well. The numerical analysis indicates that the load distribution and the balance efficiency using the proactive algorithm with user rejection are better than those in the reactive algorithms by 34.97% and 9.09%, respectively. Moreover, the proactive algorithm without user rejection with the DSM method achieves the best balance efficiency and reduces the inter-communications between the APs in some cases by 60.60%.

Key words: UDN, load balancing, proactive algorithm with or without user rejection, reactive algorithms, DSM method.

1. Introduction

In recent years, the wireless data demand has increased explosively, as the use of smart devices and applications has significantly increased. To fulfill the heavily growing data demands, the UDN network is considered as a promoting solution for the 5G cellular networks [1]. However, owing to the UEs mobility, the random deployment of small cells and the preference of small cells during the selection and reselection, the load across the APs becomes unbalanced. This causes network performance degradation. When UEs move onto overloaded small cells, even if neighboring cells remain underloaded, the deficit of resources in overloaded cell results in handover failures or poor quality of service (QoS) requirements, while other neighboring cell resources remain unused. Therefore, a load-balancing algorithm (LBA) becomes a necessity to redistribute the load across the APs of UDN networks in selective way with respect to some constraints imposed on the UEs.

On the other hand, the design structure matrix (DSM) method provides a simple, compact, and visual

representation of a complex system that supports innovative solutions to decomposition and integration problems used in the system engineering of products [2]. To the best of our knowledge, the DSM method has not been exploited yet in the previous studies that deal with reducing the latency and balancing the load at the same time. In this paper, the DSM method with the proactive algorithms are adapted particularly to reduce the inter-communications between the APs and also to redistribute the load across the small cells.

To balance the load and improve the performance of cellular networks, the centralized self-organized network (cSON) is a promoting solution to configure and optimize the network [3]. The cSON has many features, like mobility robustness, optimization, mobility load balancing (MLB), interference management, and so on [4]. The MLB algorithm in a cSON optimizes the handover parameters and achieves load balancing (LB) without affecting the UE experience. Thus, it is necessary to study a LBA that can adapt to various network environments and avoid the load ping-pongs.

2. Related Work

Researchers have proposed several solutions to address the load-balancing (LB) problem and enhance cellular network performance. The first LBAs within wireless networks were proposed by Balachandran and Aleo [5], [6]. Nonetheless, the proposed algorithms were very simple and only balance the load between two cells with an overlapping zone. A channel borrowing scheme has been used to offload the overloaded cells by using an unused channel from the neighboring unloaded cells in [7]. This method without a strict channel locking strategy may result in a co-channel interference. Additionally, strategies based on cell breathing and power control have been presented in [8]. These can offload the overloaded cells by simultaneously reducing the power of the APs in the overloaded cells and increasing the power of the APs in the underloaded cells. However, this can cause a disconnection of some UEs located on the cell edges and increase the co-channel interference, and the AP can remain overloaded even after reducing the coverage area. Moreover, a LBA in UDN networks based on a stochastic differential game scheme has been suggested in [9], [10] without any policy for optimizing the selection of the UEs candidate for handovers.

On the other hand, the authors in [11] proposed an MLB algorithm considering constant-traffic UEs with a fixed threshold to determine overloaded cells in Long Term Evolution (LTE) networks. Nevertheless, owing to the fixed threshold, the algorithm is not able to perform LB adaptive to varying network environments. In [12], a traffic-variant UEs LBA has been proposed considering small cells; however, this algorithm also considered a fixed threshold to identify the overloaded cells. In [13], the authors proposed an MLB algorithm considering an adaptive threshold to decide overloaded cells. The algorithm estimates the loads in both overloaded cells and neighboring cells, and achieves handovers based on the measurements reported by UEs.

The authors in [14] have mathematically proved the balance efficiency of the proposed LBAs based on the overlapping zones between the intersecting small cells. The authors focused on the optimization issue of the overlapping zone selection using different approaches. The proposed LBA was small cell cluster-based and intended first to determine the best overlapping zone among several overlapping zones and then, to select the best UE to be handed-over in order to reduce the number of the handovers and improve the performance of the whole UDN network. Nonetheless, the proposed algorithm was reactive, i.e., it is only executed when the user density of the chosen small-cell cluster reaches a predefined threshold. Besides, the reactive algorithm was not compared yet to the proactive algorithms that distribute the new incoming UEs, one by one, to the APs. Likewise, the reduction of the inter-communications between the APs was not considered by the reactive algorithm.

In this paper, we propose proactive algorithms that construct clusters of the small cells and perform the LB across the small cells. For cluster formation, the algorithm considers an overloaded small cell and two

neighboring small cells. Consequently, in each cluster the algorithm performs the LB locally and updates cell individual offset (CIO) parameters of the cells. The proposed proactive algorithms are always on standby and ready to be triggered for distributing the new UEs to the small cells. The second contribution is to employ the DSM method in reducing the inter-communications between the APs and in balancing the load across the small cells as well.

The rest of this paper is organized as follows: Section III describes the system model and assumptions we made. The different LBAs are proposed in Section IV followed by the DSM method in Section V. Section VI presents the DSM algorithms. The results of the performance evaluation are discussed in Section VII. Section VIII concludes the paper.

3. System Model

3.1. System Description

We consider a heterogeneous LTE network composed of a set of macro cells and small cells, N, and a set of users, U, as done in [13], [14]. We consider the UDN small cells with overlapping zones and each set of small cells constitutes a so-called cluster. The LB is achieved in the small-cell clusters. In the simulation model, we consider a cluster consists of three intersecting small cells, as done in [14], as depicted in Fig. 1. The cells interconnect with each other via *X2* interface. This allows them to perform the needed functionalities such as handovers, load management, and so on [15]. Therefore, the UEs can move seamlessly among the cells. To optimize the parameters in the network, a cSON subsystem is considered [4]. The cells are connected to the cSON subsystem via *S1* interface [16]. The cSON subsystem collects the required load-related information from the network and optimizes the parameters of the cells to perform the LB process.

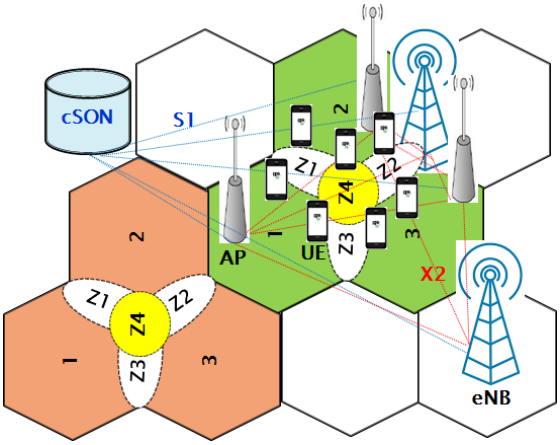


Fig. 1. System model with a cSON.

