Anticipatory Power Cycling of Mobile Network Equipment for High Demand Multimedia Traffic

Martin Dräxler
University of Paderborn
Paderborn, Germany
Email: martin.draexler@upb.de

Philipp Dreimann*
University of Paderborn
Paderborn, Germany
Email: dreimann@mail.upb.de

Holger Karl
University of Paderborn
Paderborn, Germany
Email: holger.karl@upb.de

***Abstract*—The increasing energy demand of mobile phone networks is a problem for the environment and a cost factor for mobile network operators. Multimedia content is increasingly popular, demands high data rates, and increases energy usage of mobile phone networks. Thus it is important to reduce energy usage of mobile phone networks while serving multimedia content with a high quality of experience (QoE).**

We developed an approach to power-cycle base stations and control the playback of video streams while reducing the energy consumption and not decreasing the QoE for users. We present two implementations: an optimization problem and an iterative algorithm.

Up to 80% of energy can be saved by combining power cycling and video stream controlling in low-load situations. With increasing load, the energy consumption also increases while the QoE can still be improved.

I. INTRODUCTION

Mobile services with high data rate demands, like high-quality video streaming, are already widely adopted by users of mobile broadband networks and the demand for such services is predicted to grow rapidly in the next years. Existing studies [1] suggest an increase of mobile traffic by a factor of 25 until 2016 compared to 2011, with two thirds of this traffic being video streaming.

Conflicting with this trend of providing higher data rates for mobile broadband networks is the trend of reducing the energy consumption of mobile broadband networks. Existing research [2] has identified the power cycling of base stations (i.e. switching on/off complete base stations or high-power-consumption components) as the most promising approach to reducing the energy consumption of mobile broadband networks. While this might not be possible to implement for macro base stations without sacrificing coverage, it is a promising approach to deploy and operate small cells (e.g. pico base stations) in a more energy-efficient way.

In this paper we describe a scheduling approach that integrates uninterrupted high-quality video streaming with power cycling unneeded base stations. For video streaming, we use segmented video streaming over HTTP like HLS [3] or MPEG DASH [4], [5]. In these protocols the video is not streamed continuously but divided into segments. One segment usually contains a few tens of seconds of video data and each segment is downloaded individually by the video player as a normal HTTP download. Each segment can be available on the server in different qualities (i.e. resolutions, bitrates, codecs) and the player application chooses in which quality to download which segment. As different qualities result in different file sizes for each segment, the download quality is usually determined by the data rate observed when downloading previous segments, like it is implemented in the VLC media player [6]. This default mechanism can be replaced by a precise *download schedule* to exactly determine *when* each segment is downloaded and in *which quality*.

In our previous work [7], [8] we have shown how a mobile network operator can use such a *download schedule* to adapt video streaming to changing available data rates for uninterrupted video streaming at a high quality. This approach is based on the prediction [9] of future available data rates in the range of seconds and minutes, which we call *anticipation*. The same prediction mechanisms can also be used to predict future user locations. This allows the anticipatory download of more video segments in advance to switch off base stations which are then not needed to download segments.

The rest of the paper is structured as follows: First we give an overview of related work and provide a more description of the outlined scheduling problem. We then present both an optimization problem and a heuristic algorithm to solve the scheduling problem in Sections II and III. After that we compare and evaluate both approaches in Section IV and conclude our work in Section V.

A. Related work

The anticipatory approach for video download [7], [8] has been evaluated before and we base our optimization problem on it. A similar approach with the same system model and a different optimization problem has been proposed by Sadr et al. [10]. Furthermore, Evensen et al. [11] use a data rate prediction service developed by Riiser et al. [12] to predict the data rate available to a user on a preselected movement path. This data is then used to compute a buffer filling scheme for a video stream. Radhakrishnan et al. [13] propose a video streaming protocol which takes CQI values into account. None of these approaches for scheduling of video streaming incorporates the aspect of energy consumption.

The power cycling of small cells without the aspect of video download scheduling has also been investigated: Bennis et al. [14] studied a combination of small cells together with wireless LAN access points to separate multimedia and

---

*Philipp Dreimann is now with Deutsche Telekom, Germany

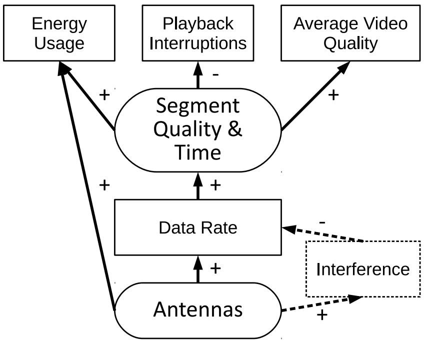


Figure 1. Correlation between problem variables and resulting effects

voice traffic for a more energy-efficient operation of the network. Fehske et al. [15], [16] evaluated the energy efficiency of small cells and conclude that the deployment of small cells can increase energy efficiency, which is further enhanced by the power cycling of those. Finally, Blume et al. [17] proposed various ways how to save energy in cellular networks by analyzing traffic statistics and adapting the network to them, including the power cycling of base stations or individual antennas/sectors. These investigations are more general and are not combined with a specific application like multimedia streaming.

