Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

2016 IEEE 27th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC): Mobile and Wireless Networks

Reallocation Strategies for User Processing Tasks in Future Cloud-RAN Architectures

Sebastian Scholz*, Heidrun Grob-Lipski†
*Institute of Communication Networks and Computer Engineering, Universität Stuttgart, Stuttgart, Germany.
†Bell Labs, Nokia, Stuttgart, Germany
sebastian.scholz@ikr.uni-stuttgart.de, heidrun.grob-lipski@nokia.com

***Abstract*—In this paper we evaluate strategies to reduce the required processing capacity in a Cloud-Radio Access Network (C-RAN) architecture by improving the placement of user processing tasks. Our approach of assigning compute tasks in a pool of compute resources is based on fine granular tasks, where one compute task per served user is introduced. We compare different strategies in order to balance the load in the pool and save processing resources. Therefore we evaluate the best possible reallocation method by formulating an optimization problem including extensions to reduce the number of reassignments. We also introduce an algorithm for dynamic reallocations that can be implemented in real systems. From the evaluation results we can conclude that all strategies reduce the total overload by enhanced load balancing. Further all strategies improve the perceived Quality of Experience (QoE) of individual users.**

I. Introduction and Related Work

Cloud-Radio Access Network (C-RAN) is a key enabler for future mobile networks. It has the potential to save deployment and operational costs by simultaneously improving the system performance. Compared with traditional Decentral-RAN (D-RAN) we can expect a factor of four for the maximum statistical multiplexing gain [1].

Traffic variations in time and space hold significant potentials for multiplexing gains in C-RAN environments [2] and with C-RAN we are able to flexibly adapt to the current traffic demand [3].

The critical aspects like increased link capacity requirements and the choice of the applied virtualization techniques are discussed intensively [4]. Promising solutions to address the fiber consumption are given in [5] but the virtualization of wireless communication systems is still challenging due to the strict real-time processing requirements.

In order to tackle compute time variations induced from virtualization the authors of [6] propose an open-source software-based cloud execution architecture that shows nearly as good processing deadlines as dedicated implementations.

The feasibility of end-to-end applications running over virtual base station pools on multi-core platforms is proved in [7] and [8] demonstrates that strict timing requirements can be met by use of computational load balancing and massive parallelization. In [9] the authors introduce concepts to discuss flexible functionality assignment and cloud technologies in 5G RAN with the aim to support network densification and centralization.

A publication from [10] introduces a flexible functional split between the RAN front-ends and the Base Band Unit (BBU) pool. These splits need to address constraints coming from architecture and implementation as well as existing Long Term Evolution (LTE) Radio Access Network (RAN) specifications in order to support different applications.

The C-RAN architecture introduced in [11] constitutes the basic concept for the investigations presented in this publication. The uniqueness of the solution is that it enables flexible pooling on user and even bearer granularity. Each Remote Radio Head (RRH) is associated to its Home-BBU, which performs the Physical Layer Cell functions, like Framing/Deframing, iFFT/FFT, for the RRH's cell. The traffic load dependent functions per user, further on called User Processing (UP), comprise S1 termination, PDCP, RLC, MAC and User Scheduling and the load dependent functions of the physical layer. The big advantage of such an approach is that a UP stack can be reassigned quickly and the throughput can be kept during migration from one BBU to another. When a new user arrives, the UP is initially placed on its Home-BBU. Load peaks are encountered locally within the BBU by overload prevention first. If this measure does not succeed the architecture allows reallocating UPs to other BBUs.

Starting from this architecture [12] already presents heuristics, which decide for the initial UP assignment to a processing unit. Compared with a classical cell-based allocation strategy the proposed heuristic enables about 50% savings of hardware resources by keeping the required service quality.

As a follow-up and in contrast to [12] in this publication we allow for dynamic reallocations of user jobs in an optimized way in order to achieve balanced load within the pool and save compute resources.