3.2. Small Cells Load

To measure the small cells load in each cluster, the average resource block utilization ratio (*RBUR*) is calculated from the physical resource blocks (*PRBs*) allocation information, as done in [13]. The small cell load, ρ_i, of cell *i* for a given time duration, *T*, is given as

$$\rho_i = \frac{1}{T.N_{PRB}} \sum_{j=1}^{u} RB_{(i,j)} \quad (1)$$

where N_{PRB} and $RB_{(i,j)}$ denote the total *PRBs* and the total allocated *PRBs* for all the UEs, *U*, in cell *i*, respectively. Hence, the average cluster load, *ACL*, is calculated as

$$ACL = \sum_{i=1}^{m} \rho_i \Big/ m \tag{2}$$

where m is the maximum number of the small cells constituting the cluster. In order to determine overloaded, balanced and underloaded small cells in each cluster, we introduce two adaptive thresholds; upper and lower thresholds, δ_1, δ_2, respectively, as done in [14] as follows

$$\delta_1 = ACL + \alpha \times ACL \tag{3}$$

$$\delta_2 = ACL - \alpha \times ACL \tag{4}$$

where α is the tolerance parameter, which controls the balance zone's width. A small value of α requires many handovers to reach the needed balance, and vice-versa. In this paper, α is set to 0.05, as done in [14]. Equation (3) and (4) show that the thresholds are a function of *ACL* and α.

3.3. Handover Procedure

In this paper, A3 and A4 event measurements are used to trigger a handover and select the UEs candidate for handovers, and the reference signal received power (*RSRP*) is assumed reporting signal quality for measurements [13], [17]. Actually, event A3 is widely used for triggering handovers in wireless networks [18]. In that way, event A3 is triggered and the UEs report the measurement results to the serving cell when the signal of a neighboring cell in a cluster is offset better than that of the serving cell. If the event A3 triggering criteria remains satisfied for longer than the time to trigger (TTT), the cell decides to trigger a handover. The event A3 measurement is reported if the following condition is satisfied [13]:

$$Mn + Ofn + Ocn - Hyst > Mp + Ofp + Ocp + off \tag{5}$$

where *Mn* and *Mp* denote the average *RSRP* values. *Ofn* and *Ofp* are the frequency-specific offsets. *Ocn* and *Ocp* are the cell individual offsets for the target and the serving cells, respectively. *Hyst* is the hysteresis parameter. *Off* is the A3 event offset between the serving and the target cells. The cSON performs the LB by shifting the UEs in the overloaded cells to the underloaded cells. However, to balance the load, the system needs information about the edge-UEs distribution. For that, the event A4 is used. All the cells share the UEs information with the cSON. The condition for triggering event A4 is expressed as done in [13],

$$Mn + Ofn + Ocn - Hyst > Thresh \tag{6}$$

where *Thresh* is event A4's threshold. The UEs that satisfy this condition report measurements for the serving and neighboring cell within the cluster in question. In that way, each cell makes a set of edge-UEs based on A4 event reports. Then the cSON collects all the edge-UEs' information from all the cells. The LBA in its turn selects the best candidate edge-UE and hands over it to the best target cell according to the chosen LB scheme.

4. Proposed Load Balancing Algorithms

In this section, we present the different proposed LBAs that can be used to balance the load across the small cells.

4.1. Proactive Algorithm with (User) Rejection (ProR)

The proactive algorithm with rejection (ProR) distributes the new UEs to the covering APs and rejects the extra-unconstrained users, as depicted in Algorithm 1. This algorithm is always on standby and ready to be triggered each time a new UE enters the network. In the ProR, the resources of the APs are considered

limited; each AP has a maximum capacity, ρ_{th}. For each new UE, the algorithm selects the best AP. The selected AP is the least loaded AP so that if the load of this unconstrained UE, $RBUR_j$, is added to the load of this AP, ρ_i, the new AP's load should not exceed ρ_{th}. If there is no AP satisfies this condition, the unconstrained UE is rejected. In contrast, if the new UE is a constrained UE, this UE will be accepted by the chosen AP even if this results in exceeding the ρ_{th} limit. A constrained UE is a user that is communicating with another one located in the same cluster, which is a so-called DSM UE. This process is repeated for each new UE moves onto the network until the user density, D of the chosen cluster reaches the density threshold, D_{th}.

4.2. Proactive Algorithm without (User) Rejection (Pro)

The proactive algorithm without rejection (Pro) is similar to the ProR, as depicted in Algorithm 2. However, the APs are considered having enough resources (e.g. ρ_{th} is greater than that in the case of ProR by 20%) to accept the new UEs as long as the user density of the current cluster does not exceed D_{th}. Therefore, this algorithm does reject the DSM UEs, even if this slightly deteriorates the load balancing process. In practice, the density condition is not necessary to be checked, as this algorithm is always on standby and triggers for each new UE. This condition is only imposed in this study in order to compare the results of these two proactive algorithms to those in the previous reactive algorithms with the same user density.

4.3. Reactive Algorithm (Rea)

The reactive algorithm (Rea) has been proposed in [14] to balance the load across the APs. Nonetheless, this algorithm is only triggered when the user density of the cluster reaches D_{th}. To achieve the reactive algorithm, the authors have proposed three approaches based on the overlapping zones concept. In the common zone (CZ) approach, the load is only balanced via the UEs that are located in the CZ between the three overlapping small cells; zone 4 (Z_4), as shown in Fig. 1. The second approach is the so-called worst zone (WZ) approach. The LB in this approach is performed in the WZ, which has the smallest value of the Jain's fairness index, β (explained later). Note that the balance efficiency of the WZ approach has been mathematically proven in [14]. The third approach is the mixed approach (MA). This approach is a hybrid approach that combines the CZ approach and the WZ approach. It starts balancing the load in the CZ and then, it transits into the WZ with or without returning to the CZ.