B. Problem & model description

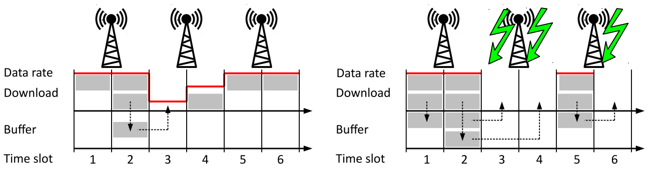
The previously outlined scheduling problem can be summarized by the following two question:

1) *When* should each segment be downloaded at *which quality*?
2) *Which* base station (or antenna) should be turned on *when*?

To properly formulate the scheduling problem as an optimization problem we assume a discrete time model. For each scenario we assume that a video stream is download by all users and the streaming starts in the first time slot t_0 and ends in the last time slot $t_{|T|}$. The length of a video segment is equal to the length of a time slot and the video is $|T|$ segments long. This means that for an uninterrupted playback a segment s_i has to be downloaded at the latest in time slot t_i. The quality of the video is determined by the resolution, bitrate or codec and different qualities result in different file sizes for each segment in each quality. For simplicity we assume that all segments in one quality q have the same file size, hence, quality and file size are interchangeable. Thus for each segment s, a quality q and a download time t have to be determined per user.

The base station power cycling can be modeled in a simple way: For each time slot t an antenna can be switched on or off. If it is switched on, users can be associated with it and the base station can provide data rate to its associated users.

Figure 1 illustrates the correlation between the two scheduling decisions and the resulting effects: enabling more antennas potentially increases the available data rate, but also increases the energy consumption. The more data rate is available the more freedom for download times exists and the playback interruptions are reduced. Also, a higher video quality can be downloaded if more data rate is available. Furthermore, additional active antennas would generate harmful interference which would decrease the available data rate. We only model interference as a static, worst-case value and do not compute it dynamically because it would introduce too much complexity into our model.



(a) Existing scheduler (download) (b) New scheduler (download + BSs)

Figure 2. Example

Before introducing our optimization model, we give a short example how the described scheduling problem works. Figure 2a shows how the existing anticipatory scheduling approach works. In time slot 3 there is a decrease in data rate, so an additional segment has to be downloaded and buffered in time slot 2. This avoids a playback interruption, but additionally available data rates in time slots 1, 5 and 6 is not exploited. Our new approach (shown in Figure 2b) downloads and buffers additional segments in time slots 1, 2 and 5. With this download schedule, the base station (or its antennas) can be switched off in time slots 3, 4 and 6 without causing any playback interruption. For simplicity, the selection of the video quality is omitted in this example.

II. OPTIMIZATION PROBLEM

In this section we describe the optimization problem, which is based on our previous work [7]. Two schedulers, *OptBasic* and *OptFlex*, are implemented with the optimization problem. The input parameters are defined in Table I, and the internal variables in Table II.

The starting point for the optimization problem are the antennas. Since a scenario consists of multiple antennas the

Table I. INPUT PARAMETERS

T	set of time slices, e.g. {0,1,2,3}
S	set of segments to transfer, e.g. {0,1,2,3}
U	set of users
Q	set of qualities, e.g. {10, 20, 50}
A	set of antennas
$DR_{u,a,t} \in \mathbb{Q}^+$	data rate of user u connected to antenna a at time t
SU	scheduling units per antenna and time, e.g. {0, ...99}
$DR^{SU}_{u,a,t} \in \mathbb{Q}^+$	data rate per user u antenna a time t, per scheduling unit

