Task Assignment Strategies for Pools of Baseband Computation Units in 4G Cellular Networks

Thomas Werthmann*, Heidrun Grob-Lipski†, Sebastian Scholz*, Bernd Haberland†
*Institute of Communication Networks and Computer Engineering, Universität Stuttgart, Stuttgart, Germany.
†Bell Labs, Alcatel-Lucent, Stuttgart, Germany
thomas.werthmann@ikr.uni-stuttgart.de, heidrun.grob-lipski@alcatel-lucent.com,
sebastian.scholz@ikr.uni-stuttgart.de, bernd.haberland@alcatel-lucent.com

***Abstract*—A promising architecture for future cellular mobile networks is to place remote radio heads on the cell towers and connect those via fibers with a centralized pool of baseband units. Among other things, the centralization of baseband computation facilitates multiplexing gains and can thereby save compute resources. To realize these gains, an efficient and load-balancing assignment of compute jobs to computation units is required. In contrast to approaches in literature, our architecture virtualizes the processing for each UE separately. It thereby provides a finer granularity and allows to newly decide the assignment of a compute job to a processing unit whenever a UE starts transmitting data. In this publication, we present multiple heuristics to decide this assignment. We compare the efficiency of the assignment and the perceived service quality realized by the heuristics with an ideal assignment and the classical static cell-based assignment. Our evaluation shows that by using a good assignment heuristic, about 50 % of the hardware resources can be saved.**

I. Introduction

The C-RAN (Cloud or Centralized Radio Access Network) was introduced by China Mobile Research Institute in 2010 [1] to initiate the architectural evolution of the currently deployed distributed base station infrastructure. It is based on the Wireless Network Cloud (WNC) concept from IBM [2] as a first assembly of IT and wireless network platforms.

The basic idea of C-RAN is to separate the Remote Radio Head (RRH) and the BaseBand Unit (BBU) by using interfaces like Common Public Radio Interface (CPRI) or Open Base Station Architecture Initiative (OBSAI) [3]. Within this architecture the RRHs provide the radio transmit and receive components like digital processing, frequency filtering and power amplification and the BBUs perform the centralized signal processing functionalities as modulation, coding and Fast Fourier Transformation (FFT) for several RRHs.

With C-RAN the number of conventional sites and hardware resources as well as backhaul connections can be reduced considerably [4]. Beyond this LTE-Advanced features like enhanced Inter-Cell Interference Coordination (eICIC), Carrier Aggregation and Coordinated Multi-Point (CoMP) will benefit from the centralized architecture.

The V-RAN (Virtualized-RAN) as a logical evolution of C-RAN virtualizes the BBU functionalities and dynamically allocates them to the virtual BBUs. Based on a C-RAN deployment V-RAN has the capability to further reduce capital and operating expenditures [5]. However, virtualization of wireless communication systems requires thorough planning due to the strict real-time processing requirements [6].

A. Contributions and Related Work

The virtual base station introduced in [7] constitutes the base concept of the investigations presented in this publication. This future V-RAN concept has been derived to support virtualization and flexible pooling on user respectively bearer granularity level. The authors identify the traffic load dependent functions per user or per radio bearer in uplink (UL) and downlink (DL) direction in an eNodeB for virtualization. These functions, herein after called User Processing (UP), comprise the S1 termination, Packet Data Convergence Protocol (PDCP), Radio Link Control (RLC), Medium Access Control (MAC) and user scheduling and the load depending functions of the physical layer (PHYuser). The physical layer cell functions (PHYcell: Framing/De-framing, inverse FFT/FFT, etc.) will not be virtualized.

The architecture comprises multiple RRHs, which are connected with high speed optical links to the associated multi-site BBU (MS-BBU). Such a MS-BBU covers several BBUs. Each RRH has a statically associated *Home-BBU*, which performs the PHYcell functions for the RRH's cell. A centralized controller decides to which BBU a UP of a newly arriving UE is assigned. If necessary, e.g., in case of overload, the architecture allows to reallocate a UP to a different BBU.

An earlier publication deals with multiplexing gains that can be achieved with the just explained V-RAN architecture [8]. Therein we describe a detailed multi-layer model of the virtual base station and present simulation results showing short-term pooling effects induced from user load distribution and traffic heterogeneity. We prove that traffic variations in time and area hold a considerable potential for multiplexing gains by pooling processing resources. The publication also derives how the multiplexing gain scales with the number of aggregated cells and investigates the influence of the spatial user distribution on the utilized compute resources.