Using the same C-RAN architecture the authors of [13] aim at reducing the energy consumption. Similar to our approach they modify the assignment of UPs to BBUs to save processing capacity. In contrast, they assume the same constant processing effort for all UPs, which significantly simplifies the allocation of UPs to BBUs.

The remainder of the paper is structured as follows. In Section II, we describe the system model comprising radio network, user distribution, traffic and processing effort model. Section III introduces different variants of dynamic reallocations and formulates them as optimization problems. In Section IV we explain an algorithm for dynamic reallocation used in the system level simulation for comparison with the optimization results. Further we introduce a necessary overload prevention mechanism. We present the evaluation results in Section V comprising overload values for optimized

978-1-5090-3254-9/16/$31.00 ©2016 IEEE

and heuristic reallocations as well as results showing the user experience under limited compute resource conditions. Finally, the paper is concluded in Section VI.

II. System Model

We use a 10MHz LTE system as the basis for the evaluations. The RRHs (macro cells) are arranged in 19 hexagonal sites each comprising 3 sector cells In total 57 cells are available. To avoid border effects wrap-around is applied. We focus on the downlink (DL) operation in the physical layer, as the physical processing generates the highest processing effort. Even though uplink (UL) processing causes higher computational effort in the base station or BBU, the principal behavior is similar to the DL [14]. The Multiple-Input-Multiple-Output (MIMO) and Forward Error Correction (FEC) processing is more complex in the receiver side than for the transmitter. In the downlink this effort scales quadratically with the number of used transmit antennas, in uplink the processing effort scales cubically.

A. Radio network model

As a basis for our evaluation we modeled the radio network according to the parameters in Table I, which are compliant to 3GPP specifications.

Our model supports to choose between 2x2 MIMO, Space-Frequency Block Coding (SFBC) and Single-Input-Single-Output (SISO), depending on the channel conditions. The transmit power and the signal degradation between all active transmitters and the receiver as well as the noise is used to determine the channel conditions in terms of the Signal-to-Interference-and-Noise-Ratio (SINR). An appropriate LTE Modulation and Coding Scheme (MCS) is chosen according to the calculated SINR. With MCS, transmission mode and Block Error Rate (BLER) tables it is possible to derive the bit rate. Further we assume ideal channel knowledge and a target decode probability of 80 %. To model the Automatic Repeat Request (ARQ) process, failed transmissions are reinserted into the sending buffer after 8 ms.

B. User and traffic model

The user distribution has a large impact on the achievable multiplexing gain. To replicate the non-uniform distribution of users in the real world, we place the users with a probability of 50 % in three hotspots which are located in the scenario so that the distance between the centers of the hotspots is maximized. The remaining 50 % are placed uniformly over the whole scenario.

Because the traffic pattern in the network influences possible multiplexing gains, we do not apply a full buffer approach. Instead we model the traffic as request-response pairs. After each finished transmission of request and response the user is moved to a new position. The object sizes of request and response are based on measurements in a campus network [15]. As we want to avoid problems with large objects the distributions are clipped at 100 Mbyte. The load in the network can be controlled by varying the negative exponentially distributed Inter-Arrival Time (IAT) of new request-response pairs. Further we have defined a simple admission control, which drops arriving requests when there are more than 100 users active in the sector.

C. Processing effort model

We apply a model for the compute effort generated by a UP first introduced in [2]. It comprises the physical layer processing of a UP as explained in Section I. The input of the model is based on the scheduling decision, i. e., the number of radio resources, the transmission mode and applied MCS. The output is processing effort per UP per Transmission Time Interval (TTI).

The following equation describes the compute resource effort $P_{u,t}$ in Giga Operations Per Second (GOPS) that is required to serve UP u at time t:

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

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 UP u at time t.

III. Optimal dynamic reallocation

In this chapter we introduce different variants of dynamic reallocations and show their formulation as optimization problems. The purpose of dynamic reallocations of UPs is to achieve a load balancing between the BBUs. Equal load distribution results in maximized multiplexing gains.