The reactive algorithm is adopted again in this paper to compare it to the proactive algorithms and to the DSM method. This algorithm is periodically executed in the cSON subsystem. To achieve the LB, the algorithm needs to identify the cluster with the highest density and then, the overlapping zone and the best candidate UE (BC) to be handed-over. For that, it **first** starts checking the user density, D within each cluster and then, it compares the density of the cluster with the highest density to the density threshold, D_{th}. If the user density does not exceed the threshold, the algorithm is stopped. Otherwise, the algorithm sets the UE's load, $RBUR_j$ of each UE_j, its zone and the tolerance parameter α. Next, the algorithm calculates the load of each AP, ρ_i, and the ACL with (1) and (2), respectively. Meanwhile, the algorithm determines the state of each AP by the transfer policy. This policy verifies which AP must exclude an UE (overloaded AP) and which one must include this UE (underloaded AP). For that, two thresholds, δ_1 and δ_2 with (3) and (4) are needed. According to the transfer policy, an underloaded AP can accept new UEs and handed-over UEs from an overloaded AP. A balanced AP can only accept new UEs, while an overloaded AP does not receive any new or handed-over UEs. In the **second** step, the algorithm checks if there is at least one overloaded AP within the cluster with the highest user density (cluster of first order). If not, the algorithm transits into the cluster of second or third order successively and rechecks the user density condition. If this condition is not satisfied in these three clusters, the algorithm is stopped. Otherwise, the algorithm calculates the Jain's fairness index (β) [19] as follows

$$\beta = \left(\sum_{i=1}^{n} \rho_i\right)^2 \Big/ \left(n \times \sum_{i=1}^{n} \rho_i^2\right) \tag{7}$$

where n is the number of the small cells that overlap on the zone in question, i.e., each overlapping zone has its own β. When all the APs have the same load, β is equal to one. Otherwise, β approaches $1/n$, so $\beta \in [1/n, 1]$. The **third** step is to apply the selection policy for identifying the BC candidate for handover. For that, the difference (Δ) between the load of the chosen overloaded AP and the *ACL* is calculated by

$$\Delta = \rho_{overloaded\_AP} - ACL \tag{8}$$

Of all the UEs located in the overlapping zone in question and connected to the chosen overloaded AP, the BC is the one for which the difference of the UE's load and Δ has the smallest absolute value as follows

$$BC_j = \left| RBUR_j - \Delta \right| \tag{9}$$

Note that some constrained users may be excluded from any handovers, as it will be explained later.

The **fourth** step is to calculate the new β if the BC is handed-over. This is performed by the distribution policy to ensure that the expected handover will definitely improve the balance before achieving the handover. Thus, the handover will be carried out if and only if β_{new} is greater than β_{old}. If this condition is satisfied, the algorithm selects this BC and the handover occurs. Otherwise, the algorithm transits into the next target zone. The target zone is one of the overlapping zones, which changes or not according to the selected LB scheme. For instance, the target zone in the WZ approach is the zone that has the smallest value of β, as depicted in Algorithm 3. Then, the algorithm repeats the last policies in the new target zone. The **fifth** step is to check if there is still an overloaded AP and also, if the balance improvement is still valid. If so, the balance enhancement is evaluated again in the new target zone and so on. Otherwise, the algorithm is stopped and waits for the next trigger.

Algorithm 1: Proactive algorithm with rejection (ProR)

```
1: Get RSRP and PRB measurements of UE j and cell i, Dth and UE's zone
2: if D < Dth then
3:       Find the cell that covers this UE and has the smallest ρi
4:       if ρi < δ1 and (ρi+RBURj) > ρth then
5:            Reject this UE and update the call drop rate (PR)
6:       else
7:          Transfer the new UE to the target cell
8:          Update ρi of the target cell
9:       end if
10: end if
```

Algorithm 2: Proactive algorithm without rejection (Pro)

```
1: Get RSRP and PRB measurements of UE j and cell i, Dth, and UE's zone,
2: if D < Dth then
3:     Find the cell that covers this UE and has the smallest ρi
4:     Transfer the new UE to the target cell
5:     Update ρi of the target cell
6: end if
```

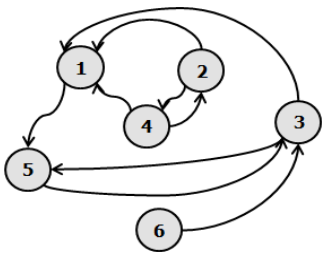
Algorithm 3: Worst zone algorithm (WZA)

```
1: Get RSRP and PRB measurements of UE j and cell i, Dth, UE's zone and α
2: Find the cluster with the highest user density
3: if D >= Dth then
4:       Calculate ρ for each cell i, ACL, δ1 and δ2
5:          if one of the chosen cluster's cell has ρi > δ1 then
6:               Calculate β1, β2, β3 and β4, and then find the worst zone
7:               Apply the transfer policy
```

```
8:              Calculate Δ and determine the BC_j
9:              if β_new > β_old then
10:                Transfer the BC_j to the target cell (execute a handover)
11:                Update ρ for each cell i and go to step 5
12:            else
13:                 if there are UEs of 2nd order then
14:                     Find the new BC_j and execute a handover
15:                     Update ρ for each cell i and go to step 5
16:                 else
17:                     Transfer to the zone of 2nd order and go to step 7
18:                 end if
19:            end if
20:        else
21:             if there is a cluster of the next order then
22:                 Go to step 3
23:             end if
24:        end if
25: end if
```

5. Use of the DSM Method

In the following, the design structure matrix (DSM) method is employed in reducing the inter-communications between the APs in addition to balance the load across the small cells. In fact, the DSM method deals with partitioning of graphs in order to realize a cooperation between the nodes (terminals), and to organize complex tasks in projects with respect to parallel, consecutive and coupled tasks. A simple example is considered to explain this method. Fig. 2 (a) shows a graph composed of six nodes, which communicate together to perform a predefined task. The intended aim is to redistribute these terminals on two nodes (switches/APs) with respect to the type of tasks (serial, parallel). For that, the adjacency matrix *A(G)* is **first** determined, as demonstrated in Fig. 2 (b). This matrix is concerned with the direct arcs among the nodes, i.e., the directional and the short communications between the current node and the neighboring nodes. Each arc that starts from a node and heads to another is represented by "1", while the other cases are left empty ''0''.



	1	2	3	4	5	6
1	X				1	
2	1	X		1		
3	1		X		1	
4	1	1		X		
5			1		X	
6			1			X

Fig. 2. The graph of the nodes (a) and adjacency matrix (b).