Other architectures also benefit from the statistical multiplexing effect. In [9] the Colony-RAN network architecture is introduced. A BBU is able to connect to one or more RRHs in dependence of the traffic demand, by which the number of BBUs can be reduced enormously. A dynamic BBU to RRH mapping scheme based on the asymmetric DL/UL conditions in a Time Division Duplex (TDD) LTE system is presented in [10], [11]. This approach is combined with a novel interference coordination, solved with clustering for DL/UL partitioning. As these approaches focus on the interference in a TDD LTE system, they are not directly applicable to the Frequency Division Duplex (FDD) LTE system considered here.

978-1-4673-6305-1/15/$31.00 ©2015 IEEE

The publication [12] outlines the potential pooling gain when exploiting the variations of processing load across base stations. It presents a resource management framework for the trade-off between network quality and network operation costs. The authors in [13] analyze the statistical multiplexing gain and parameterize the network to maximize the potential cost savings. Their packet based architecture adapts to changing traffic conditions during the day. The optimized mix of cells with different traffic profiles and BBU pool positions leads to a reduced number of required BBUs and fibers. Also [14] evaluates the influence on the statistical multiplexing gain in a BBU pool and propose an architecture that is able to adapt to diurnal load patterns of cells. The architecture requires a packet based fronthaul, to change the relation between RRHs and BBU pools dynamically.

The referenced approaches do all concentrate on the assignment of RRHs or cells to BBUs. They partially allow to change these assignments during runtime of the system, which means that compute tasks have to be migrated.

In this paper we concentrate on the initial placement based on the V-RAN architecture as described above. We propose a heuristic for the assignment of UPs to computation units when the respective UE arrives at the system. Although in principle possible with our architecture, this allocation will not be changed afterwards. This distinguishes our approach from the other referenced approaches. The results presented in this paper are achieved without the need to migrate compute tasks. This saves implementation complexity. In addition, our system does not suffer from service interruptions caused by task migration, as e.g., observed by [15].

Based on a system simulation we demonstrate that by considering average processing load our advanced heuristic performs significantly better than a classical static allocation or a random placement. With some impact on the user experience the advanced heuristic delivers possible savings of 50 % of the compute resources.

B. Structure

In Section II, we start with the description of the system models comprising user and traffic model, radio network model, computation resource model, and the overload prevention mechanism. Then, Section III explains the initial placement strategies, starting with the baseline and an optimization. Then four heuristics will be introduced, which can be employed in reality for compute effort assignment. Section IV gives the overload results and the pooling gains occurring for the different placement heuristics and comparing them with the evaluation results from optimal placement. Further the user experience under limited compute resources in combination with the overload prevention mechanism is evaluated. Finally, we conclude the paper in Section V.

II. System Model

Our model needs to capture all influencing factors on the compute resource usage. Subsection II-A depicts the traffic load caused by the users. This load is carried by the radio network as explained in subsection II-B. The baseband processing effort caused in the radio network depends on the two preceding models and is defined in subsection II-C. Finally, in subsection II-D we define overload and describe how our system copes with this by allocating less frequency resources.

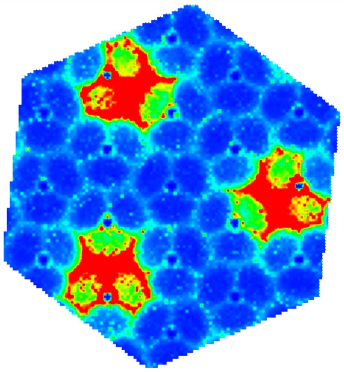

Fig. 1. Heat map of the active user density

We assume a 10 MHz LTE system. The macro base stations are placed in a hexagonal arrangement of 19 sites. Each base station supplies three sector cells, resulting in 57 sectors. We apply wrap-around to avoid border effects. In our evaluations, we concentrate on DL transmissions. Although UL also causes high computational effort at the base station, it behaves similar as the DL direction.

A. User and Traffic Model

The traffic model has a large influence on the resource usage. Opposed to a full buffer assumption, real Internet traffic is bursty and has a heavy-tailed object size distribution [16]. This directly influences the processing effort required to serve the UPs. Therefore it is important to have a proper model for the per-user traffic demands.

We model traffic as pairs of request and response objects. Requests are generated by the users and send to a server that reacts by sending the response. This covers many of today's Internet applications. The objects are transmitted as quickly as possible, i.e., there is no rate limitation introduced by the sender. For our scenario, we are interested in the effects at the network layer and below. Therefore, we idealize transport layer effects and assume that both, the request and response objects arrive as a whole at the BBU respectively User Equipment (UE) buffers. As we concentrate on the DL in this publication, the UL objects are not discussed further. Our model is based on the assumption that the network load is caused by a high number of independent users. The Inter-Arrival Time (IAT) of these request-response pairs follows a negative exponential distribution and is used to control the offered traffic in the system. We use an object size distribution measured on a campus link [16]. To avoid problems arising from very large objects, we clip the distribution at 10^8 bytes. Thereby, we cut off a part of the heavy tail of the distribution. However, objects above this size contribute only 0.7 % of the traffic volume.

