Using the Proactive Algorithms and the User Transfer Algorithms for Load Balancing in Ultra-Dense Networks

Mohamad Salhani
Department of Communications and Networking
Aalto University
Espoo, Finland
e-mail: mohamad.salhani@aalto.fi

Abstract—**Ultra-Dense Networks (UDNs) were introduced to improve the network coverage and support high data rate services. However, the dense deployment of small cells generates an uneven traffic distribution. The unbalanced load causes performance degradation and may be responsible for radio link failures. To address this problem, this paper proposes proactive algorithms to balance the load across the small cells based on the previous user transfer and the reactive algorithms. The proactive algorithms distribute the users, one by one, to the access points, while the reactive ones are only triggered when the load of the chosen small-cell cluster reaches a predefined threshold. The user transfer algorithms offload the small cells by transferring the extra users to the macrocells. The user transfer can be occurred before or after balancing the load by the reactive algorithms. The results indicate that the transfer_after algorithm improves the load distribution and the balance efficiency better than the proactive algorithm with the transfer_before algorithm by 3.46% and 15.71%, respectively.**

Keywords-UDN; load balancing; proactive algorithms; user transfer algorithms; reative algorithms.

I. INTRODUCTION

The rapid growth of traffic in the coming years will cause macrocell networks to evolve, becoming more tightly packed and eventually ultra-dense. To support the data demand for mobile broadband services and increase network capacity as well, the small cells will play an important role in the future 5G network and can significantly increase the capacity and throughput of the network [1]. Due to the low cost of the small cells, subscribers may have their own small cells and deploy them anywhere, even to turn on and off at any time. Therefore, the small cells will be mostly randomly distributed throughout the network [2]. Since the small cells have low transmission power, only a few users can be served by each small cell, and the mobility of users leads to an unbalanced load across the network. In addition, the preference of small cells during cell selection and reselection loads more traffic onto them; this also causes an overloaded network. When users move onto overloaded small cells, the deficit in resources results in handover failures or poor Quality of Service (QoS) [2]. Hence, some small cells do not satisfy the QoS requirements, while other neighboring small cells resources remain unused.

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 user (UE) experience. Thus, it is necessary to study a Load-Balancing Algorithm (LBA) that can adapt to various network environments and avoid the load ping-pongs.

The rest of this paper is organized as follows: Second II presents the related work. Section III describes the system model. The different LBAs are explained in Section IV followed by the performance evaluation in Section V. Section VI concludes the paper.

II. RELATED WORK

Researchers have proposed several solutions to address the LB problem and enhance cellular network performance. The authors in [5] proposed an MLB algorithm considering constant-traffic users 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 [6], a traffic-variant users LBA has been proposed considering small cells; however, this algorithm also considered a fixed threshold to identify the overloaded cells. In [2], the authors proposed an MLB algorithm considering an adaptive threshold to decide overloaded cells in a small cell network. The algorithm estimates the loads in both overloaded cells and neighboring cells, and achieves handovers based on the measurements reported by users.

The authors in [7] 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 aimed first to determine the best overlapping zone among several overlapping zones and then, to select the Best Candidate user (BC) for handover in order to reduce the number of the handovers and improve the network performance. However, the proposed algorithm was reactive; it is only executed when the user density of the chosen small-cell cluster reaches a predefined threshold.

On the other hand, the load balancing by transferring users has not been highlighted enough in the recent studies. Elgendi *et al* [8] have proposed new schemes to find the optimal number of sessions to be transferred from Unlicensed Long Term Evolution (U-LTE) networks to Licensed Long Term Evolution (L-LTE) or Wi-Fi networks. They have shown that it is possible to transfer the users from programmable Base Stations (BSs) to Access Points (APs) in order to achieve a win-win outcome for both networks. Nonetheless, they have focused on the users' velocity and the distance between the user and the BS more than the data offloading. Besides, the proposed schemes have transferred a higher number of users. In contrast, the authors in [9] have proposed three user transfer algorithms to offload the small cells of UDN networks by transferring the extra users to the macrocells. They first identify the best overlapping zone among the overlapping zones and then, the BC is handed over to another AP or transferred to the BS by selective way. The results indicated that these algorithms can improve the performance of the whole UDN network.