Table II. VARIABLES

$d_{s,t,u} \in \{0,1\}$	deliver segment s at time t to user u
$e_{s,u,q} \in \{0,1\}$	deliver segment s for user u at quality q
$f_{s,u} \in \mathbb{Q}^+$	quality for segment s for user u
$g_{s,t,u} \in \mathbb{Q}^+$	quality for segment s for user u at time t
$l_{s,u} \in \mathbb{N}$	lateness per segment s for user u
$m_{s,u} \in \mathbb{N}$	summed lateness per segment s for user u
$c_{u,a,t} \in \{0,1\}$	connection of user u to Antenna a at time t
$dr_{u,t} \in \mathbb{Q}^+$	data rate per u and time t
$p_{a,t} \in \{0,1\}$	power status of an antenna a at time t
$s_{u,a,t,su} \in \{0,1\}$	scheduling unit su assigned to user u at antenna a and time t

data rate for each user and antenna in each time slot has to be derived. The $DR_{u,a,t}$ input parameter as defined in Table I contains these data rate values. It is important to note that the data rates are a worst case estimation with respect to the interference. First, we compile a list of users that can connect to an antenna by checking if the SINR is above the minimum threshold of valid SINR values, as defined by the GreenTouch radio model [2], [18] we use. Each user then gets an equal share of bandwidth in frequency per antenna where the SINR is above the threshold. The second step is to use the SINR value to look up the spectral efficiency. This value is then multiplied with the bandwidth to derive the actual data rate for each user per time and antenna.

Equation 1 defines that a user can at most be connected to one antenna per time. The available data rate per user and time $dr_{u,t}$ is then defined in Equation 2.

$$\sum_{a \in A} c_{u,a,t} \leq 1, \forall u \in U, t \in T \tag{1}$$

$$dr_{u,t} = \sum_{a \in A} DR_{u,a,t} \cdot c_{u,a,t}, \forall u \in U, t \in T \tag{2}$$

Equation 3 assures that each user can only download each segment once. Equation 4 controls that each segment is only downloaded in one quality. Note that each segment also has to be downloaded once. Which segment is downloaded in which quality is made available for further equations as $f_{s,u}$ in Equation 5.

$$\sum_{t \in T} d_{s,t,u} = 1, \forall s \in S, u \in U \tag{3}$$

$$\sum_{q \in Q} e_{s,u,q} = 1, \forall s \in S, u \in U \tag{4}$$

$$\sum_{q \in Q} e_{s,u,q} \cdot q = f_{s,u}, \forall s \in S, u \in U \tag{5}$$

The time at which a segment at a certain quality is downloaded is defined in Equations 6. A segment can only be downloaded if there is enough available data rate as defined in Equation 7.

$$g_{s,t,u} = f_{s,u} \cdot d_{s,t,u}, \forall s \in S, t \in T, u \in U \tag{6}$$

$$\sum_{s \in S} g_{s,t,u} \leq dr_{u,t}, \forall u \in U, t \in T \tag{7}$$

Uninterrupted playback of a video stream is only possible if all segments are available for playback when they are needed. Equation 8 defines the individual lateness per user and segment $l_{s,u}$. The individual lateness values per user and segment are summed up as $m_{s,u}$ in Equation 9.

$$l_{s,u} = \sum_{t \in [s+1, \max(T)]} d_{s,t,u} \cdot t, \forall s \in S, u \in U \tag{8}$$

$$m_{s,u} = \sum_{x \in [0, s+1]} l_{x,u}, \forall s \in S, u \in U \tag{9}$$

A. Antenna power cycling

So far, all antennas are modeled as always powered. To model whether an antenna is enabled, a new state variable $p_{a,t}$ per antenna and time is needed. These variables are used to constrain which antennas users can connect to. Equation 10 allows connections (variable $c_{u,a,t}$) only to enabled antennas.

$$c_{u,a,t} \leq p_{a,t}, \forall u \in U, a \in A, t \in T \tag{10}$$

The opposite direction needs to be considered too. Equation 11 makes sure that an antenna is disabled if no users are connected.

$$p_{a,t} \leq \sum_{u \in U} c_{u,a,t}, \forall a \in A, t \in T \tag{11}$$

Additionally, users only need to be connected if something is downloaded. The sum of segments has to be rounded up in case segment qualities smaller than 1 are available. This constraint is shown in Equation 12.

$$c_{u,a,t} \leq \lceil \sum_{s \in S} g_{s,t,u} \rceil, \forall u \in U, a \in A, t \in T \tag{12}$$

B. Objective function

The first scheduler, called *OptBasic*, combines all the properties that were introduced so far. We combine the different goals video quality, lateness, and enabled antennas in an objective function which is introduced in this section. Each goal has an individual weight factor to create a trade-off between the different goals. The optimization goals are combined to a single objective function in Equation 13.

$$\begin{aligned} \text{maximize:} &+ 1 \cdot W_q \cdot \sum_{s \in S, u \in U} f_{s,u} && \text{video quality} \\ &- 1 \cdot W_l \cdot \sum_{s \in S, u \in U} m_{s,u} && \text{lateness} \\ &- 1 \cdot W_p \cdot \sum_{a \in A, t \in T} p_{a,t} && \text{enabled antennas} \end{aligned} \tag{13}$$

OptBasic is not a too complex model, but already takes a long time to solve. Still, we wanted to investigate if a more flexible and realistic model for the assignment of data rates to users shows any significant gains and further extended the model for the *OptFlex* scheduler.