In order to simulate changing user locations, each request originates from a new user with a new location. The users are placed with a probability of 50 % uniformly over the whole scenario. The other 50 % are placed in three hotspots which are located in the scenario so that the distance between the centers of the hotspots is maximized. Hotspots are defined by their center and two normal distributions with mean 0 and a

TABLE I. SYSTEM MODEL PARAMETERS

Property	Value
Cell layout	19 tri-sectorized sites, 500 m distance, wrap-around
BS TX power	46 dBm
BS / UE height	32 m / 1.5 m
Path-loss [dB]	$128.1 + 37.6 \cdot \log_{10} d\,[\mathrm{km}]$, from [18]
BS antenna model	3D, 15° tilt, from [18]
Shadowing	8 dB log-normal
UE velocity	0 km/h (for fast fading model: 3 km/h)
Carrier frequency	2 GHz
System bandwidth	10 MHz
Subframe duration (TTI)	1 ms

standard deviation of 180 m for the user coordinates relative to the centers of the hotspots. The resulting user density in our scenario is depicted in Figure 1. During transmission, the users do not move. After a user has finished his transmission, he leaves the system. Note that, as users with low channel quality need more time to transmit their requests, the density of active users is higher at the cell edge. We apply a simple Admission Control (AC), which drops arriving requests when there are more than 100 users active in the sector.

B. Radio Network Model

Besides the user and traffic model, the radio network model is important to determine the required compute resources for a cell. For the radio propagation, we consider path-loss and shadowing. The parameterization of the radio propagation is summarized in Table I and complies with 3GPP specifications. From the transmit power and the signal degradation between all active transmitters and the receiver as well as the noise level, we determine the mean Signal-to-Interference-and-Noise-Ratio (SINR) of a user.

With our system level simulation, we want to look at effects on time scales of hundreds of seconds. Therefore, due to the computational complexity, it is difficult to model multipath-propagation. Instead, we assign resources in a round-robin fashion and use the model in [17] to consider fast-fading and frequency-selective scheduling with the commonly known proportional fair scheduler. This model uses the number of active users and their respective mean SINR to determine an effective SINR diversity gain. With the enhanced SINR, we derive the possible rate on the channel according to LTE Modulation and Coding Schemes (MCSs). For this, we use Block Error Rate (BLER) tables generated from link layer simulations, including two Multiple-Input-Multiple-Output (MIMO) modes. Above an SINR of about 4 dB, we use 2x2 spatial multiplexing MIMO. At lower channel qualities, we apply Space-Frequency Block Coding (SFBC). We assume ideal channel knowledge at the base station and apply a target decode probability of 80 %. Failed transmissions are reinserted into the sending buffer after 8 ms.

C. Computation Resource Model

From the traffic and radio network models, we know which radio resources are actually in use and which transmission mode has been chosen. With this, we are able to determine the processing effort per UP with the computation resource model introduced in [8]. We concentrate on the PHYuser components of the UP, i.e., the compute resources for physical layer calculations which can be directly associated to a UE (Forward Error Correction (FEC) encoding, modulation, MIMO processing). The following equation describes the compute resource effort in Giga Operations Per Second (GOPS) $P_{u,t}$ that is required to serve UE u at time t:

$$P_{u,t} = \left(3A_{u,t} + A_{u,t}^2 + \frac{1}{3}M_{u,t}C_{u,t}L_{u,t}\right) \cdot \frac{R_{u,t}}{10} \qquad (1)$$

where A is the number of used antennas, M the modulation bits, C the code rate, L the number of spatial MIMO-layers and R the number of Physical Resource Blocks (PRBs), each as allocated to UE u at time t.

D. Handling of Overload

As we evaluate how to handle reduced processor capacities, the system has to be able to cope with overload situations. We call this functionality overload prevention mechanism. In our model, UPs are assigned to BBUs. Thereby the load of a BBU consists of the sum compute effort caused by UPs assigned to that BBU. We define the overload of a BBU b at time t as load of the BBU which exceeds the capacity of the BBU:

$$O_{b,t} = \max\left(0, \sum_{u \in U_b} P_{u,t} - C_b\right) \qquad (2)$$

where C_b is the capacity of BBU b and U_b is the set of UPs assigned to BBU b.