Second, the attainability matrix *R(G)* is determined. The latter takes care of the direct and indirect connections between the nodes. Each arc starting from a node and reaching another, even after many hops, is represented by "1", and the other cases are left empty ''0''. **Third**, the parallel and in series tasks are deduced as follows. The coupled components matrix *C(G)* is obtained by

$$C(G) = R(G) \ \ AND \ R(G)^t \tag{10}$$

Each row from the *R(G)* matrix is multiplied by the corresponding column of this matrix and the results are put in the new row of the *C(G)* matrix. **Fourth**, the new groups (components) are determined after reordering these groups using reorganized *C(G)* matrix, as depicted in Fig. 3.

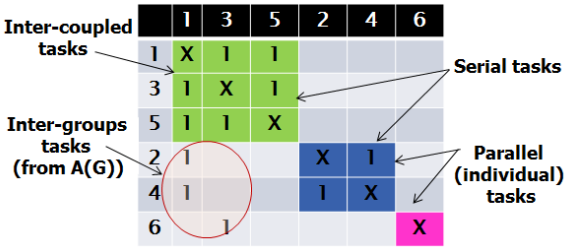


Fig. 3. The reorganized *C(G)* matrix.

This matrix clarifies the relationships between the new groups, i.e., the parallel and serial tasks, whereas the inter-group tasks are deduced from the *A(G)* matrix. Therefore, the new groups become as follows: C_1= (1, 3, 5), C_2=(2, 4) and C_3=(6). We notice that the nodes of group C_1 are inter-coupled tasks. While the groups C_2 and C_3 are parallel tasks, the groups C_2 and C_3 are in series with C_1. Consider each switch has only four ports. The nodes can be partitioned on the two switches as follows: C_1=(1, 3, 5) and C_2=(2, 4, 6). This distribution can be developed using a refinement method in order to reduce the inter-communications between the switches in the following manner. The replacement gain for each node is introduced. It is the difference between the number of the connections of a node with the other groups and the number of the connections of this node with the nodes existing in its group. Refer to the above-mentioned distribution; we refine it according to the replacement gain concept. Fig. 4 illustrates the gain matrix of each group (G_1 and G_2) in the two steps of the refinement.

	G1			G2		
	1	3	5	2	4	6
G1	2	3	3	1	1	1
G2	2	1	0	2	2	0
Gains	0	-2	-3	-1	-1	1

	G1			G2		
	3	5	6	1	2	4
G1	3	2	1	2	0	0
G2	1	1	0	2	3	3
Gains	-2	-1	-1	0	-3	-3

Fig. 4. The refinement steps of the DSM method.

The refinement process is based on checking and taking care of the nodes with positive gains (node 6 in G_2 with G=1). Accordingly, node 6 must be replaced by node 1, which has the biggest gain within G_1 with G=0. In this context, the load index, τ is introduced with intent to evaluate the replacement performance. The load index, τ is defined as the ratio of the number of inter-group connections, N_i, to the total number of interconnections of all the nodes, N_t as follows:

$$\tau = N_i / N_t \tag{11}$$

The new distribution of the nodes becomes C_1=(3, 5, 6) and C_2=(1, 2, 4). The initial value, τ_{inital} is 3/9 and the final value, τ_{final} after the refinement becomes 2/9. Consequently, the inter-communications between the switches are reduced using the refinement concept. The refinement process is stopped once all the positive values of gains become negative or at least get zero. The question is how the DSM method and the refinement process can be employed in balancing the load within the small cells of UDN networks with or without the proactive algorithms. In fact, the reactive algorithms are triggered when the user density condition is satisfied and the DSM is applied after distributing the UEs to the APs by the proactive algorithms. Therefore, at that time and in these both cases, the UEs have already been connected to the APs and each AP has already been constituted a group of some connected UEs. Accordingly, the required task is only how the refinement process can be applied. Actually, to apply the DSM method, either, the user replacement stage is first applied and then, the balancing stage is carried out by one of the previous reactive

algorithms (CZ, WZ or MA). This policy is achieved by the DSM_first algorithm (DSMf). Or, the constraints of the DSM method are respected by the LBA during the selection policy. This policy is performed by the DSM_included algorithm (DSMi). In both policies, the DSM constraints impose that the selection of an UE to be replaced is only possible if the number of hops of the user's connection is kept constant, i.e., the index τ remains constant, or rather this number of hops will be reduced from 3 to 2 hops. Thus, the DSM method aims to reduce as far as possible the end-to-end (E2E) delay between the DSM UEs in addition to balance the load across the small cells.

6. DSM Algorithms (DSMAs)

To apply the DSM method, two DSM algorithms (DSMAs) are proposed without or with one of the proactive algorithms as follows. First, the two types of the DSMAs are described without the proactive algorithms. Second, the DSM method is applied with the proactive algorithms to improve the LB more and reduce the inter-communications between the APs at the same time.

6.1. DSM Algorithms without Proactive Algorithms

The DSMf first reduces the inter-communications between the APs and then, it starts the LB using one of the previous reactive LBAs. Alternatively, in the DSMi, the replacement gain of each UE is taken into account during the steps of the reactive LBA. Indeed, the DSMi is one of the LBAs (CZ, WZ or MA); however, during the selection policy the replacement gain is respected as follows. The selected UE will not be the BC and thereby handed-over, if this handover will increase the replacement gain. Otherwise, the algorithm selects the UE of second order at the cost of decreasing the quality of balance. Thus, the selected UE is the one that has the highest value of the replacement gain.

The DSMf, which is depicted in Algorithm 4, **first** checks if the D of the cluster with the highest density exceeds the D_{th}. If this condition is not satisfied, the algorithm is stopped and waits for the next trigger. Otherwise, the algorithm sets the load of each UE, the UE's zone and α. Then, it calculates the following values: ρ_{AP1}, ρ_{AP2}, ρ_{AP3}, ACL, δ_1, and δ_2. In the **second** step, the algorithm checks if there is at least one overloaded AP within the chosen cluster. If not, the algorithm transits into the second order cluster or even to the third order one from the user density perspective. If the density condition for these three clusters is not satisfied, the algorithm is stopped. In case there is at least one overloaded AP, the gain matrix for each AP is computed. Each matrix represents the replacement gains for the UEs connected to the AP in question. In the **third** step, the algorithm searches, in the gain matrix of the most loaded AP, for an UE that has the highest positive gain and is connected to this AP. This means that this UE is currently communicating with another UE (its partner), which is connected to another AP. If there is no UE that has a positive gain, the algorithm goes to the next most loaded AP. Conversely, if there are many UEs satisfying these conditions, the UE with the highest load and positive gain is selected. In the **fourth** step, the algorithm checks the coverage condition: the AP of the candidate UE and the AP of the partner should cover the two UEs. If so, this means that anyone of them can be transferred (handed-over) to the AP of the other. Thus, the algorithm replaces the selected UE by the BC. The selected UE is a DSM UE, has the highest load, is connected to the most loaded AP and has the highest positive gain. Alternatively, the BC is the one that is connected to the partner's AP, is located in the same zone of the selected UE and has the lowest load. The BC can also be a DSM UE with a positive gain, if it is not a partner to the selected UE. In case the coverage condition is not satisfied, the **fifth** step is to check if there are still other DSM UEs that have a positive gain and are connected to the most loaded AP. If so, a new DSM UE is selected and the third step is repeated. Otherwise, the algorithm transits into the next most loaded AP and repeats the third step. When all the APs are checked and the replacement process is over, the algorithm calls one of the reactive LBAs to check again if the balance improvement is still valid and it continues the balance task as usual. Likewise, the DSMi is one of