Real systems might not be able to support unlimited number of reallocations. E.g. if UPs are realized as Virtual Machines (VMs), reallocations generate additional network load when the memory content has to be transferred from one BBU to another. Therefore we present an extension that reduces the number of reallocations. In a real system there are even more possibilities how to restrict the number of reallocations to numbers that can be handled. E. g., in our proposed algorithm in Section IV reallocations are only performed for BBUs that are currently overloaded.

A. Baseline

The ideal baseline (referred to as *Ideal*) is determined by recording the total overload of all BBUs in the system.

$$O_{ideal} = \frac{1}{|T|}\sum_{t\in T}\max\left(0, \sum_{u\in U}P_{u,t} - \sum_{b\in B}C_b\right) \quad (2)$$

B defines the set of all BBUs, U the set of all UPs and C_b the processing capacity of BBU b. The total overload is divided by the duration $|T|$ to make it independent of the considered time range T. Here we assume only discrete time steps, as LTE also

TABLE I. System model parameters

Property	Value
Cell layout	19 sites, 3 sectors per site, 500 m inter site 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 [16]
BS antenna model	3D, 15° tilt, from [16]
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

operates in a slotted fashion with time steps of 1 ms. Note that the ideal baseline equals a system with only one BBU with a processing capacity of $\sum_{b \in B} C_b$.

B. Optimization problem

The objective of the optimization problem is to perform the reallocations of UPs so that the total overload O_{opt} in the system is minimized. The overload is defined as:

$$O_{opt} = \frac{1}{|T|} \sum_{t \in T} \sum_{b \in B} O_{b,t} \tag{3}$$

where $O_{b,t}$ represents the overload of BBU b at time t. In the optimization problem we introduce the binary flags $a_{u,b,t}$, which are set to 1 if UP u is served by BBU b at time t, and to 0 otherwise. The restriction

$$\sum_{b \in B} a_{u,b,t} = 1 \quad \forall u \in U, t \in T \tag{4}$$

ensures that each UP is served by exactly one BBU during the considered time interval. With the help of the binary flags, we can define the overload for BBU b at time t as:

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

C. Optimization problem with intervals

As an extension of the original optimization problem we introduce the concept of intervals to allow reallocations only at fixed points in time. During the intervals the UPs stay on the BBU they are assigned to. Newly arriving users are initially assigned to their Home-BBU. This concept decreases the necessary signaling and load caused by reallocations.

Still, the objective of the optimization problem is to minimize the total overload O_{opt}. The only difference is the way the binary flags have to be handled:

$$a_{u,b,t} = \begin{cases} 1 & b = h_u, \forall u \in U, t < \lceil \frac{t_a}{t_i} \rceil t_i \\ 0 & otherwise \end{cases} \tag{6}$$

Where h_u is the Home-BBU of UP u, t_a the first time index where $P_{u,t} > 0$ (the start of the activity phase of a newly arriving user) and t_i the length of the reallocation interval. This ensures that the UP is initially assigned to the Home-BBU and stays there until the next reallocation is permitted. Note that the problem from III-B is a special case with $t_i = 1$.

At the times $t = nt_i$ the binary flags may be changed to minimize the overload for all UPs with $t_a < nt_i$. Here, n is an integer variable to count the reallocation intervals. Whereas for the times $t = nt_i, \ldots, (n+1)t_i - 1$ the binary flag $a_{u,b,t}$ has the value of a_{u,b,nt_i}.