Each subframe, our system handles overload by modifying the scheduler's decisions, especially reducing the number of allocated PRBs $R_{u,t}$. For that purpose, each UP is assigned a reduced processing capacity:

$$P_{u,t,\text{reduced}} = P_{u,t}\left(1 - O_{b,t}\frac{P_{u,t}}{\sum_{v \in U_b} P_{v,t}}\right) \qquad (3)$$

This ensures that the overload is distributed to the UPs relative to the UPs' requested processing capacity. Subsequently, for each UE, the number of allocated PRBs is reduced such that the allowed processing capacity of the respective UP is not exceeded:

$$R_{u,t,\text{reduced}} = \left\lfloor R_{u,t}\frac{P_{u,t,\text{reduced}}}{P_{u,t}} \right\rfloor \qquad (4)$$

Note that thereby we assume that single PRBs can be assigned to UEs, which is not possible in current LTE systems due to signaling restrictions.

By disabling PRBs, UEs in neighboring cells experience reduced interference. The MCS of these UEs could be adapted to take advantage of that and increase the throughput. However, this would require tight integration of the overload handling with the MCS selection of neighboring cells. In addition, as the processing effort depends on the selected MCS, adapting the MCS to the reduced interference would result in a circular dependency. Therefore, we assume that our system cannot take advantage of the reduced interference. To model this, we calculate the interference based on the PRBs assigned by the schedulers of the neighboring cells, without taking into account the PRBs disabled to handle BBU overload. The reduced number of PRBs is only considered for the calculation of the capacity of the transmissions.

III. Initial Placement Strategies

In this publication, we compare different heuristics to assign the processing effort of UPs to BBUs. For the evaluation, we first define an ideal baseline and an optimal assignment. Subsequently we introduce the heuristics in subsection III-C.

A. Baseline

We define ***ideal*** as the baseline for the comparisons as follows: We do not regard the assignment of UPs to BBUs, but instead compare the total processing effort to the total capacity. The overload of the *ideal* case is defined as

$$O_{\text{ideal},t} = \max\left(0, \sum_{u\in U} P_{u,t} - \sum_{b\in B} C_b\right) \tag{5}$$

For the evaluation in subsection IV-A, we average the overload over the evaluated time span: $O_{\text{ideal}} = \frac{1}{|T|}\sum_{t\in T} O_{\text{ideal},t}$

B. Optimization Problem

When UPs have to be placed to dedicated BBUs, we expect the utilization to be less than ideal because of two reasons: (1) For a single subframe the effort caused by a UP is indivisible. (2) As we do not allow to change the assignments of UPs to BBUs after initial placement, the whole series of efforts caused by a UP at subsequent subframes has to fit into the capacity of the BBU. To evaluate the performance loss caused by these two effects, we introduce an optimization problem (herein after called ***optimal***) as additional baseline for the evaluation.

To restrict the complexity of the optimization problem, we evaluate a limited time span T of 30 subframes (=30ms). The input of the optimization problem consists of the compute effort $P_{u,t}$ caused by each UP u for all subframes $t \in T$. Note that this is equivalent to knowing the future compute effort caused by all UPs, which is difficult to achieve in reality.

We introduce the following variables: The binary flag $a_{u,b} \in \{0,1\}$ is set to 1 if UP u is served by BBU b, and to 0 otherwise. The restriction

$$u \in U: \quad \sum_{b\in B} a_{u,b} = 1 \tag{6}$$

ensures that each UP is served by exactly one BBU (here, B denotes the set of all BBUs and U the set of all UPs). We define the overload of BBU b at time t to be

$$O_{b,t} = \max\left(0, \sum_{u\in U} a_{u,b} P_{u,t} - C_b\right) \tag{7}$$

and the total overload to be

$$O_{\text{opt}} = \frac{1}{|T|}\sum_{t\in T}\sum_{b\in B} O_{b,t} \tag{8}$$

The constant factor $\frac{1}{|T|}$ is introduced for comparison with the other heuristics in subsection IV-A. The objective of the optimization problem is to minimize the total overload O_{opt}.

C. Heuristics

In this subsection, we define four heuristics to assign the processing effort of UPs to BBUs. Each heuristic is to be executed when an UE starts a transmission. The assignment of UPs to BBUs is not changed afterwards.

Heuristic ***static*** resembles the traditional scenario of a non-centralized RAN. Each BBU handles the load of the UPs which are served by three sectors of a single site. This heuristic suffers from the inability to perform load-balancing between BBUs serving hot-spot cells and BBUs serving lowly loaded cells. However, it benefits from a mutual restriction of the effort caused by UPs of the same cell: As the scheduler assigns each PRB to only one UE, the sum of the processing effort caused by all UPs of a cell is limited.

Heuristic ***random***, in contrast, assigns UPs to BBUs randomly (and uniformly) without taking the serving cell of the UE into account. Thereby, the load is implicitly balanced between the BBUs, i.e., the long term average of the load assigned to each BBU is the same. However, as UPs of many cells can potentially be allocated to the same BBU, the variance of the load per BBU is higher than with the static assignment.