C. Flexible data rates

To model the wireless channel resources more realistically we introduce scheduling units (*SUs*). *SUs* are resource allocation units in bandwidth of an antenna to a user, similar to physical resource blocks (PRBs) in LTE.

Due to the *SUs*, the $DR^{SU}_{u,a,t}$ input parameter has to be used differently than $DR_{u,a,t}$. A new internal variable $s_{u,a,t,su}$

is used to model the assignment of SUs to users. Instead of dividing the bandwidth by the number of users, the data rate is calculated as if each user would be the only user connected to an antenna and this data rate is then divided by the number of SUs ($|SU|$). The result is a data rate $DR_{u,a,t}^{SU}$ per user and su which can be used in the linear equations as follows.

The number of scheduling units per antenna and time is limited by the constraint in Equation 14.

$$\sum_{u \in U, su \in SU} s_{u,a,t,su} \leq |SU|, \forall a \in A, t \in T \tag{14}$$

The constraints in Equation 15 and 16 control from which antenna a user can use a SU. To use a SU, a user must be connected to an enabled antenna (Equation 15). If a user cannot use the number of SUs it got assigned, it should not be connected (Equation 16).

$$s_{u,a,t,su} \leq c_{u,a,t}, \forall u \in U, a \in A, t \in T, su \in SU \tag{15}$$

$$\sum_{su \in SU} s_{u,a,t,su} \cdot DR_{u,a,t}^{SU} \geq c_{u,a,t}, \forall u \in U, a \in A, t \in T \tag{16}$$

The data rate per user and time can now be computed by Equation 17. Note that this replaces Equation 2, the definition of a data rate per user and time as used by *OptBasic*.

$$dr_{u,t} = \sum_{a \in A, su \in SU} DR_{u,a,t}^{SU} \cdot s_{u,a,t,su}, \forall u \in U, t \in T \tag{17}$$

III. HEURISTIC ALGORITHM

Because of the complexity and long solving time of the optimization problem, we also implemented a heuristic scheduling algorithm called HSched. It works in two phases: The first phase finds the highest quality per segment that can be downloaded without lateness. Based on this result, all antennas that are not needed for downloads are disabled in the second phase. The results from the two phases are then used to determine both the download schedule for each user (*When* should each segment be downloaded at *which quality*?) and the power cycling schedule (*Which* base station or antenna should be turned on *when*?).

The most important goal of the scheduler is to find a segment quality assignment that does not introduce lateness. The power consumption is only considered afterwards and reduced as much as possible. We now describe the two phases of HSched in Sections III-A and III-B.

A. Quality selection phase

The first step of this phase is to calculate the anticipated available data rate per user, time and antenna $dr_{u,a,t}$.Then, it calls Algorithm 1 for each user. The goal of the algorithm is to find the highest video quality that can be streamed by a user at the anticipated data rate. A list of all possible quality assignments in assigned to *qa_options* in Line 3. The quality assignments are increasing from the last to the first segment. In Line 4, the variable $segment\_demand$ is set to a list in which each value represents the number of segments that have to be downloaded to support the lowest quality. These two data structures are the basis for the following functions.

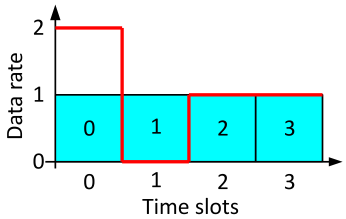


Figure 3. Segment quality and available data rate example. $Duration = 4$ and $Qualities = [1]$, the data rate is defined as the red line. The resulting $segment\_demand$ is {-1:0, 0:2, 1:2, 2:3, 3:4}.

The crucial problem of this algorithm is to define in which quality each segment can be downloaded while keeping the lateness low and the average quality high. While defining the quality, it also generates a list of antennas for each time slot which provide sufficient data rate for the segment downloads.

The algorithms solves the problem by checking all possible quality assignments, starting from the lowest to the highest by the use of a fast validation loop over all options (Line 8ff.). The content of the *qa_options* list in Line 3 is as follows for a scenario with $Qualities = [1, 2]$ and $Duration = 2$: $[[1, 1], [1, 2], [2, 1], [2, 2]]$.

