Anticipatory Admission Control and Resource Allocation for Media Streaming in Mobile Networks

Nicola Bui[12], Ilaria Malanchini[3], Joerg Widmer[1]
[1]IMDEA Networks Institute, Leganes (Madrid), Spain
[2]UC3M, Leganes (Madrid), Spain
[3]Bell Labs, Alcatel-Lucent (Stuttgart), Germany

ABSTRACT

The exponential growth of media streaming traffic will have a strong impact on the bandwidth consumption of the future wireless infrastructure. One key challenge is to deliver services taking into account the stringent requirements of mobile video streaming, e.g., the users' expected Quality-of-Service. Admission control and resource allocation can strongly benefit from the use of anticipatory information such as the prediction of future user's demand and expected channel gain. In this paper, we use this information to formulate an optimal admission control scheme that maximizes the number of accepted users into the system with the constraint that not only the current but also the expected demand of all users must be satisfied. Together with the optimal set of accepted users, the optimal resource scheduling is derived. In order to have a solution that can be computed in a reasonable time, we propose a low complexity heuristic. Numerical results show the performance of the proposed scheme with respect to the state of the art.

Categories and Subject Descriptors

C.2.3 [**Computer Systems Organization**]: Networks—*Network management*

General Terms

Prediction, Resource Allocation, Admission Control, Mobile Networks

Keywords

Anticipatory networking; Optimization; Multi-user

1. INTRODUCTION

Many factors contribute to the exponential growth of mobile traffic and multimedia contents will be the dominant component among the causes of this growth, e.g. [1, 28]. In this paper we investigate prediction based media streaming in mobile networks and we discuss admission control and resource allocation.

The quality of a media stream is characterized by the following key performance indicators (KPIs) [10]: (i) streaming continuity and (ii) average stream quality. The former is assumed to have higher priority, since in general interruptions may jeopardize the comprehension of the content and therefore are perceived as the worst quality degradation. The latter is optimized with lower priority, since, even if it has a weaker impact on user's perception, users appreciate when a certain agreed quality-of-service (QoS) is guaranteed. In this paper we consider it to be directly proportional to the stream bitrate [19].

An additional characteristic of prediction based optimization is that the prediction reliability varies in time and, usually, decreases as the prediction horizon length grows [7]. Therefore, anticipatory optimization schemes should consider this either explicitly in the problem formulation [18] or evaluate the impact of prediction error a posteriori [3]. Here we focus on joint admission control and resource allocation with perfect system state prediction to obtain upper bounds on the achievable gains. The extension to imperfect knowledge (e.g. [9]) is left for future work.

We follow a lexicographic approach where, first, we maximize the number of users that are served with guaranteed QoS for the whole duration of the media stream, minimizing the total interruption time, and maximizing the streaming quality. Thus, the streaming requests that cannot be scheduled with guaranteed quality must wait for the system to have enough resources for them to start streaming. Furthermore, we assume that it is always preferable to admit a new user in the system than increasing the quality of a user who is already admitted and the streaming continuity is always preferred to extra quality.

The contributions of our work are the following:

- mixed integer linear program (MILP) formulation of the joint admission control and resource allocation problem;
- online algorithm based on linear programming (LP) and binary search that allows for a very fast solution computation;
- trace-based simulation discussing optimality and complexity of the proposed approach as well as the system performance.

We validate our approach using trace based simulation obtained from real measurement data collected by the MOMENTUM project [13] in Berlin. We show that our online

MSWiM'15, November 2–6, 2015, Cancun, Mexico.

DOI: http://dx.doi.org/10.1145/2811587.2811604.

solution closely approximates the results achieved by the MILP formulation and dramatically reduces the computational time.

The rest of the paper is structured as follows: section 2 reviews the state of the art on anticipatory networking solutions, section 3 introduces the mathematical notation and the optimization problem, section 4 describes our proposed approximate solution, section 5 illustrates our evaluation campaign, and section 6 provides our conclusions.