Heuristic ***static load-balancing*** aims at reducing the variance of the load while still achieving long-term load-balancing. Thereto we try to keep the UPs of a cell together as far as possible. Furthermore, we assume that it is beneficial for the operation of the BBU pool to assign as many UPs to the Home-BBU of their respective cells as possible. We define an assignment probability $\alpha_{c,b}$ for each combination of cell c and BBU b. These probabilities are calculated offline based on a measurement of the processing effort on a per-cell basis. Upon start of a transmission, the BBU to serve a UP is then selected based on the assignment probabilities of the UEs cell.

As preparation for the heuristic *static load-balancing*, we calculate the assignment probabilities by solving the following optimization problem (not to be confused with the optimization problem in Section III-B). Input to the optimization is the long-term processing effort P_c generated by the UPs of each cell c and the Home-BBU $h_c \in B$ of each cell. Variables are the assignment probabilities $\alpha_{c,b} \in [0,1]$ as introduced above and binary flags $f_{c,b} \in \{0,1\}$, which are set whenever the corresponding assignment probability is larger than zero: $\forall c \in C, \forall b \in B : \alpha_{c,b} \leq f_{c,b}$. In addition, we define the restrictions that all load is served $\forall c \in C : \sum_{b\in B}\alpha_{c,b} = 1$ and that each BBU serves an equal share of the total load $\forall b \in B : \sum_{c\in C}\alpha_{c,b}P_c = \frac{1}{|B|}\sum_{c\in C} P_c$. Based on these definitions and restrictions, we minimize the usage of non-Home-BBUs $U = \sum_{c\in C}\sum_{b\in B|b\neq h_c} f_{c,b}$. This optimization problem is solved offline. At runtime, the heuristic *static load-balancing* assigns each arriving UP to a BBU based on the optimized probabilities $\alpha_{c,b}$.

The heuristic ***dynamic*** uses the actual load of the BBUs to derive a placement decision. The current load $\overline{P_b}$ of a BBU b is calculated as the average over the non-reduced load, i. e., the load before the overload prevention mechanism, during an averaging window of W (1 ms and 1 s evaluated). At the end of each averaging window, the value of $\overline{P_b}$ is reported to the centralized controller. The processing effort $P_{u,\text{pred}}$ for a new UP u is predicted to be the mean value of the last 1000

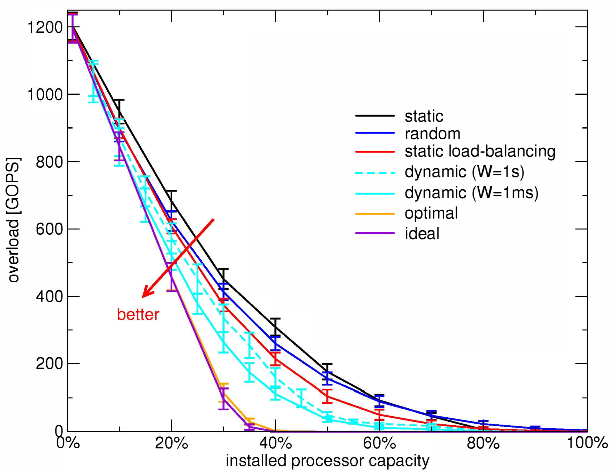

Fig. 2. Overload with different placement strategies at reduced compute capacity, evaluated at 80% system load over a time frame of 30 ms.

samples of the processing efforts of the other UPs served by the same BBU. A new UP is assigned to the Home-BBU h_c of it's cell whenever that BBU has sufficient free capacity ($\overline{P_{h_c}} + P_{u,\text{pred}} < C_{h_c}$). Otherwise, the UP is assigned to the BBU with the lowest current load. In case several BBUs have the same load, one of them is selected randomly. $\overline{P_{h_c}}$ is only updated at the beginning of the window W, i. e., the predicted processing effort of a newly arriving UP is not added to $\overline{P_{h_c}}$. This heuristic explicitly balances the load and is able to adapt to short-term load fluctuations.

IV. Evaluation

The evaluation is split into two parts. The first evaluation shows upper and lower bounds of the achievable pooling gains for different placement heuristics. This part includes mathematical programming to find the optimal placement decisions. The second part uses a simulation model of the presented V-RAN including several placement heuristics as well as the overload prevention mechanism. In this evaluation part we show the impact of reduced processing capacities to the perceived user experience.

We define 100 % system load to be reached when 1 % of newly arriving users are dropped by the AC mechanism. For the following evaluations we configured a system load of 80 % by increasing the IAT of new request-response pairs.