Algorithm 1 QUALITYASSIGMENT$(u, dr_{u,a,t})$

```
 1: // $Qualities$, $Duration$ from Scenario
 2: $validOption \leftarrow none$
 3: $qa\_options \leftarrow$ QUALITYOPTIONSINCR($Qualities, Duration$)
 4: $segment\_demand \leftarrow$ SEGMENTDEMAND($u, dr_{u,a,t}$)
 5: for all $qa \in qa\_options$ do
 6:   $antennas \leftarrow$ LIST()
 7:   $dl\_c \leftarrow 0$ // Count of downloaded segments
 8:   for $(t \leftarrow 0; t < Duration; t \leftarrow t + 1)$ do
 9:     $ants \leftarrow$ SUFFANTENNAS($t, dl\_c, qa, segment\_demand$)
10:     if $len(ants)! = 0$ then
11:       $dl\_c \leftarrow dl\_c +$ MINDOWN($t, dl\_c, qa, ants$)
12:       $antennas.append(ants)$
13:     else
14:       return $validOption$
15:     end if
16:   end for
17:   $validOption \leftarrow (qa, antennas)$
18: end for
```

1) The `segment_demand()` *function:* The semantics of the result of this function are explained with the example in Figure 3 and the implementation is shown in Algorithm 2. The parameters of the example scenario are $Duration = 4$ and $Qualities = [1]$. Only one antenna is available. Available data rate per time is given by the red line. The blue boxes 0–3 represent the data rate demand of the lowest quality min_q in the algorithm (line 2).

The *segment_demand* of the last time interval is initialized with the *Duration* of the scenario (line 3). This means that, if all segments (of which there are *Duration* many) are downloaded, the schedule has no playback interruptions. For $t = 3$, there is enough data rate available to download segment three. Therefore $segment\_demand[2] = 3$ (line 14) and segment 3 is added and then removed from the list of $late\_segments$ (lines 6 and 13).

Algorithm 2 SEGMENTDEMAND(u, $dr_{u,a,t}$)

```
1: // $Qualities$, $Duration$ from Scenario
2: $min\_q \leftarrow \min(Qualities)$
3: $segment\_demand[Duration - 1] \leftarrow Duration$
4: $late\_segments \leftarrow List()$
5: for ($t \leftarrow Duration - 1; t \geq 0; t \leftarrow t - 1$) do
6:     $late\_segments.append(t)$
7:     $dr \leftarrow \max_{a \in Antennas}(dr_{u,a,t})$
8:     $count \leftarrow 0$
9:     while ($dr \geq min\_q$)and($count \leq len(late\_segments)$) do
10:        $dr \leftarrow dr - min\_q$
11:        $count \leftarrow count + 1$
12:    end while
13:    $late\_segments \leftarrow late\_segments[count:]$
14:    $segment\_demand[t - 1] \leftarrow segment\_demand[t] - count$
15: end for
16: return $segment\_demand$
```

In time $t = 2$, the situation is the same as in $t = 3$. Sufficient data rate is available, thus $segment\_demand[1] = 2$. In $t = 1$, the data rate is not sufficient to download the lowest quality.

If the segment cannot be downloaded in an earlier time slot, the lateness would increase. This information is kept by settings $segment\_demand[0] = 2$, as in $segment\_demand[1]$. Additionally, segment 1 is kept in the list of late segments, due to $count = 0$ (line 13). At $t = 0$ sufficient data rate for two segments is available and segment 1 and 2 are in the list of $late\_segments$. Therefore $count = 2$ after the while loop, and both segments are removed from the list *late_segments*. $segment\_demand[-1] = 0$ is set. This indicates that there are schedules for this scenario that introduce no lateness.

Also, at each time t, where $0 \leq t \leq Duration$, if segments 0, 1, ..., *segment_demand[t]* are downloaded, at least one lateness-free schedule can still be found.

2) Validation loop: In the validation loop in Algorithm 1 (line 5ff), previously computed possible quality assignments are tested. First, the list of usable antennas and the counter of downloaded segments are initialized (line 6 and 7). Then, for all time slots, it is checked whether there are antennas available that provide sufficient data rate. The invariant is that *dl_count* is always greater or equal to $segment\_demand[t]$. This check is done in line 11.

SUFFANTENNAS() downloads as many segments from the *qa[dl_count:]* sub-list in the order of appearance in the list per antennas as possible. If this number of newly downloaded segments added to the *dl_count* of previously downloaded segments is greater or equal to $segment\_demand[t]$, then the antenna is returned. Additionally, *none* is added to the list of antennas if *dl_count* is greater or equal to $segment\_demand[t]$ without new downloaded segments.

If SUFFANTENNAS() does not return any antennas then there is no solution for this quality assignment and the given antennas. This validation approach does not find all possible antenna combinations for a given schedule. This is due to the fact that *dl_count* is only increased by the minimum number of segments that all sufficient antennas can download.

As soon as a quality assignment is not valid, the last successfully tested one is returned together with the list of antennas per time (line 14).