2. RELATED WORKS

Anticipatory optimization techniques are motivated by a series of seminal papers, such as [23, 26], which discuss the predictability of human mobility patterns and the link between mobility and communication. Shafiq et al. [26] studied mobile network traffic and its spatio-temporal correlation with mobility patterns. Similarly, Ahmed et al. [4] studied network user habits in terms of content: the study links content requests and user categories, aiming to their prediction.

The predictability of network capacity and the achievable rate of mobile users have been extensively studied in the literature. These studies range from short term prediction using filtering techniques [24, 25], to medium and long term forecasting solutions [12, 20] accounting for position and trajectory estimates. We contributed to the literature with a general model [7] for predicted rates in mobile networks accounting for prediction uncertainties, and we use the model to devise single user optimal resource allocation policies [9].

For what concerns the state of the art on prediction based network optimization, in what follows we review a few of the papers that are more closely related to our current work.

Majid et al. [17] and Koutsakis et al. [16] exploited medium-long term average prediction of the users' achievable rate to devise call admission control and resource allocation techniques, respectively. While the former is more focused on DiffServ system [6], the latter addressed specifically multimedia traffic in broadband mobile networks. The present work differs from these early papers as well as more recent approaches [27], since we exploit rate fluctuations on a shorter time scale instead of using averages.

More recently, Dräxler and Karl [11] tackled multimedia traffic optimization by devising a different problem formulation that considered an objective function that combined stream interruption time and average quality. The proposed schemes choose when to download a given content segment and at which quality among a discrete set of qualities. In this paper we obtain a simpler formulation by considering continuous quality and by means of approximations. This allows us to include in our objective function both admission control and resource allocation.

Abou-zeid et al. [2, 3] develop a MILP formulation of a similar problem to obtain an optimal resource allocation and to increase energy efficiency. As other prior work, these papers do not consider admission control and thus they cannot enforce Quality-of-Service in the system.

A different approach is taken in [15] and [8], which study different algorithms to solve the resource allocation problem. These approaches aim at finding practical solutions that do not require commercial solvers and can execute in real-time even with non-linear objective functions. In addition, complete solutions, such as [18], integrate prediction techniques and optimization algorithms to solve the resource allocation problem or study optimal video transcoding [5] for admission control and scheduling.

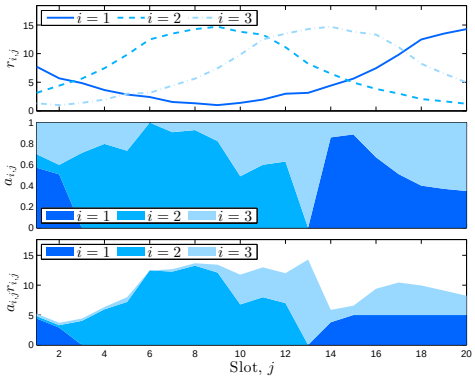


Figure 1: An example of achievable rates $r_{i,j}$ (top), assignments $a_{i,j}$ (center) and obtained rates $a_{i,j}r_{i,j}$ (bottom) in a 3-user scenario.

Compared to the aforementioned solutions, this paper proposes a different perspective of the network optimization problem as we enforce QoS by means of admission control. In addition, we propose low-complexity solutions that can be used for online optimization, which require the output to be updated within a short time.