the previous reactive LBAs, which respects the DSM constraints. Consequently, the DSMAs will reduce the inter-communications of the UEs by making the gain of all the UEs negative, G=-2, and balancing the load at the same time.

Algorithm 4: DSM_first algorithm (DSMf)

```
1: Get RSRP and PRB measurements of UE j and cell i, Dth, UE's zone and α
2: Find the cluster with the highest user density
3: if D >= Dth then
4:       Calculate ρ for each cell i, ACL, δ1 and δ2
5:          if one of the chosen cluster's cell has ρi > δ1 then
6:                Compute the gain matrix for each AP
7:                Find an UE connected to the most overloaded AP and has the most positive gain
8:                  if each AP covers the two UEs then
9:                       Replace the selected UE by the BC
10:               end if
11:                      if there are other positive gain UEs then
12:                           Go to 7
13:                      end if
14:                           if there are other APs then
15:                                Determine the AP of the 2nd order and then, go to 7
16:                           else
17:                                call one of the LBAs
18:                           end if
19:      else
20:            if there is a cluster of the next order then
21:                  Go to step 3
22:            end if
23:      end if
24: end if
```

6.2. Proactive Algorithms with DSM Method

In this case, once the UEs distribution to the APs, using the ProR or the Pro, is over, the ProR or the Pro calls the DSMi or the DSMf to balance the load again and also to reduce the APs' inter-communications, as explained previously. Note that the ProR respects the constrained UEs during the distribution stage, i.e., the ProR does not reject any DSM UE; even if the target AP will be a little overloaded if this AP includes the DSM UE in question. Because the DSM method will redistribute again the load across the APs and reduce the effect of the DSM UEs on the LB.

7. Performance Evaluation

7.1. Simulation Environment

To evaluate the performance of the proposed algorithms and compare their results to the previous reactive algorithms, we performed the simulation with a heterogeneous network with macro and small cells. The proposed scenario consists of three macro cells and 10 small cells. Each set of three-hexagonal intersecting small cells forms a cluster. The user density, D is on average equal to six UEs per small cell. Therefore, the density threshold, D_{th} is equal to 18 UEs per cluster, as considered in [14], [20]. The UEs allocate multi-traffic. Each UE selects a specific bit rate in the range of 0 to 350 Mbps [14], [21].

We consider a uniform deployment of small cells in order to diagnose the impact of the proposed algorithms on the network from different aspects. With regard to the UEs distribution, 50% of the mobile UEs were randomly distributed over the whole area, and the rest were fixed and uniformly distributed over the border areas of the small cells, because the proposed algorithms aim to hand over the UEs located in the overlapping zones. The randomly distributed UEs follow the circular way (CW) mobility model [13], [22]. In

this mobility model, the UEs move in a circular path with a 10m radius and a speed of 3.6 km/h. The bandwidth for each small cell was set to 20 MHz. The transmission power for the small cells and macro cells was set to 24 dBm and 46 dBm, respectively. To model the path loss, we considered non-line-of-sight (NLoS) propagation loss model [13], [23]. To allocate the *PRBs* among the UEs in a cell, a channel QoS-aware (CQA) scheduler was adopted [13], [24]. More parameters are listed in Table 1.

Table 1. Simulation Parameters

Parameters	**Values**
Number of small cells	10
Tx power	24 dBm (small cell) and 46 dBm (macro cell)
System bandwidth	20 MHz
Antenna mode	Isotropic
Pathloss	*PL=147.4+43.3log10(R)*
Fading	Standard deviation 4 dB, lognormal
Resource scheduling	CQA scheduler
CIO_{min} and CIO_{max}	-6dB, 6dB
Hysteresis	2 dB
ρ_{th}	1Gbps
D_{th}	18 *UE*
UE velocity	3.6 km/h
Mobility model	Uniform, 50% CW mobility UEs and 50% static UEs

7.2. Performance Evaluation Metrics

To evaluate the performance, we considered four aspects: the load distribution across the small cells, the balance improvement ratio (*BIR*), the balance efficiency (*BE*) and the reducing inter-communications ratio between the APs (*RICR*). To measure the load distribution, the standard deviation (σ) and the Jain's fairness index (β) with (7) are considered. The *BIR* is expressed as done in [14],

$$BIR = \left| \frac{\sigma_{final} - \sigma_{initial}}{\sigma_{initial}} \right| \tag{12}$$

where $\sigma_{initial}$ and σ_{final} are the standard deviation of the loads among the small cells of the cluster before and after applying the LBA in question, respectively. We also took into account the signaling load, i.e., the handover rate, *HOR* for the reactive algorithms and the probability of rejection (call drop rate) of the new incoming UEs, *PR*, for the ProR. The *BE* is measured by considering the standard deviation and also the signaling load performed in each algorithm, as done in [14]. When applying the reactive algorithm, the *BE* is given by

$$BE_{rea} = 1/(\sigma_{final} \times HOR) \tag{13}$$

By applying the ProR or the Pro, the *BE* is expressed respectively as

$$BE_{\mathrm{Pro}R} = 1/(\sigma_{final} \times PR) \tag{14}$$

$$BE_{\mathrm{Pro}} = 1/\sigma_{final} \tag{15}$$

With regard to the DSMA, the *BE* for the DSMi or DSMf is given respectively as follows

$$BE_{DSMi} = 1/(\sigma_{final} \times HOR) \tag{16}$$

$$BE_{DSMf} = 1/(\sigma_{final} \times (HOR + 2 \times RR)) \tag{17}$$

When the DSMA (DSMi or DSMf) with the ProR or Pro is considered, the *BE* is expressed respectively as

$$BE_{ProR\&DSMi} = 1/(\sigma_{final} \times (HOR + PR)) \quad (18)$$

$$BE_{ProR\&DSMf} = 1/(\sigma_{final} \times (HOR + 2 \times RR + PR)) \quad (19)$$

$$BE_{Pro\&DSMi} = 1/(\sigma_{final} \times HOR) \quad (20)$$

$$BE_{Pro\&DSMf} = 1/(\sigma_{final} \times (HOR + 2 \times RR)) \quad (21)$$

The *RICR* % between the APs is expressed as done in [14],

$$RICR\ \% = \left| \frac{\tau_{final} - \tau_{initial}}{\tau_{initial}} \right| \quad (22)$$

where $\tau_{initial}$ and τ_{final} are the load index before and after applying the DSMA, respectively.