D. Static

To allow a comparison with the worst case, we introduce the *Static* variant, which performs a static initial assignment without any dynamic reallocation. This resembles a D-RAN, where UPs are executed on the processing hardware responsible for the cell of the user. Again we record the overload:

$$O_{static} = \frac{1}{|T|} \sum_{t \in T} \sum_{b \in B} O_{b,t} \tag{7}$$

The overload $O_{b,t}$ is the same as defined in equation (5). The binary flags $a_{u,b,t}$ are set as follows:

$$a_{u,b,t} = \begin{cases} 1 & b = h_u, \forall u \in U, t \in T \\ 0 & otherwise \end{cases} \tag{8}$$

IV. Realizable reallocation mechanism

In this section we present two mechanisms needed to implement dynamic reallocations in real systems. The first one is an algorithm to perform reassignments which is an heuristic approach to imitate the behavior of the optimization problems described in the previous section. Additionally a mechanism to prevent short term overload situations is required. This mechanism has to be executed every TTI and modifies the scheduling decision in order to keep the processing effort below the available capacity.

A. Reallocation algorithm

In contrast to the optimal reallocation strategies introduced in Section III we present an heuristic that can be implemented and used to evaluate the performance of dynamic reallocations under limited processing resources. The pseudocode can be found in Algorithm 1. This procedure can be executed either every TTI or similar to the optimization problem in section III-C at predefined intervals.

Every time when the algorithm is started, it iterates over all available BBUs and checks if it is overloaded (Line 2). In case of overload all UPs which are currently assigned to the currently evaluated BBU are sorted in descending order according to their processing effort (Line 3). Sorting of UPs before reallocation has the advantage that as few UPs as possible will be reassigned. Then we start a loop to iterate over the sorted UPs as long as the currently examined BBU still suffers from overload. For each UP we check whether it can be reallocated back to its Home-BBU. A UP can be reassigned back, if it is currently on another BBU and the Home-BBU has sufficient free processing capacity (Lines 5 to 9). Otherwise the UP is reallocated to the BBU with the lowest processing effort. If this is not possible, we try to reassign the UP to other BBUs with ascending processing effort (Lines 13 to 17). The reason why we first try to reallocate UPs back to their Home-BBU is that we want to prevent the fragmentation of UPs, which are located in the same cell, onto many BBUs. Further it has been observed in [12] that assigning UPs on their Home-BBU helps to reduce the overload. This idea is reused here for dynamic reallocations.

B. Overload prevention mechanism

In the system level simulation as well as in the real system overload of BBUs has to be handled, when the available processing capacity is exceeded. Long term overload situations can be resolved through dynamic reallocations. However, short term overload needs to be tackled also. Therefore we introduce an overload prevention mechanism. Each TTI this mechanism treats 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:

$$P_{u,t,\text{reduced}} = P_{u,t} \left(1 - O_{b,t} \frac{P_{u,t}}{\sum_{v \in U_b} P_{v,t}} \right) \tag{9}$$

Algorithm 1 Dynamic Reallocation Algorithm

for all $b \in B$ **do**
  if $P_{b,t} \geq C_b$ **then**
    Sort descending $U_b = [P_{u,t} | a_{u,b,t} = 1,\ u \in U]$
    for all $u \in U_b$ **do**
      if $b \neq h_u$ **and** $P_{h_u,t} + P_{u,t} \leq C_{h_u}$ **then**
        $P_{b,t} \leftarrow P_{b,t} - P_{u,t}$
        $P_{h_u,t} \leftarrow P_{h_u,t} + P_{u,t}$
        $a_{u,b,t} \leftarrow 0$
        $a_{u,h_u,t} \leftarrow 1$
      else
        Sort ascending $B_{sorted} = [P_{b',t} | b' \in B]$
        for all $b' \in B_{sorted}$ **do**
          if $b' \neq b$ **and** $P_{b',t} + P_{u,t} \leq C_{b'}$ **then**
            $P_{b,t} \leftarrow P_{b,t} - P_{u,t}$
            $P_{b',t} \leftarrow P_{b',t} + P_{u,t}$
            $a_{u,b,t} \leftarrow 0$
            $a_{u,b',t} \leftarrow 1$
          end if
        end for
      end if
      if $P_{b,t} \leq C_b$ **then** break **end if**
    end for
  end if
end for

where U_b is the set of UPs on BBU b, $O_{b,t}$ is the overload as defined in (3) and $P_{u,t}$ the processing effort of UP u as defined in (1). 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:

$$R_{u,t,\text{reduced}} = \left\lfloor R_{u,t} \frac{P_{u,t,\text{reduced}}}{P_{u,t}} \right\rfloor \tag{10}$$