The authors in [10] have proposed proactive LBA by initiating vertical handovers before admitting call if network resources are not substantial; however, the proactivity only concerns user-cell association policy and lacks consideration of users' mobility and content demands. Moreover, a novel proactive LB scheme was suggested in [11]. The proposed framework learns users' mobility and demands statistics jointly to proactively cache future contents during their stay at lightly loaded cells. The results indicated an improvement in the quality of experience and the load distribution compared to the state-of-the art reactive schemes.

In this paper, we propose proactive algorithms that construct clusters of the small cells and perform the LB across the APs. The proposed proactive algorithms are always on standby and ready to be triggered for distributing the new users to the small cells. To improve the LB, the proactive algorithms are followed by the transfer algorithms and the reactive algorithms, which have been proposed in [7] [9]. A comparison between all the algorithms will be achieved to figure out the best LBA. For cluster formation, we consider 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.

III. SYSTEM MODEL

In this section, the system model is described and then, the measurement of the small cell load is clarified. After that, we explain the handover procedure.

A. *System description*

We consider a heterogeneous LTE network composed of a set of macro cells (evolved Node B (eNB)) and small cells (APs), *N*, and a set of users, *U*, as done in [2] [7]. We consider the UDN small cells with overlapping zones (Z_1, Z_2, Z_3 and Z_4) and each set of small cells constitutes a cluster. The LB is achieved in the small-cell clusters.

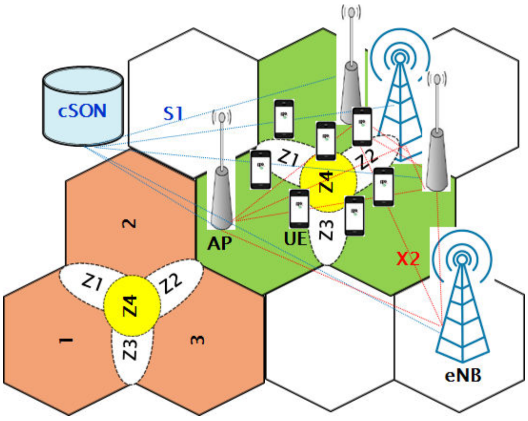

Figure 1. System model with a cSON.

In the simulation model, we considered a cluster consists of three intersecting small cells [7] [9], as depicted in Figure 1. The cells interconnect with each other via X_2 interface. This allows them to perform the needed functionalities such as handovers, load management, and so on [12]. Therefore, the users 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 S_1 interface [13]. The cSON subsystem collects the required load-related information from the network and optimizes the parameters of the cells to perform the LB process.

B. *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 [2]. For a given time duration, *T,* the small cell load, ρ_i, of cell *i* at time *t*, is given as

$$\rho_i^t = \frac{1}{T.N_{PRB}} \sum_{\tau \in (t-T,\ t)} RB_i^\tau \tag{1}$$

where N_{PRB} and RB_i, denote the total *PRBs* and the total allocated *PRBs* at time τ in cell *i*, respectively. Hence, the Average Cluster Load, *ACL*, is calculated as

$$ACL = (\textstyle\sum_{i=1}^{m} \rho_i)/m \tag{2}$$

where *m* is the 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 [7] [9] 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 width of the balance zone. A small value of *α* requires many handovers to reach the needed LB, and vice-versa. In this paper, *α* is set to 0.05, as done in [7] [9]. Equation (3) and (4) show that the thresholds are a function of *ACL* and *α*.