7.3. Results Analysis

In the following, we compare the results of the proposed proactive algorithms (ProR and Pro) with or without considering the DSMA to the previous reactive algorithms.

Fig. 5 and Fig. 6 show the standard deviation of the load distribution across the small cells of the cluster as a function of the running time, for the different algorithms with the ProR or with the Pro, respectively. To balance the load, the ProR and the Pro only carry out a distribution stage for the new UEs incoming to the APs. Nevertheless, for the rest of the algorithms, there is another stage, which is the balancing stage achieved by the DSMi or the DSMf. On the contrary, the reactive algorithm only performs the balancing stage, when the user density *D* reaches D_{th} for the chosen cluster. As shown in the two figures, the Pro&DSMf&MA and the ProR&DSMf&MA take the highest time for achieving the LB, as they perform many handovers and replacement processes in addition to the complexity of the MA, which requires more processing time to the maneuvering between the CZ and the WZ to reach the required balance.

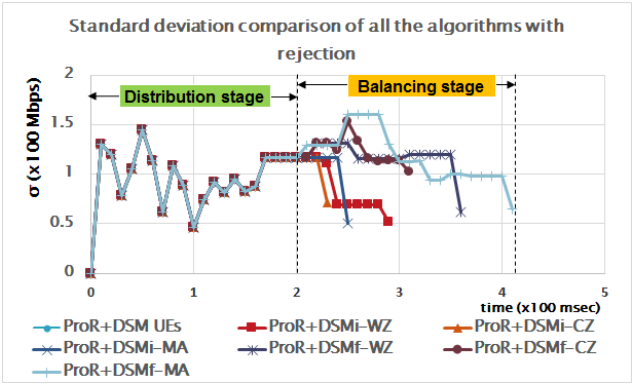


Fig. 5. Standard deviation (σ) for all the algorithms with ProR.

Moreover, the Pro needs more time to balance the load than the ProR, as this latter rejects the extra UEs and does not distribute them to the available APs. We also notice that the ProR without considering the constrained UEs (DSM UEs) shows the smallest value of the standard deviation, while the Pro without DSM UEs leads to the worst load distribution. In fact, the Pro distributes the new UEs similar to the ProR; however, the extra incoming UEs, which are not rejected when the Pro is used, deteriorate the LB process across the small cells.

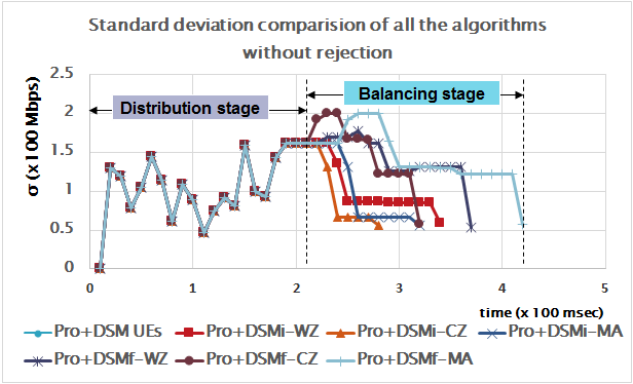


Fig. 6. Standard deviation (σ) for all the algorithms with Pro.

Furthermore, Fig. 7 clarifies that the ProR without DSM UEs improves the load distribution compared to the reactive algorithms (the average value of σ for the CZ, WZ and MA algorithms) by 34.97%. Indeed, the worst algorithm among the reactive algorithms is the CZ algorithm, since only the UEs located in the CZ can be handed-over with the objective of the LB.

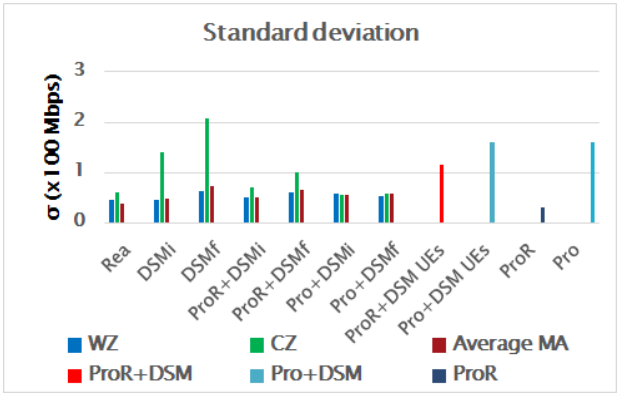


Fig. 7. Standard deviation (σ) for all the algorithms.

By considering the DSMA, the load distribution reduces compared to the reactive algorithms, because the DSM UEs are excluded from any handover, if this handover will increase the number of hops. Subsequently, this forces the DSMA to choose an UE from the second order at the price of the quality of the load distribution. For this reason, the ProR without DSM UEs and the reactive algorithms distribute the load among the small cells better than the DSMi or the DSMf. Conversely, when the ProR or the Pro is used with the DSMA (ProR&DSM or Pro&DSM), they enhance the load distribution by 42.55% and 65.12% compared to the ProR and Pro with DSM UEs, respectively. In fact, the DSMA, which is applied after the ProR or Pro, will redistribute the load across the APs and improve it again. Note that the load distribution achieved by the Pro with DSMA is better by 15.75% than that with the ProR, because the ProR rejects some UEs that may be served later as BCs. Besides, the DSMi improves the load distribution better than the DSMf, as the DSMi focuses on the LB more than reducing the inter-communications between the APs. For the same reason, the load distribution achieved by the ProR&DSMi outperforms the one performed by the ProR&DSMf by 24.49%. It is important to note that similar load distribution outcomes are obtained based on the Jain's fairness index (β).