We assume to have 19 equal BBUs. The capacity of the BBUs is configured in percent of the theoretical upper bound of the load. 100 % corresponds to a BBU capacity of 204.33 GOPS, which is the peak load required to serve three cells with the maximum MCS on all PRBs.

A. Comparison of Heuristical and Optimal Placement

This subsection evaluates the overload of the BBUs, which is influenced by the placement strategy. Here, no reduction of the allocated PRBs is performed as described in Section II-D. Instead the overload is recorded and processing continues as if the BBU capacities would be sufficient.

Figure 2 shows how much overload occurs with the compared placement strategies. At high installed BBU capacity, no overload occurs, as distributing the load to the BBUs is easy. If the installed processor capacity is reduced, the compute jobs cannot be allocated to the BBUs without causing overload. The performance of the placement strategy defines how quickly the overload raises. A good placement strategy would be able to maintain zero (or low) overload when reducing the processor capacity. With very low installed capacity, all BBUs can be filled and the remaining overload is independent of the placement strategy.

The *ideal* scheme determines the best possible performance. Down to 40 % processor capacity, it shows no overload. As there is no wasted BBU capacity, below 30 % the curve increases linearly.

The *optimal* assignment shows nearly the same performance as the *ideal* scheme. This means that, although the load caused by each UP is considered as atomic unit of work, the BBUs can be filled up almost ideally. This also leads to the conclusion that a reallocation of UPs to different BBUs during transmission is not required. However, in our evaluation the processing effort of each UP is known in advance. In reality, a reassignment may still be beneficial, because an exact prediction of the future effort is impossible. The heuristics generally perform worse than the *ideal* and *optimal* strategies.

The *static* heuristic does not show overload for more than 80 % installed processor capacity. This can be explained by the fact that cells rarely make use of the theoretical peak processing power, because it is unlikely that all PRBs are transmitted with the highest MIMO mode and MCS [8]. For smaller processor capacities, the overload quickly increases, as the *static* heuristic does not perform load-balancing.

In contrast, the *random* heuristic implicitly balances the load between the BBUs. Therefore, it shows a slightly better performance than the *static* heuristic for processor capacities below 60 %. However, as the assignment is performed randomly, overload sometimes occurs even with more than 80 % installed capacity.

The *static load-balancing* heuristic performs better than the *static* and *random* heuristics in most cases. Compared to the *random* heuristic, it reduces the variance of the load by keeping UPs of the same cell together. However, it does not regard the current load situation of the BBUs, and thereby cannot completely avoid inefficient assignments.

The *dynamic* heuristic realizes significantly lower overload than the other heuristics. By adapting the assignment to the current load of the BBUs, it does not only perform long-term load-balancing, but can also adapt to short-term load fluctuations caused by traffic variations and scheduling effects. As we did not implement advanced predictors for the load of the BBUs or the processing effort caused by a newly arriving UP, the *dynamic* heuristic can still make poor assignment decisions. As we currently do not allow to change the assignment after the transmission has started, this causes an increased overload compared to the optimal assignment. The measurement window W has an obvious influence on the performance. The shorter the measurement window, the better the heuristic can adapt to the current BBU load. The reason is that the processing effort is fluctuating on a small time scale. However, shorter measurement windows could result in increased overhead for the management of the BBU pool.