C. *Handover procedure*

In this paper, A3 and A4 event measurements are used to trigger a handover and select the users candidate for handovers, and the Reference Signal Received Power (RSRP) is assumed reporting signal quality for measurements, as done in [2] [14]. Actually, event A3 is widely used for triggering handovers in wireless networks [15]. In that way, event A3 is triggered and the users 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 [2]:

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

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 users in the overloaded cells to the underloaded cells. However, to balance the load, the system needs information about the edge-users distribution. For that, the event A4 is used. All the cells share the users' information with the cSON. The condition for triggering the event A4 is expressed as [2],

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

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

IV. LOAD BALANCING ALGORITHMS

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

A. *Proactive algorithm with (user) Rejection (ProR)*

The Proactive algorithm with (user) Rejection (ProR) distributes the new users to the covering APs and rejects the extra users, as depicted in *Algorithm 1*. This algorithm is always on standby and ready to be triggered each time a new user enters the network. For each new user, the algorithm selects the best AP, which has the least load. In the ProR, the resources of the APs are considered limited; each AP has a maximum capacity, ρ_{th}. Therefore, when an AP is selected to include a new user and the load of this AP, ρ_i will not exceed ρ_{th} if it admits this user, thus the user is accepted. Otherwise, the ProR rejects the user. The distribution process is achieved for each new user moves onto the network until the user density, *D* of the chosen cluster reaches the user density threshold, D_{th}.

Algorithm 1: Proactive algorithm with Rejection (ProR)

```
1: Get RSRP and PRB measurements of UE j and cell i, $D_{th}$ and UE's zone
2: if $D < D_{th}$ then
3:   Find the cell that covers this UE and has the smallest $\rho_i$
4:   if $\rho_i < \delta_1$ and $(\rho_i + RBUR_j) > \rho_{th}$ 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 $\rho_i$ 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, $D_{th}$, and UE's
   zone,
2: if $D < D_{th}$ then
3:   Find the cell that covers this UE and has the smallest $\rho_i$
4:   Transfer the new UE to the target cell
5:   Update $\rho_i$ 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, $D_{th}$, UE's zone and
   $\alpha$
2: Find the cluster with the highest user density
3: if $D >= D_{th}$ then
4:   Calculate $\rho$ for each cell i, ACL, $\delta_1$ and $\delta_2$
5:     if one of the chosen cluster's cell has $\rho_i > \delta_1$ then
6:       Calculate $\beta_1$, $\beta_2$, $\beta_3$ and $\beta_4$, and then find the worst zone
7:       Apply the transfer policy
8:       Calculate $\Delta$ and determine the BCj
9:       if $\beta_{new} > \beta_{old}$ then
10:        Transfer the BCj to the target cell (achieve a handover)
11:        Update $\rho$ for each cell i and go to step 5
12:      else
13:        if there are UEs of 2nd order then
14:          Find the new BCj and execute a handover
15:          Update $\rho$ 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
```

B. *Proactive algorithm without (user) rejection (Pro)*

The Proactive algorithm without (user) rejection (Pro) is similar to the ProR, as depicted in *Algorithm 2*; however, the APs are considered having enough resources to accept the new users as long as the user density of the current cluster does not exceed D_{th}. In practice, the density condition is not necessary to be checked, as this algorithm triggers for each new user. This condition is only imposed in this work to compare the results of these two proactive algorithms to the reactive and transfer algorithms with the same user density.

C. *Reactive algorithms (rea)*

The reactive algorithm (rea) has been proposed in [7] to balance the load across the APs. Nevertheless, this algorithm is only triggered once the user density of the cluster reaches D_{th}. To achieve the reactive algorithm, the authors have suggested three approaches based on the overlapping zones concept. In the *Common Zone (CZ) approach,* the load is only balanced via the users that are located in the CZ between the three overlapping small cells; zone 4 (Z_4), as shown in in Figure 1. The second approach is the so-called *Worst Zone (WZ) approach*. The LB in this approach is only achieved 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 [7]. 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.

To achieve the LB, the reactive algorithm needs to identify the cluster with the highest density and then, it figures out the overlapping zone and the BC for handover. 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 D_{th}, the algorithm is stopped. Otherwise, the algorithm sets the user's load, $RBUR_j$ of each $user_j$, 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 a user (overloaded AP) and which one must include this user (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 users and handed-over users from an overloaded AP. A balanced AP can only accept new users, while an overloaded AP does not receive any new or handed-over users. 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) [16] as

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

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 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 users 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 user's load and Δ has the smallest absolute value as follows

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

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 so, 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 again if there is still an overloaded AP, and also if the balance improvement is still valid. If so, the LB enhancement is evaluated again in the new target zone and so on. Otherwise, the algorithm waits for the next trigger.