With regard to the *BIR*, Fig. 8 reveals that the reactive algorithms show a *BIR* better than the other algorithms with the DSMA, because of the constrained UEs. In other words, the reactive algorithms achieve

the highest *BIR* of 89.16%. In contrast, the *BIR* achieved by the DSMA with the ProR or the Pro is less than that of the DSMA, as the load distribution performed by the ProR&DSM or the Pro&DSM is already distributed well and there is no need to improve the balance more. Moreover, the Pro&DSM leads to a *BIR* better than that in the ProR&DSM, because the latter rejects some UEs that may be used later as BCs.

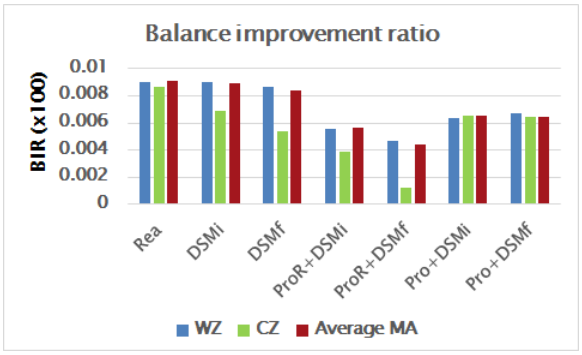


Fig. 8. Balance improvement ratio (*BIR*) for all the algorithms.

To determine the best LBA, the signaling load achieved by each algorithm is considered. Fig. 9 clarifies the *HOR* for the reactive algorithms and the DSMA, the *PR* for ProR and the *RR* for the DSMf. We notice that the DSMf requires more signaling than the DSMi, as the DSMf replaces the DSM UEs in addition to the handover procedures. Note that each replacement process requires two handovers. Additionally, the ProR without DSM UEs rejects UEs more than the ProR with DSM UEs. Although the ProR with DSM UEs can slightly overload the target cell during the distribution stage; however, the LB will be achieved again by either DSMf or DSMi during the balancing stage. Furthermore, the ProR&DSMi/f requires signaling more than the Pro&DSMi/f, as the latter does not reject any UE. As a result, the highest signaling load is caused by the ProR&DSMf.

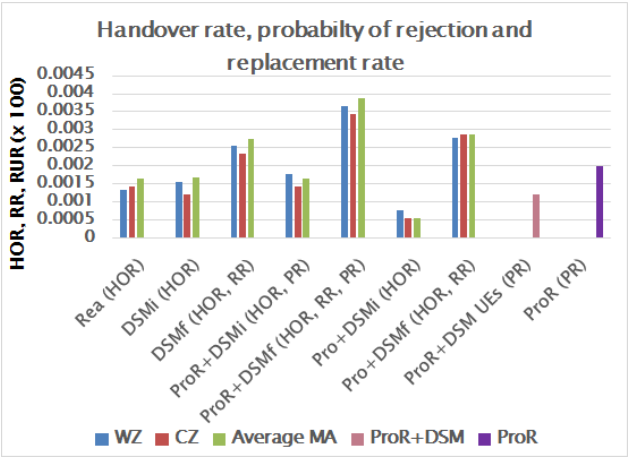


Fig. 9. *HOR*, *RR* and *PR* for all the algorithms.

Regarding the *BE*, the Pro&DSMi achieves the best *BE*, since this algorithm does not require so much signaling compared to other algorithms, as depicted in Fig. 10. In contrast, the worst *BE* is observed in the case of the Pro with or without DSM UEs. Moreover, the DSMi with or without Pro/ ProR demonstrates a *BE* better than that in the case of the DSMf with or without Pro/ProR, respectively. The reason is again that the DSMi does not replace the UEs like the DSMf.

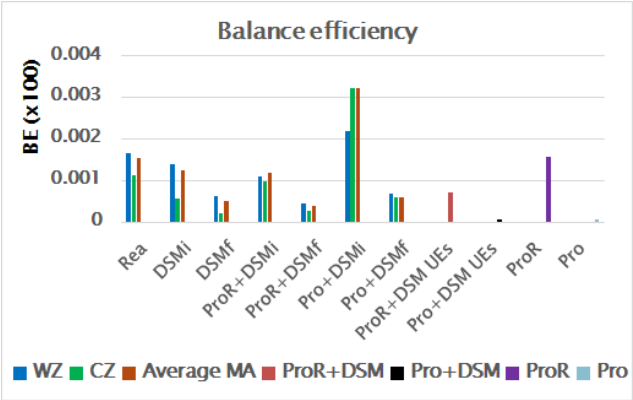


Fig. 10. Balance efficiency (*BE*) for all the algorithms.

Finally, concerning the RICR % that is carried out by the DSM method, Fig. 11 shows that the DSMf significantly reduces the inter-communications between the APs. The best RICR is achieved using the Pro&DSMf, which reaches 60.60%. In fact, since the Pro does not reject any UE, this increases the probability of finding the BCs for replacement. Furthermore, the Pro&DSMf performs a RICR higher by 26.25% and 19.21% than that in the case of the DSMf and ProR&DSMf, respectively.

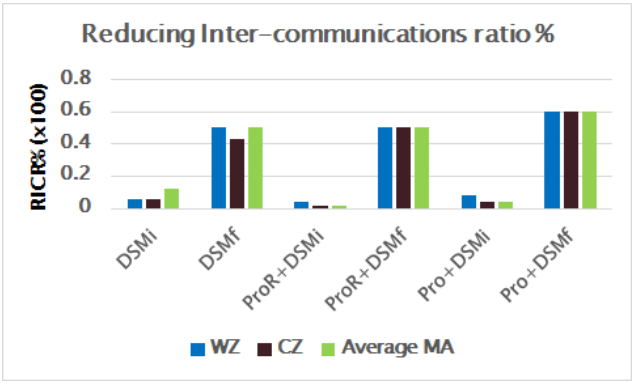


Fig. 11. *RICR* % for all the algorithms.

As a conclusion, if the LB has higher priority than reducing the inter-communications between the APs, the Pro&DSMi or the ProR would be two promoting solutions. Conversely, if the desired goal is mainly to reduce the inter-communications between the APs in addition to the LB across the cells, the Pro&DSMf using the WZA would be the best solution.