3) Returned values: In the end of this phase the returned values from QUALITYASSIGNMENT() (Algorithm 1) are passed on to the next phase as two sets of variables $q_{u,s}$ and $as_{u,t}$, where $q_{u,s}$ is the best valid quality assignment per user u for segment s and $as_{u,t}$ the antennas which can fulfill the quality assignment $q_{u,s}$ per user u and time t, as described in Section III-A2.

B. Antenna disabling phase

The second phase disables unneeded antennas. The previous phase decided in which quality which segment can be downloaded and from which potential antennas at each point in time. An assignment of users per time to an exact antenna is needed for a valid schedule. Options of antennas per time can be removed until as few antennas as possible are left. HSched sequentially uses three strategies to disable antennas.

1) Unused antennas: The first strategy searches for completely unused antennas and disables them. It iterates over all possible time slots and checks per time slot for each antenna if any user can use it according to $as_{u,t}$. If not, the antenna is added to the list of disabled antennas for that time slot.

2) Weak antennas: The second strategy removes those antennas that provide less data rate than others. It achieves this by creating a list of all users from $as_{u,t}$ that can connect to an antenna per time slot. The data rates of those users are then summed per antenna and sorted in increasing order. The algorithm continues with trying to disable antennas for that time slot. At least one antenna must remain usable per user. If the user does not need to be connected, then this antenna option is kept as well.

3) Segment buffering: The third strategy tries to disable antennas if their data rate is not essential. Not essential means that the data rate to download segments in this time slot is also available in previous time slots. Thus the segments can also be downloaded earlier and the data rate is not needed to fulfill the $sd_{u,t}$ requirement.

IV. EVALUATION

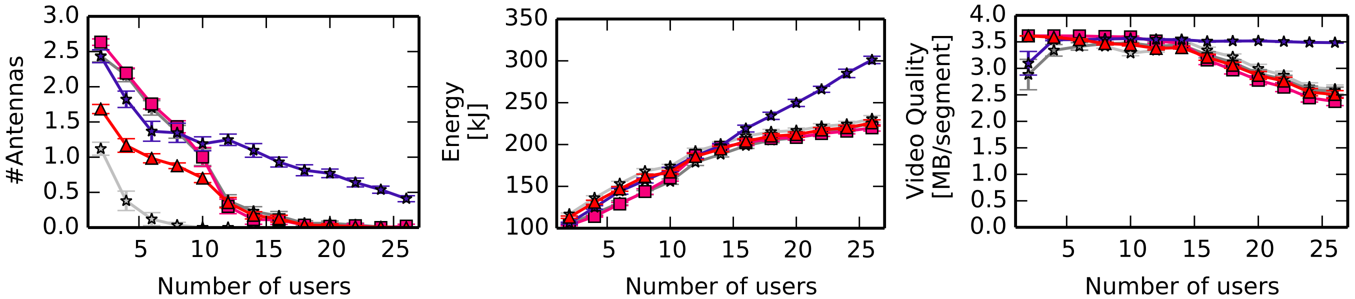
Our evaluation is twofold: First, we compare the optimization problem, the heuristic algorithm and existing algorithms [7] in a small scenario. Second, we compare the heuristic algorithm with an existing algorithm based on the current behavior of the VLC media player [6].

A. Scenarios

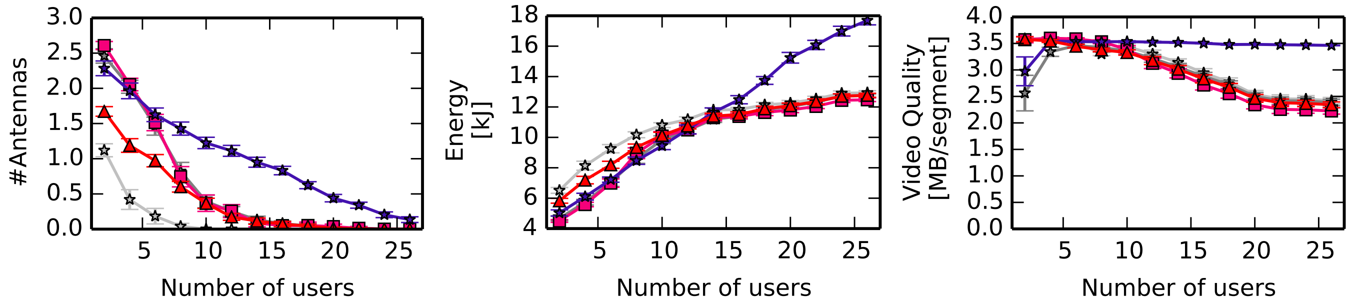
Both evaluations use the radio and power models from the GreenTouch project [2], [18]. Video qualities are set to 14.14 MB for low, 29.51 MB for medium and 36.13 MB for high quality.