Note that thereby we assume that single PRBs can be assigned to UEs, which is only possible in LTE resource allocation type 1 [17]. However, this concept can be easily extended to other resource allocation types. Also, we do not exploit the effect that disabling PRBs reduces interference experienced by neighboring cells. The MCS is not adapted to the reduced interference and the disabled PRBs are only considered for calculating the capacity of the affected transmission.

V. Evaluation

The investigation is split into three parts. The first part shows upper and lower bounds of the achievable multiplexing gains for different reallocation variants by solving the optimization problems from sections III-A to III-D. We use traces of the processing effort per UP generated in a system level simulation as input for the optimization problems. Since the optimization is too complex and time consuming to execute for longer time intervals, we only consider a short time range of 30 ms, which is feasible because the traffic model generates many short living transmissions. Then we present results of the algorithm from section IV implemented in a system level simulation. During the simulation run the overload is recorded without activated overload prevention mechanism. These results are also compared with the optimized results. Finally, the last evaluation uses only the simulation model of the presented network including the overload prevention mechanism. In this part we show the impact of reduced processing capacities on the user experience.

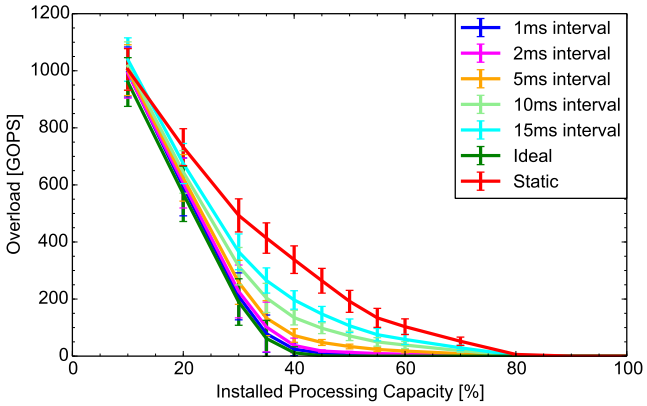

Fig. 1. Overload of optimized reallocations

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 processing effort required to serve three cells with the best MCS and transmission mode on all PRBs. The following results show the mean of 10 independent runs as well as the 95 % confidence intervals.

During the evaluations the sum downlink rate of the system is configured to 430 Mbit/s by adjusting the IAT of new request-response pairs. This load leads to an admission control drop rate for newly arriving users of 1 %.

A. Overload of optimized reallocations

The results of the first assessment can be seen in Figure 1. We reduce the available processing capacity C_b from 100 % down to 10 % and record the total overload as defined in equation (3). A satisfactory task reallocation variant would keep the overload at zero or at least at a low level. If the installed processing capacity is small, the measured overload is independent of the applied strategy, because all BBUs are fully loaded. Note that the results denoted with "1 ms interval" are equal to the results obtained with the optimization without any intervals (Section III-B).

As we can see in the figure, the gap between 1 ms Interval and *Ideal* solution is quite small, whereas the overload of the *Static* solution is much higher. The reason for the difference between *Ideal* and 1 ms Interval solutions is the granularity of the UPs. In the 1 ms Interval solution they have to be assigned to the 19 available BBUs, whereas in the *Ideal* case they are assigned to a 19 times larger BBU. If we consider the processing capacity needed to serve the UPs without causing overload, we observe in Figure 1 that more than 70 % processing capacity is needed in the *Static* case. In contrast, overload occurs for the *Ideal* and 1 ms Interval solution only for processing capacities lower than 40 %.