8. Conclusion

In this paper, several algorithms are proposed to balance the load across the small cells in UDN networks. A proactive algorithm with user rejection (ProR) distributes the new incoming users to the APs and rejects the extra-unconstrained users. A proactive algorithm without user rejection (Pro) distributes the new users to the small cells even if this slightly overloads the APs. On the other hand, to balance the load and reduce the inter-communications between the APs at the same time, the design structure matrix (DSM) method is suggested in this paper. In this context, we found that without considering the DSM method, the ProR improves the balance efficiency (*BE*) by 9.09% compared to the previous reactive algorithms. However, when the DSM method is considered, two DSM algorithms can be accompanied. If the load balancing (LB) is more important than reducing the inter-communications between the APs (*RICR*), the Pro&DSMi would achieve the best *BE*. On the contrary, if the *RICR* has higher priority than the LB, the Pro&DSMf, using the

worst zone approach, would lead to the highest *RICR*. Ongoing works is dealing with studying the impact of the small-cell layout of the cluster on the LB results. The preliminary results indicate that the intersecting small cells model adopted in this paper would achieve better LB than other small-cell cluster layouts.

References

[1] Ge, X., Tu, S., Mao, G., Wang, C., & Han, T. (2016, February). 5g ultra-dense cellular networks. *IEEE Wireless Commun., 23(1)*, 72–79.

[2] Tyson, B. R. (2001). Applying the design structure matrix to system decomposition and integration problems: A review and new directions. *IEEE Transactions on Engineering Management, 48*, 292–306.

[3] Evolved Universal Terrestrial Radio Access Network (E-UTRAN). (2010, Sep.). Self-configuring and self-optimizing network (SON) use cases and solutions, document TS 36.902. 3rd Generation Partnership Project.

[4] Feng, S., & Seidel, E. (2008, May). Self-organizing networks (SON) in 3GPP long term evolution. *Newsletter, Nomor Research GmbH*. Munich, Germany, Tech. Rep.

[5] Balachandran, A., Bahl, P., & Voelker, G. M. (2002). Hotspot congestion relief and service guarantees in public-area wireless networks. *ACM SIGCOMM Computer Communication Review, 32(1)*, 59-59.

[6] Aleo, V. (2003, March). *Load Distribution in IEEE 802.11 Cells*. Master of science thesis, KTH, Royal Institute of Technology.

[7] Patra, S. S. M., Roy, K., Banerjee, S., *et al*. (2006). Improved genetic algorithm for channel allocation with channel borrowing in mobile computing. *IEEE Transactions on Mobile Computing, 5(7)*, 884-892.

[8] Hanly, S. V. (1995). An algorithm for combined cell-site selection and power control to maximize cellular spread spectrum capacity. *IEEE Journal on Selected Areas in Communications, (7)13*, 1332-1340.

[9] Xu, H., He, Z., & Zhou, X. (2015). Load balancing algorithm of ultra-dense networks: A stochastic differential game based scheme. *KSII Transactions on Internet and Information Systems, 9(7)*, 2452-2467.

[10] He, Z., & Wang, J. (2014). Non-cooperative differential game based load balancing algorithm in radio-over-Fiber system. *IEEE, 11(2)*, 79-85.

[11] Zia, N., & Mitschele-Thiel, A. (2013, November). Self-organized neighborhood mobility load balancing for LTE networks. *Proceedings of the IFIP WD*.

[12] Huang, Z., Liu, J., Shen, Q., Wu, J., & Gan, X. (2015, March). A threshold-based multi-traffic load balance mechanism in LTE-A networks. *Proceedings of IEEE Wireless Commun. Netw. Conf. (WCNC)* (pp. 1273–1278).

[13] Hasan, M. M., Kwon, S., & Na, J. H. (2018, April). Adaptive mobility load balancing algorithm for LTE small-cell networks. *IEEE Trans. Wireless Commun., 17(4)*, 2205–2217.

[14] Salhani, M., & Liinaharja, M. (2018). Load balancing algorithm within the small cells of heterogeneous UDN networks: Mathematical proofs. *Journal of Communications, 13(11)*, 627-634.

[15] Evolved Universal Terrestrial Radio Access Network (E-UTRAN). (2014 t, Sep.). X2 Application Protocol (X2AP), document TS 36.423. 3rd Generation Partnership Projec.

[16] Evolved universal terrestrial radio access network (E-UTRAN). S1 application protocol (S1AP). 3rd Generation Partnership Project (3GPP), TS 36.413.

[17] Evolved universal terrestrial radio access (E-UTRA). Radio resource control (RRC). Protocol specification. 3rd Generation Partnership Project (3GPP), TS 36.331.

[18] Dimou, K., Wang, M., Yang, Y., Kazmi, M., Larmo, A., Pettersson, J., Muller, W., & Timner, Y. (2009). Handover within 3gpp lte: Design principles and performance. *Proceedings of the IEEE VTC.*

[19] Huang, M., Feng, S., & Chen, J. (2014, June). A practical approach for load balancing in LTE networks.

Journal of Communications, 9(6), 490-497.

[20] Ding, M., Lopez-Perez, D., & Mao, G. (2017, April). A new capacity scaling law in ultra-dense networks. arXiv: 1704.00399V1 [cs.NI].

[21] Kela, P. (2017). *Continuous Ultra-Dense Networks: A System Level Design for Urban Outdoor Deployments* (pp. 1799-4942). Publication series doctoral dissertations 86, Aalto University.

[22] Ley-Bosch, C., Medina-Sosa, R., González, I. A., & Rodríguez, D. S. (2015). Implementing an IEEE 802.15.7 physical layer simulation model with OMNET++. *Proceedings of the 12th Int. Conf. Distrib. Comput. Artif. Intell*. (pp. 251–258).

[23] Andersen, J. B., Rappaport, T. S., & Yoshida, S. (1995, Jan.). Propagation measurements and models for wireless communications channels. *IEEE Commun. Mag., 33(1)*, 42–49.

[24] Ruiz-Avilés J. M., *et al*. (2012, Oct.). Design of a computationally efficient dynamic system-level simulator for enterprise LTE femtocell scenarios. *J. Electr. Comput. Eng., 2012(802606)*.

Mohamad Salhani is an associate professor at the Department of Computer and Automation Engineering (CAE), Faculty of Mechanical and Electrical Engineering (FMEE), Damascus University since 2016. He received his B.S degree in electrical engineering. from the FMEE in 2000, M.Sc degree from National Polytechnic Institute of Lorain (INPL), France in 2005 and Ph.D degree from National Polytechnic Institute of Toulouse (INPT), France in 2008. He was an assistant professor at the CAE, FMEE in Damascus University in 2009. In 2016, he was a vice-dean for Administrative and Scientific Affairs at the Applied Faculty, Damascus University. He is currently a visiting professor at the Department of Communications and Networking, School of Electrical Engineering, Aalto University, Espoo, Finland. His research interests include 5G mobile communication systems, Ultra-dense networks (UDNs), Internet of Things and LoRa technology.