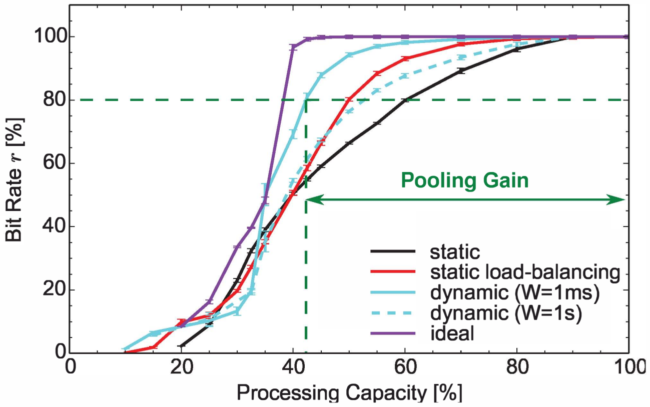

Fig. 3. Experienced bit rates at reduced compute capacity

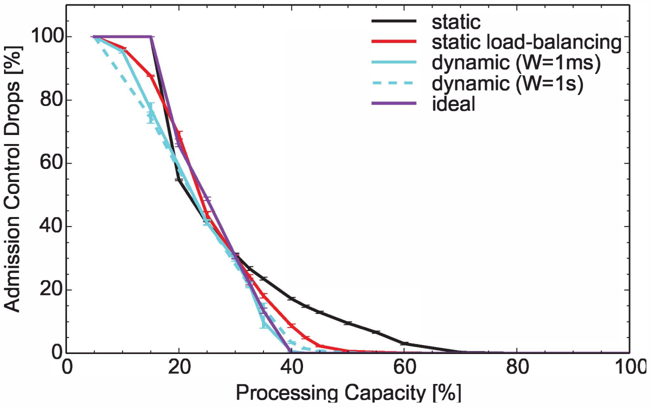

Fig. 4. AC drops at reduced compute capacity

B. *User Experience under Limited Compute Resources*

In the following we evaluate the performance of the presented placement strategies in combination with limited compute resources and the overload prevention mechanism. We measure the achieved DL bit rate for individual transmissions as an indicator for the user experience. Because the implemented AC mechanism influences the number of users in the system, a bad placement decision may lead to higher drop rates and therefore a higher bit rate for remaining users. Therefore, we define the bit rate r as follows:

$$r = \begin{cases} \frac{\text{object size}}{\text{transmission time}} & \text{UE accepted} \\ 0 & \text{UE dropped by AC} \end{cases} \tag{9}$$

Transmission time is defined as the duration between sending the object at the server and receiving it in the UE (including additional 20 ms to model the effects of the core network). An equal distribution of the processing effort to the BBUs, i. e., a better load balancing, results in higher bit rates. Additionally, a better placement strategy leads to less AC drops. The simulation results can be seen in Figure 3 for the bit rate and in Figure 4 for the AC drops. A bit rate of 100 % corresponds to the bit rate in a system with sufficient processing capacity. We define a pooling gain according to the rate degradation. We assume that a degradation of the rate by 20 % is acceptable. The pooling gain is then the amount of hardware saved by accepting this degradation in comparison to the 100 % hardware deployment. For processing capacities smaller than 30 %, the impact of the AC drops becomes dominant. Therefore, it is pointless to evaluate the data rate for these configurations.

The *static* placement strategy is the reference for the evaluation of the heuristics performance. A significant degradation in terms of bit rate as well as an increase of the AC drops begins for processing capacities below 80 %. According to our definition, this placement strategy achieves a pooling gain of 40 %. This is realized by accepting the rate degradation and by the pooling of three cells, but without making use of load-balancing between the sites. The lack of load-balancing results in an imbalance of the AC drops, i.e., there are high drop ratios in cells serving hotspots.

The *random* heuristic can not be applied in real systems, since the random assignment may lead to situations where a single BBU is in overload all the time. In this case the overload prevention mechanism reduces the processing effort in such a way that the granted processing effort of a single user is not even sufficient to process one PRB. This worsens the situation even more, because more users are allocated to the BBU but no user is served at the same time. Therefore, we do not include the *random* heuristic in this evaluation.

Best bit rates can be achieved by the *ideal* setup. However, the bit rate decreases by more than 50 % if the processing capacity is reduced to values lower than 40 %. This is caused by a minimal overload, which occurs for these processing capacities (compare Figure 2). As a result, equation 4 of the the overload prevention mechanism reduces the number of allocated PRBs of every active UE. By the flooring operation, the processing effort can be reduced more than required. Because every UE is affected, the bit rate shows the sharp edge. Interesting is the fact that 100 % AC drops are reached earlier than with the *dynamic* or *static load-balancing* heuristics. The reason is that the overload prevention mechanism reduces the number of PRBs for all active UEs and thereby the UEs stay longer inside the system. In a system with independent smaller BBUs (like in the evaluation of *dynamic* and *static load-balancing*) the overload prevention becomes only active for a subset of all active UEs. So the probability for a single UE to be not affected by the overload prevention is higher and the UE is able to leave the system earlier. By ignoring the inefficiencies caused by assigning UPs to BBUs, the *ideal* configuration achieves a pooling gain of 62 %.

Our proposed *dynamic* placement variant shows higher bit rates than the other heuristics as well as lower AC drops for $W = 1\,\text{ms}$. It achieves a pooling gain of 57 %. For this dimensioning, no AC drops arise. AC drops start occurring for processing capacities below 40 %. We can conclude that the *dynamic* placement strategy ($W = 1\,\text{ms}$) in combination with the overload prevention mechanism is able to achieve results close to the optimum. The overload prevention can shift the required processing effort to a later point in time and thus reduce the negative impact on the user experience.

The *dynamic* placement with $W = 1\,\text{s}$ performs worse than the *static load-balancing* variant, because during the time window all newly arriving UPs are placed to the same BBU. This results in increasing load on the respective BBU over the duration of W. This effect is not visible in the plot in Figure 2, because there only the first 30 ms of the window have been evaluated.