The first scenario consists of three base stations placed in a line and a variable number of stationary users uniformly placed between them. We use this scenario once with macro BSs (one undirected sector, 8x2 MIMO, rural environment, 80% of rural ISD: 3464 m) and once with pico BSs (ISD 200 m) according to the GreenTouch model.

For the second scenario we simulate a train ride on a regional train: A group of users moves along a line with several



(a) Disabled antennas (macro) (b) Energy usage (macro) (c) Average video quality (macro)



(d) Disabled antennas (e) Energy usage (f) Average quality

Figure 4. Three-BSs scenario for both macro [(a)-(c)] and pico BSs [(d)-(f)]

stations in between where the train stops. Based on a real train schedule[1] in Germany, the train moves a certain time from one station to the next station and then stops there for one or two minutes. Each station is equipped with a pico base station that can be turned on to allow the users on the train to download and buffer more video segments.

We compare five different schedulers in this evaluation:

- *OptBasic* is the optimization problem without flexible data rate assignment (Section II-A)
- *OptFlex* is the optimization problem with flexible data rate assignment (Section II-C)
- *HeuristicScheduler* (HSched) is the heuristic scheduler (Section III)
- *OptLegacy* is an existing optimization problem [7] without explicit power cycling. Base stations are initially all switched on and only base stations with no associated users are considered switched off in a post processing step.
- *GreedyLegacy* represents the current behavior of a video player (VLC [6]) without any optimization. Power cycling is also only considered in a post processing step.

All plots show confidence intervals at 95% confidence level unless they are covered by the plot markers.

B. *Three-BSs Scenario Results*

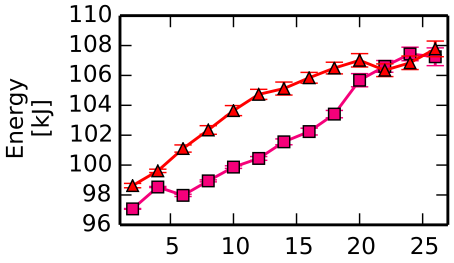
The most interesting result in this scenario is the number of disabled antennas as shown in Figures 4a and 4d. In both the macro and pico scenarios *OptFlex* can disable the overall highest number of base stations because it can schedule with the highest degree of freedom by being able to flexibly assign data rates to users. It is closely followed by the *OptBasic* and *HeuristicScheduler* which perform equally. With more than 15 users both schedulers are not able to disable any antenna as the load is too high. With *GreedyLegacy* a few antennas can be disabled for a low number of users because this scheduler will buffer segments and does not have to download a new segment in each time slot. *OptLegacy* performs worst as it tries to minimize buffering of segments and downloads segments in almost all time slots without considering energy.

As the power consumption of a base station is not only influenced by the enabled antennas but also depends on the traffic load, the overall energy usage varies as shown in Figures 4b and 4e. For a low number of users *OptBasic* and *HeuristicScheduler* achieve a slightly lower energy usage than the other schedulers. For a higher number of users *OptFlex* uses significantly more energy. This is because this scheduler transfers more data as it uses a higher video quality (Figures 4c and 4f). Considering both metrics the energy used per transferred bit is in the same range for all schedulers.

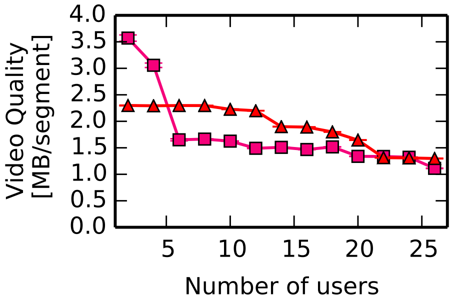
The overall energy usage if no power cycling scheme is employed is 399 kJ for macro and 19.8 kJ for pico. Thus all presented power cycling schemes are able to reduce the energy usage.

---

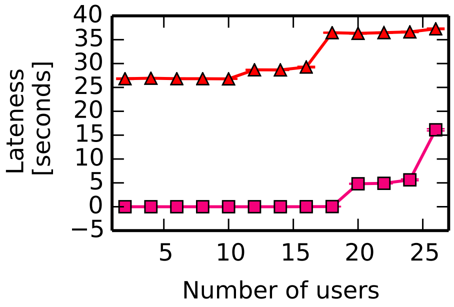
[1] Kursbuchstreckennummer 405, see http://kursbuch.bahn.de



(a) Energy usage



(b) Average Quality



(c) Average Lateness

Figure 5. Train scenario results

C. Train Scenario Results

As the train scenario is too big to solve with the optimization-based schedulers, we only compare *Heuristic-Scheduler* and *GreedyLegacy*. For up to 20 users *Heuristic-Scheduler* uses significantly less energy (Figure 5a). For a very low number of users *HeuristicScheduler* delivers a higher video quality, but for more users it drops slightly below the quality delivered by *GreedyLegacy* (Figure 5b). This drop in video quality is mitigated by the significantly lower occurrence of video interruptions (lateness) with *HeuristicScheduler* (Figure 5c).