The results clearly show that the interval should be chosen as small as possible, to reduce overload. However, the dif-

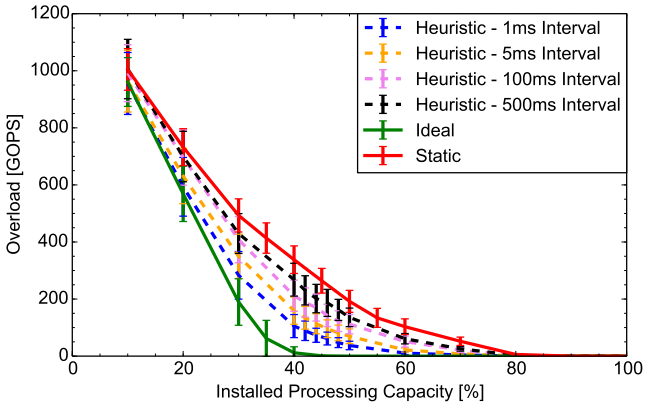

Fig. 2. Overload of heuristic reallocations

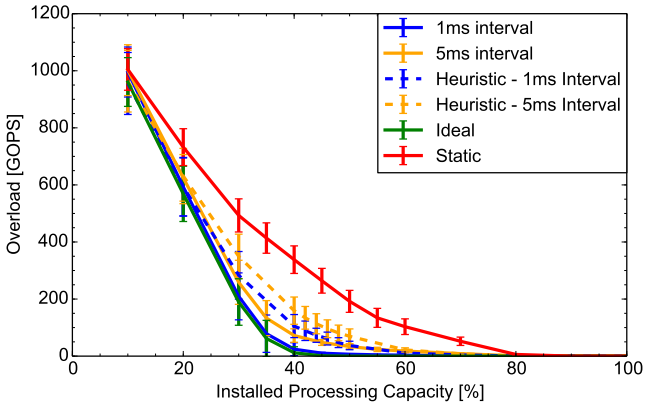

Fig. 3. Comparison of optimized and heuristic reallocations

ference between intervals of 1 ms and 2 ms are quite small. Intervals of 5 ms or larger significantly increase the overload compared to smaller intervals or the *Ideal* variant. Even if during the considered time range reallocations are performed only once, as in the case of an interval of 15 ms, the overload is smaller than for the *Static* variant.

The goal of introducing intervals is to reduce the number of reallocations, because depending on the system design only a limited number of reallocations might be possible. Table II shows that the approach of performing reallocations only after discrete intervals really reduces the number of reallocations. The left column of the table shows the configured interval, the column in the middle the total number of performed reallocations during the considered time range of 30 ms and the right column the average number of performed reallocations per interval. The total number of reallocations can be reduced noticeably. However, the number of reallocations per interval does not decrease significantly. The reason is the applied traffic model, which generates many relatively short living users. So the number of reallocations per interval depends on the number of users that are active from one interval to the next.

B. Overload of heuristic reallocations

The results of the reallocation algorithm introduced in Section IV-A can be found in Figure 2. During simulations it is possible to consider larger time ranges, so we included reallocation intervals up to 500 ms. Besides the outcomes of the algorithm we also include the results of the *Ideal* and *Static* case as upper and lower bounds.

Similar to the optimized results the overload rises with increasing reallocation intervals. But even if reallocations are only triggered every 500 ms the overload is smaller than for the *Static* variant. However, the gap between *Ideal* solutions and the solutions of the reallocation algorithm with an interval of 1 ms is significantly larger than in the optimized case.