V. Conclusion

In this paper we investigated a V-RAN scenario with non-uniformly distributed load. The evaluations have shown that an advanced dynamic heuristic for initial assignment of UPs to BBUs balances the load of the processing units significantly better than a random or static assignment strategy. With the *dynamic* placement heuristic combined with the proposed overload prevention mechanism pooling gains (savings of compute resources) of 57 % can be achieved when a certain bit rate degradation is accepted. This strategy enables highest bit rates and lowest AC drops in relation to conventional heuristics. However, there is still room for improvement when we compare these results with the outcomes generated with the optimal placement. In future studies we will first examine a mechanism that reallocates UPs to different BBUs as a reaction to load disparities between BBUs and find out to which degree such a reallocation can be used to mitigate the effects of missing or poor predictions of future load. In a next step we want to evaluate to which degree the processing effort caused by a UP can be predicted.

References

[1] China Mobile Research Institute, "C-RAN The Road Towards Green RAN," Tech. Rep. v2.5, 2011. [Online]. Available: http://labs.chinamobile.com/report/view_59826

[2] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, "Wireless network cloud: Architecture and system requirements," *IBM Journal of Research and Development*, vol. 54, no. 1, 2010.

[3] P. Chanclou, A. Pizzinat, F. Le Clech, T.-L. Reedeker, Y. Lagadec, F. Saliou, B. Le Guyader, L. Guillo, Q. Deniel, S. Gosselin *et al.*, "Optical fiber solution for mobile fronthaul to achieve cloud radio access network," in *Future Network and Mobile Summit (FutureNetworkSummit), 2013*. IEEE, 2013.

[4] A. Checko, H. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks - a Technology Overview," *Communications Surveys Tutorials, IEEE*, 2014.

[5] Fujitsu Network Communicaitons Inc., "The Benefits of Cloud-RAN Architecture in Mobile Network Expansion," White Paper, 2014. [Online]. Available: www.fujitsu.com/downloads/TEL/fnc/whitepapers/CloudRANwp.pdf

[6] J. Huang, R. Duan, C. Cui, and I. Chih-Lin, "Overview of cloud RAN," in *General Assembly and Scientific Symposium (URSI GASS), 2014 XXXIth URSI*, Aug 2014.

[7] B. Haberland, F. Derakhshan, H. Grob-Lipski, R. Klotsche, W. Rehm, P. Schefczik, and M. Soellner, "Radio Base Stations in the Cloud," *Bell Labs Technical Journal, General Papers Issue*, vol. 18, no. 1, June 2013.

[8] T. Werthmann, H. Grob-Lipski, and M. Proebster, "Multiplexing Gains Achieved in Pools of Baseband Computation Units in 4G Cellular Networks," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2013.

[9] S. Namba, T. Matsunaka, T. Warabino, S. Kaneko, and Y. Kishi, "Colony-RAN architecture for future cellular network," in *Future Network Mobile Summit (FutureNetw), 2012*, July 2012.

[10] D. Zhu and M. Lei, "Traffic and interference-aware dynamic BBU-RRU mapping in C-RAN TDD with cross-subframe coordinated scheduling/beamforming," in *Communications Workshops (ICC), 2013 IEEE International Conference on*, June 2013.

[11] ——, "Traffic adaptation and energy saving potential of centralized radio access networks with coordinated resource allocation and consolidation," in *Communications and Networking in China (CHINACOM), 2013 8th International ICST Conference on*, Aug 2013.

[12] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, "CloudIQ: A Framework for Processing Base Stations in a Data Center," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: ACM, 2012.

[13] S. Namba, T. Warabino, and S. Kaneko, "BBU-RRH switching schemes for centralized RAN," in *Communications and Networking in China (CHINACOM), 2012 7th International ICST Conference on*, Aug 2012.

[14] A. Checko, A. Checko, H. Holm, and H. Christiansen, "Optimizing small cell deployment by the use of C-RANs," in *European Wireless 2014; 20th European Wireless Conference; Proceedings of*, May 2014.

[15] C. Wang, Y. Wang, C. Gong, Y. Wan, L. Cai, and Q. Luo, "A study on virtual BS live migration - A seamless and lossless mechanism for virtual BS migration," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*, Sept 2013.

[16] F. Hernández-Campos, J. S. Marron, G. Samorodnitsky, and F. D. Smith, "Variable heavy tails in internet traffic," *Perform. Eval.*, vol. 58, no. 2+3, Nov. 2004.

[17] J. Ellenbeck, J. Schmidt, U. Korger, and C. Hartmann, "A concept for efficient system-level simulations of ofdma systems with proportional fair fast scheduling," in *GLOBECOM Workshops, IEEE*, Dec. 2009.

[18] "Further advancements for E-UTRA physical layer aspects, v9.0.0," 3GPP WSG RAN, Tech. Rep. TR 36.814, 2010.