D. Reactive and user transfer algorithms (rea&transfer)

The reactive algorithm with the transfer algorithms are combined (rea&transfer) in this paper in order to compare them to the proactive algorithms. In order to transfer the users from the small cells to the macrocells, two transfer algorithms are suggested as follows:

1) Transfer_After Algorithm (TAA)

The Transfer_After Algorithm (TAA) takes care of the users that should be transferred to the macrocells. This algorithm is composed of two stages. The first one is the balance stage achieved by the reactive algorithm. The second is the transfer stage, which is carried out after the balance stage. Therefore, the TAA has the same first steps of the reactive algorithm; however, when there are no more balance improvements, the transfer stage with new selection and transfer policies are initialized. In the first step of the transfer stage, the algorithm checks if at least one of the APs is overloaded, i.e., its load exceeds the ρ_{th}. If not, the algorithm is stopped. Otherwise, the second step is to achieve the new selection policy in order to determine the BC to be transferred as follows. First, the algorithm calculates the new delta as a difference between the most overloaded AP and ρ_{th} as follows,

$$\Delta_{new} = \rho_{most\_overloaded\_AP} - \rho_{th} \tag{10}$$

Second, the best candidate value $BC_{(j)}$ is calculated for each user connected to the selected AP as a difference between the user load and the new delta as follows:

$$BC_{(j)} = \mathrm{RBUR}_{(j)} - \Delta_{new} \tag{11}$$

Of all the users connected to the AP in question, the BC is the one for which the $BC_{(j)}$ has the smallest positive value. Otherwise, the BC is the one that has the smallest negative value, if all the values of $BC_{(j)}$ are negative. The transfer from the chosen AP is repeated until the AP load becomes less than or equal to ρ_{th}. In the third step, the algorithm determines the next most overloaded AP and repeats the second step. When all the APs have checked and there is no more users for transfer, the TAA waits for the next trigger.

2) *Transfer_Before Algorithm (TBA)*

The Transfer_Before Algorithm (TBA) is similar to the TAA; however, the transfer stage is initialized as a first step for each AP's load exceeding ρ_{th}. Once the loads of all APs do not exceed ρ_{th} anymore or if there are no more available users to be transferred, the balance stage starts calling the reactive algorithm to continue the LB task as usual.

E. *Proactive algorithms with the transfer and the reactive algorithms (Pro&transfer&rea)*

In this case, the proactive algorithms are integrated with the transfer algorithms and the reactive algorithms. This means that first the Pro distributes the users to the small cells. After that, the TBA transfers the extra users to the macrocells before applying the reactive algorithms, i.e., Pro&before&rea. Instead, the TAA transfers the extra users to the macrocells after balancing the load by the reactive algorithms, i.e., Pro&after&rea. Note that the ProR does not need to be followed by the transfer algorithms or the reactive algorithms. It is itself able to balance the load without any help form these two algorithms at the price of higher rejected users.

V. PERFORMANCE EVALUATION

In the following section, we present the simulation environments and the performance evaluation metrics. Then, the simulation results are analyzed.

A. *Simulation environments*

We performed the simulation with a heterogeneous network with macro and small cells using *ns-3*. 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 users per small cell. Therefore, the density threshold, D_{th} is equal to 18 users per cluster, as considered in [7] [9]. The users allocate multi-traffic. Each user selects a specific bit rate in the range of 0 to 350 Mbps [7] [17]. 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 users' distribution, 50% of the mobile users were randomly distributed over the whole area, and the rest were fixed and uniformly distributed over the border areas of the small cells, as listed in Table I, because the reactive algorithms hand over the users located in the overlapping zones.

TABLE I. 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	$PL=147.4+43.3log_{10}(R)$
Fading	Standard deviation 4 dB, lognormal
Resource scheduling	CQA scheduler
CIO_{min} *and* CIO_{max}	-6dB, 6dB
Hysteresis	2 dB
ρ_{th}	1Gbps
BS capacity	2Gbps
D_{th}	18 user
User velocity	3.6 km/h
Mobility model	Uniform, 50% CW mobility users and 50% static users

The randomly distributed users follow the Circular Way (CW) mobility model [2] [18]. In this mobility model, the users 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 [2] [19]. To allocate the *PRBs* among the users in a cell, a Channel QoS-Aware (CQA) scheduler was adopted [2] [20].