TABLE II. AVERAGE NUMBER OF PERFORMED REALLOCATIONS FOR 60 % INSTALLED PROCESSING CAPACITY

Interval [ms]	Total Number of Reallocations	Reallocations per Interval
1	1861.2	64.1
2	910.6	65.0
5	320.6	64.1
10	113.2	56.6
15	37.5	37.5

By comparing the optimized results and the outcomes of the reallocation algorithm in Figure 3 we can conclude that the performance of the simple heuristic is worse than the optimal reallocation scheme. However, the proposed algorithm can be implemented in real systems and requires only little state information of the system. Especially it does not include any predictions of future processing efforts.

C. User experience under limited compute resources

In the last step we evaluate how the overload impacts the service quality perceived by the users. The results can be found in Figure 4. We measure the achieved DL bit rate for individual transmissions as an indicator for the user experience. Poor reallocation decisions lead to situations where users get less processing capacity and therefore stay in the system for a longer time. This may lead to higher admission control drop rates and therefore to a higher bit rate for remaining users. Thus, we define the bit rate r as follows:

$$r = \begin{cases} \frac{\text{object size}}{\text{transmission time}} & \text{UP accepted} \\ 0 & \text{UP dropped by admission control} \end{cases} \quad (11)$$

Transmission time is defined as the duration between sending the object at the server and receiving it in the UE. Here we define the 100 % bit rate level to be achieved in the case of 100 % processing capacity. Besides the effect of admission control drops, a system with a better reallocation variant is able to serve more users at the same time, because the processing effort generated by the users is balanced on the available compute resources. Therefore the individual user is affected less by the overload prevention mechanism.

As expected from the previous outcomes a short reallocation interval leads to higher bit rates. Also the variant with an interval of 1 ms achieves results close to the *Ideal* case. For processing capacities above 44 % the *Static* variant performs worst. However, for capacities below 44 % *Static* becomes better then all heuristic variants. The reason is as stated above that for capacities below 44 % all reallocation algorithms are able to serve more users and in consequence the bit rate is

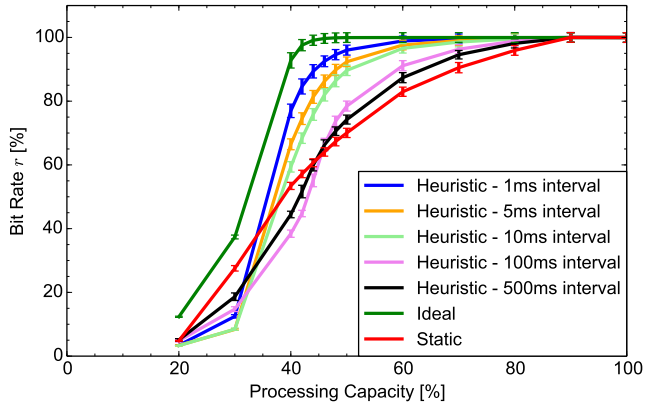

Fig. 4. QoE under restricted compute capacity

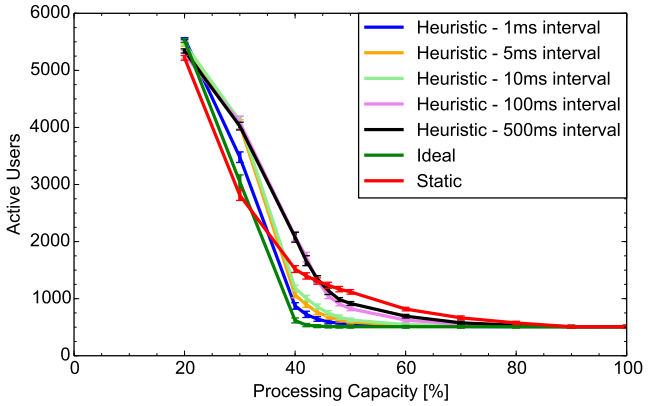

Fig. 5. Number of simultaneously active users

smaller for the individual user. The average number of users in the system served at the same time is depicted in Figure 5.

By observing the behavior of the Quality of Experience (QoE) the trade-off between bit rate and required processing capacity gets evident. By allowing a certain degradation of the bit rate the achievable multiplexing gain can be adjusted.