These results show that our algorithm also works in a practical scenario and is able to reduce the energy usage by maintaining a satisfying QoE for the users. The overall energy consumption without any power cycling scheme would be 522.72 kJ.

V. CONCLUSION

Our results show that combining the existing approach for anticipatory video streaming with a new approach for power cycling base stations reduces the overall energy consumption of the base stations while maintaining a high QoE for the users. In the small Three-BSs scenario the energy usage is reduced by 40 to 75% and in the train scenario by around 80% compared to a case with no power cycling scheme.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 318115.

REFERENCES

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2012-2017," 2013.

[2] GreenTouch, "GreenTouch Green Meter Research Study: Reducing the Net Energy Consumption in Communications Networks by up to 90% by 2020," 2013.

[3] R. Pantos, W. May, and Apple Inc., "HTTP Live Streaming," http://tools.ietf.org/html/draft-pantos-http-live-streaming-11, April 2013.

[4] I. Sodagar, "The MPEG-DASH Standard for Multimedia Streaming Over the Internet," *IEEE Multimedia*, vol. 18, no. 4, pp. 62–67, Apr. 2011.

[5] International Organization for Standardization, "ISO/IEC 23009-1: Dynamic adaptive streaming over HTTP (DASH) Part 1: Media presentation description and segment formats," 2012.

[6] VideoLAN Organization, "VideoLAN," http://www.videolan.org/videolan/.

[7] M. Dräxler and H. Karl, "Cross-layer scheduling for multi-quality video streaming in cellular wireless networks," in *Proceedings of the 9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, 2013.

[8] M. Dräxler, J. Blobel, P. Dreimann, S. Valentin, and H. Karl, "Anticipatory Buffer Control and Quality Selection for Wireless Video Streaming," *arXiv preprint arXiv:1309.5491v1*, 2013.

[9] N. Bui, F. Michelinakis, and J. Widmer, "A Model for Throughput Prediction for Mobile Users," in *Proceedings of European Wireless 2014*, 2014.

[10] S. Sadr and S. Valentin, "Anticipatory Buffer Control and Resource Allocation for Wireless Video Streaming," *arXiv preprint arXiv:1304.3056v1*, 2013.

[11] K. Evensen, A. Petlund, H. Riiser, P. Vigmostad, D. Kaspar, C. Griwodz, and P. l. Halvorsen, "Mobile video streaming using location-based network prediction and transparent handover," in *Proceedings of the 21st international workshop on Network and operating systems support for digital audio and video - NOSSDAV '11*, 2011.

[12] H. Riiser, T. Endestad, P. Vigmostad, C. Griwodz, and P. Halvorsen, "Video streaming using a location-based bandwidth-lookup service for bitrate planning," *ACM Transactions on Multimedia Computing, Communications, and Applications*, vol. 8, no. 3, pp. 1–19, Jul. 2012.

[13] R. Radhakrishnan and A. Nayak, "Cross layer design for efficient video streaming over LTE using scalable video coding," in *Proceedings of the IEEE International Conference on Communications (ICC)*, 2012.

[14] M. Bennis, M. Simsek, A. Czylwik, W. Saad, S. Valentin, and M. Debbah, "When cellular meets WiFi in wireless small cell networks," *IEEE Communications Magazine*, vol. 51, no. 6, Jun. 2013.

[15] A. J. Fehske, F. Richter, and G. P. Fettweis, "Energy Efficiency Improvements through Micro Sites in Cellular Mobile Radio Networks," in *Proceedings of the IEEE Globecom Workshops*, 2009.

[16] F. Richter, A. J. Fehske, and G. P. Fettweis, "Energy Efficiency Aspects of Base Station Deployment Strategies for Cellular Networks," in *Proceedings of the IEEE 70th Vehicular Technology Conference Fall*, 2009.

[17] O. Blume, H. Eckhardt, S. Klein, E. Kuehn, and W. M. Wajda, "Energy savings in mobile networks based on adaptation to traffic statistics," *Bell Labs Technical Journal*, vol. 15, no. 2, pp. 77–94, Aug. 2010.

[18] GreenTouch, "Mobile Communications WG Architecture Doc2: Reference scenarios," 2013.