B. *Performance evaluation metrics*

To evaluate the performance, we considered three aspects: the load distribution across the small cells, the Balance Improvement Ratio (BIR) and the Balance Efficiency (BE). 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 [7] [9],

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

where $\sigma_{initial}$ and σ_{final} are the standard deviation of the small cells loads before and after applying the LBA, respectively.
We also considered the signaling load, which is; the handover rate, HOR for the reactive algorithms, the probability of rejection (call drop rate) of the new users from the APs, PR_AP for the ProR, and the probability of rejection from the BSs and the transfer rate, PR_BS and TR for the transfer algorithms.
The BE is measured by taking into account the standard deviation and the signaling load for each algorithm [7] [9]. 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

$$BE_{\mathrm{Pr}oR} = 1/(\sigma_{final} \times PR\_AP) \tag{14}$$

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

Considering the transfer algorithms with or without the Pro, the BE is given by

$$BE_{\mathrm{Pr}o+transfer} = 1/(\sigma_{final} \times (HOR + TR + PR\_BS)) \tag{16}$$

C. *Results analysis*

To analyze the results and evaluate the performance of the different algorithms, we compare the results of the proposed combination of the proactive and transfer algorithms to the previous reactive and transfer algorithms suggested in [7] [9]. Figure 2 shows the standard deviation of the small cells loads for the different algorithms. We notice that the Pro&before&rea distributes the load across the APs better than the other algorithms except the TAA (after&rea). However, the load distribution achieved by the TAA is better only by 3.46%. Moreover, the best load distribution is achieved by the TAA using the MA. In total, the load distribution performed by the Pro&transfer&rea (the average value of the Pro&after&rea and the Pro&before&rea) outperforms the proactive algorithms (the average value of the Pro and the ProR), the transfer algorithms and the reactive algorithms by 66.62%, 22.38% and 39.17%, respectively. This demonstrates the importance of combining the proactive algorithms with the transfer algorithms and the reactive algorithms to balance the load. On the contrary, the Pro leads to the worst load distribution. In fact, the Pro distributes the new users to the APs similar to the ProR; however, the incoming users, which are not rejected when the Pro is applied, will deteriorate the LB. We also found that if the Pro&before was not followed by the reactive algorithm, the load distribution will be worse than the Pro&before&rea by 20.14%. This clearly illustrates the importance of the reactive algorithm for the transfer algorithms. Note that similar load distribution results are obtained based on the Jain's fairness index, β.

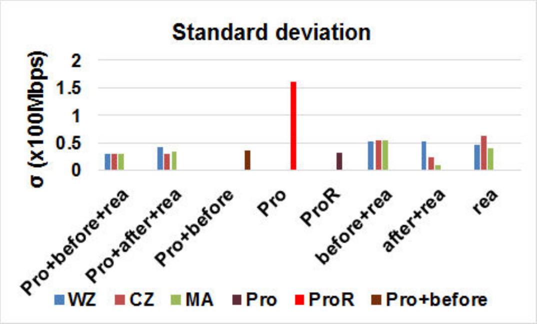

Figure 2. Standard deviation (σ) for the different algorithms.

With regard to the BIR, Figure 3 demonstrates that the best BIR is accomplished by the transfer algorithms (TAA and TBA), which is better than the Pro&transfer&rea by 13.41%. This is because the load distribution performed by the Pro&transfer&rea outperforms the one achieved by the transfer algorithms by 22.38% and then, the Pro&transfer&rea does not need to improve the balance more. For the same reason, the BIR of the reactive algorithms is higher than the Pro&transfer&rea by 11.40%.

In order to determine the best LBA, the signaling load caused by each algorithm is considered, as depicted in Figure 4. We observe that the TAA leads to the highest signaling load. This algorithm requires more signaling than the Pro&before&rea by 42.53%. In contrast, the ProR shows the smallest signaling load compared to the Pro&transfer&rea and the Pro&before, as the ProR does not achieve any handover or transfer processes. On the country, this algorithm rejects the highest rate of the new users from the APs. This rejection rate reaches 20%. In addition, the Pro&before&rea requires signaling higher than the Pro&before only by 5%, which is the value of the HOR achieved by the reactive algorithms.