VI. Conclusion

In this paper we have investigated a mobile network architecture supporting fine granular reallocation of user processing tasks in a pool of multiple BBUs. The results show that the combination of task reassignment and short term overload prevention mechanism allows to utilize the trade-off between compute resources and QoE. In the best case we can economize up to 40 % of processing resources without any QoE degradation. If a degradation of QoE is accepted, even more compute resources can be saved.

A promising extension of this work is the combination of reallocation and improved initial placement of UPs, as presented in [12]. We plan to investigate the interplay of both algorithms in a future publication and expect that even higher multiplexing gains are possible. Additionally we will analyze for which types of traffic initial placement or dynamic reallocations are more appropriate.

Acknowledgment

We are thankful to our colleagues Bernd Haberland from Bell Labs and Thomas Werthmann from the Institute of Communication Networks and Computer Engineering who provided their expertise to develop the ideas and algorithms presented in this paper.

References

[1] A. Checko, H. L. Christiansen, and M. S. Berger, "Evaluation of energy and cost savings in mobile cloud ran," *OPNETWORK 2013*, 2013.

[2] 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.

[3] 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*, 2012.

[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*, vol. 17, no. 1, 2015.

[5] 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*, 2014.

[6] I. Alyafawi, E. Schiller, T. Braun, D. Dimitrova, A. Gomes, and N. Nikaein, "Critical issues of centralized and cloudified LTE-FDD radio access networks," in *ICC 2015, IEEE International Conference on Communications*, 2015.

[7] Z. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, H. Franke, and S. Sarangi, "Virtual base station pool: Towards a wireless network cloud for radio access networks," in *Proceedings of the 8th ACM International Conference on Computing Frontiers*, ser. CF '11. ACM, 2011.

[8] H. Paul, D. Wübben, and P. Rost, "Implementation and Analysis of Forward Error Correction Decoding for Cloud-RAN Systems," in *Second International Workshop on Cloud-Processing in Heterogeneous Mobile Communication Networks (IWCPM 2015)*, 2015.

[9] P. Rost, C. Bernardos, A. Domenico, M. Girolamo, M. Lalam, A. Maeder, D. Sabella, and D. W??bben, "Cloud technologies for flexible 5g radio access networks," *Communications Magazine, IEEE*, vol. 52, no. 5, 2014.

[10] A. Maeder, M. Lalam, A. D. Domenico, E. Pateromichelakis, D. Wübben, J. Bartelt, R. Fritzsche, and P. Rost, "Towards a flexible functional split for cloud-ran networks," in *European Conference on Networks and Communications*, 2014.

[11] 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, 2013.

[12] T. Werthmann, H. Grob-Lipski, S. Scholz, and B. Haberland, "Task Assignment Strategies for Pools of Baseband Computation Units in 4G Cellular Networks," in *Second International Workshop on Cloud-Processing in Heterogeneous Mobile Communication Networks (IWCPM 2015)*, 2015.

[13] X. Wang, S. Thota, M. Tornatore, H. S. Chung, H. H. Lee, S. Park, and B. Mukherjee, "Energy-efficient virtual base station formation in optical-access-enabled cloud-ran," *IEEE Journal on Selected Areas in Communications*, vol. PP, no. 99, 2016.

[14] C. Desset, B. Debaillie, V. Giannini, A. Fehske, G. Auer, H. Holtkamp, W. Wajda, D. Sabella, F. Richter, M. Gonzalez, H. Klessig, I. Godor, M. Olsson, M. Imran, A. Ambrosy, and O. Blume, "Flexible power modeling of LTE base stations," in *Wireless Communications and Networking Conference (WCNC), IEEE*, 2012.

[15] 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, 2004.

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

[17] "Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures, v9.0.0," 3GPP WSG RAN, Tech. Rep. TR 36.213, 2010.