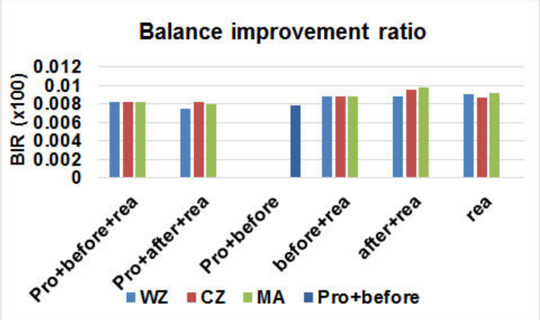

Figure 3. The balance improvement ratio for the different algorithms.

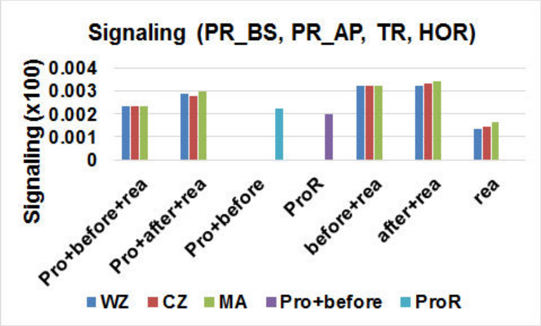

Figure 4. Signaling load for the different algorithms

With respect to the BE, we figured out that the BE of the Pro&before&rea is better than the Pro&before by 14.42%, as shown in Figure 5. This clarifies the importance for the Pro&before to be followed by the reactive algorithms. Additionally, the BE of the TAA outperforms the Pro&before&rea by 15.71%. Furthermore, the BE of the ProR and the Pro&before&rea is similar, but with a PR_AP of 20% for the ProR against a PR_BS of only 1.11% for the Pro&before&rea. However, the PR_BS of the TAA is only 0.81%. Alternatively, the signaling load caused by the TAA is higher by 42.53% than the Pro&before&rea.

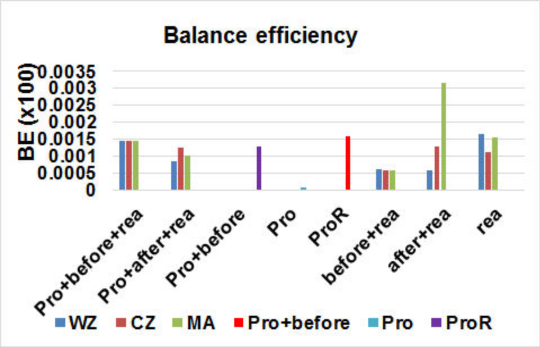

Figure 5. The balance efficiency for the different algorithms.

As a result, the ProR is a refused choice, since it leads to the highest rejection rate of all the algorithms. Thus, the Pro&before&rea and the TAA would be two promoting solutions to balance the load in UDN networks.

VI. CONCLUSION

In this paper, several load-balancing algorithms are proposed for balancing the load in UDN networks. The proactive algorithms distribute the new users, user by user, to the small cells. This can occur with or without rejecting the extra users that overload the target small cells. The user transfer algorithms can offload the small cells before or after balancing the load by the reactive algorithms. The proposed proactive algorithms with user transfer algorithms and the reactive algorithms are compared to the previous user transfer algorithm and the reactive algorithms. As a result, two promoting solutions would be used to balance the load in UDN networks. The first solution would be the transfer_after algorithm using the mixed algorithm with a probability of users rejected from the macrocells of 0.22%; however with a signaling load higher by 42.53% than the proactive algorithm with transfer_before algorithm and the worst zone algorithm. The second solution would be the proactive algorithm with the transfer_before algorithm and the worst zone algorithm with a probability of rejection from the macrocells of 1.11% and a balance efficiency smaller than that with the transfer_after algorithm by 15.17%. Future works will deal with integrating the Design Structure Matrix (DSM) method with the proposed algorithms. This would load the balance among the small cells and reduce the APs inter-communications at the same time.

REFERENCES

[1] J. Hoadley and P. Maveddat, “Enabling small cell deployment with HetNet,” IEEE Wireless Commun., vol. 19, no. 2, pp. 4–5, Apr. 2012.

[2] M . M. Hasan, S. Kwon, and J. H. Na, “Adaptive mobility load balancing algorithm for LTE small-cell networks,” IEEE Trans. Wireless Commun., vol. 17, no. 4, pp. 2205–2217, Apr 2018.

[3] Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-Configuring and Self-Optimizing Network (SON) Use Cases and Solutions, document TS 36.902, 3rd Generation Partnership Project, Sep. 2010.

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

[5] N. Zia and A. Mitschele-Thiel, “Self-organized neighborhood mobility load balancing for LTE networks,” in Proc. IFIP WD, pp.1-6, 13-15 Nov 2013.

[6] Z. Huang, J. Liu, Q. Shen, J. Wu, and X. Gan, “A threshold-based multi-traffic load balance mechanism in LTE-A networks,” in Proc. IEEE Wireless Commun. Netw. Conf. (WCNC), pp. 1273–1278, Mar. 2015.

[7] M. Salhani, and M. Liinaharja, “Load balancing algorithm within the small cells of heterogeneous UDN networks : Mathematical proofs,” Jounrnal of Communications, vol. 13 , no. 11 , pp. 627-634, 2018.

[8] I. Elgendi, K. S. Munasinghe and A. Jamalipour, "Traffic offloading for 5G: L-LTE or Wi-Fi," 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Atlanta, pp. 748-753, 2017.

[9] M. Salhani and M. Liinaharja, “Load Migration Mechanism in Ultra-Dense Networks,”. In Proceedings of the 2nd International Conference on Telecommunications and Communication Engineering (ICTCE 2018). ACM, New York, NY, USA, pp. 268-274, 2018.

[10] D. Ma, and M. Ma, “Proactive load balancing with admission control for heterogeneous overlay networks. Wireless Communications and Mobile Computing,” 13(18), pp.1671-1680, 2013.

[11] S. Manzoor, et al., “Leveraging mobility and content caching for proactive load balancing in heterogeneous cellular networks,” Transactions on Emerging Telecommunications Technologies, p.e3739, 2019.

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

[13] “Evolved universal terrestrial radio access network (E-UTRAN); S1 application protocol (S1AP),” 3rd Generation Partnership Project (3GPP), TS 36.413, Sep. 2014.

[14] “Evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC); protocol specification,” 3rd Generation Partnership Project (3GPP), TS 36.331, Jan. 2016.

[15] K. Dimou, et al., “Handover within 3GPP LTE: Design Principles and Performance”, VTC 2009-Fall, 2009 IEEE 70th, pp.1-5, 20-23 Sept. 2009.

[16] M. Huang, S. Feng, and J. Chen, “A Practical Approach for Load balancing in LTE Networks,” Journal of Communications Vol. 9, No. 6, pp. 490-497, June 2014.

[17] P. Kela, “Continuous Ultra-Dense Networks, A System Level Design for Urban Outdoor Deployments,” book 1799-4942 (electronic), Aalto University publication series DOCTORAL DISSERTATIONS 86/2017.

[18] C. Ley-Bosch, R. Medina-Sosa, I. A. González, and D. S. Rodríguez, “Implementing an IEEE802.15.7 physical layer simulation model with OMNET++,” in Proc. 12th Int. Conf. Distrib. Comput. Artif. Intell., pp. 251–258, 2015.

[19] J. B. Andersen, T. S. Rappaport, and S. Yoshida, “Propagation measurements and models for wireless communications channels,” IEEE Commun. Mag., vol. 33, no. 1, pp. 42–49, Jan. 1995.

[20] J. M. Ruiz-Avilés, et al., “Design of a Computationally Efficient Dynamic System-Level Simulator for Enterprise LTE Femtocell Scenarios,” Journal of Electrical and Computer Engineering, vol. 2012, Article ID 802606, pp. 1–14, 2012.