3. PROBLEM DEFINITION

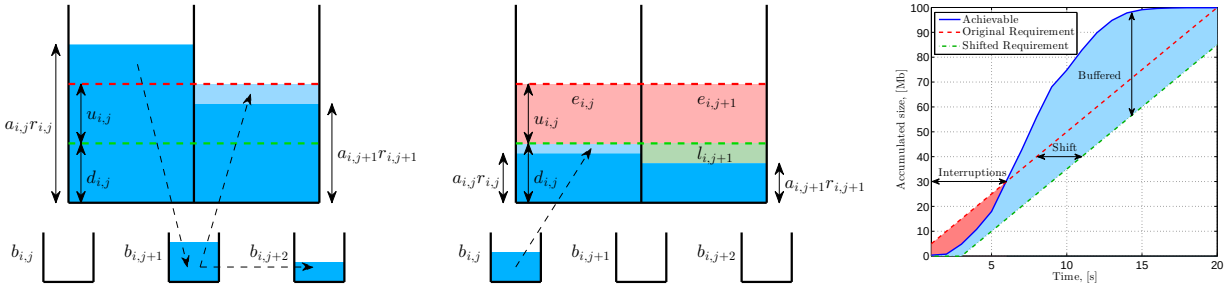
The admission control and resource allocation problem can be modeled as a centralized decision making problem, where a set $\mathcal{N}$ of N users share a given quantity of network resources. Prediction is assumed to be perfect over a set $\mathcal{T}$ of T time slots. In the following, we consider slot duration $t = 1$, thus data rate and download size can be used interchangeably. In the rest of the paper we use the following assumptions: (a) the future knowledge is perfect and (b) the average video bitrate is continuous between 0 and q_M (e.g., by averaging over segments of different quality [22]).

We consider the following input parameters, all of which defined for each user $i \in \mathcal{N}$ and slot $j \in \mathcal{T}$:

- Predicted achievable download rate $r_{i,j} \in [0, r_M]$ is the prediction of the rate a user would achieve if no other user is scheduled. r_M is the maximum achievable data rate.
- Minimum requirement $d_{i,j} \in [0, q_M]$ is the minimum amount of bytes needed in a given slot to stream the content at the minimum bitrate with no interruptions.
- Maximum extra video bitrate $u_{i,j} \in [0, q_M]$, is the maximum amount of additional bytes that can be used in a given slot to obtain the maximum content bitrate.

The problem is characterized by the following variables:

- Resource assignment $a_{i,j} \in [0, 1]$ represents the average fraction of resources assigned to user i in slot j. In each slot, each user can be assigned at most the total available rate, $0 \leq a_{i,j} \leq 1$, and the sum cannot exceed the total available resources, $0 \leq \sum_{i \in \mathcal{N}} a_{i,j} \leq 1$. Figure 1 shows an example with $N = 3$ and $T = 20$. In the top graph the achievable rates are plotted independently. In the center plot, a possible resource assignment is visualized by stacking the fraction of resources assigned to each of the users $a_{i,j}$ on top of each other. In the bottom graph, the cell capacity variation is addressed by stacking the product of the achievable rate and the fraction of assigned resources $a_{i,j}r_{i,j}$.



(a) Buffer (b) Outage (c) Shift

Figure 2: Three examples of the system quantities: 2(a) exemplifies the buffer usage over two subsequent slots; 2(b) shows lateness and extra quality outage examples; 2(c) illustrates the impact of pre-buffering.

• Buffer state $b_{i,j} \in [0, b_M]$ tracks the amount of bytes stored in the buffer and b_M is the buffer size in bytes.
• Pre-buffering time (or waiting time) $w_{i,k} \in \{0,1\}$ with $k \in \{1,\dots,T+1\}$ defines when the actual playing of the content starts: there must be a single starting point ($\sum_{k=1}^{T+1} w_{i,k} = 1, \forall i \in \mathcal{N}$). Thus user i will wait for $W_i = (\text{argmax}_k w_{i,k}) - 1$ slots where she can only fill the buffer. This waiting implies the requirement sequence has to be shifted to later slots. Thus, in slot j user i is obtaining the rate $a_{i,j}r_{i,j}$ and should satisfy the shifted requirements $\overrightarrow{d_{i,j}} = \sum_{k=1}^{T+1} D_{i,j,k} w_{i,k}$ and $\overrightarrow{u_{i,j}} = \sum_{k=1}^{T+1} U_{i,j,k} w_{i,k}$, where D and U are $N \times T \times T+1$ matrices whose vectors $\mathbf{d_{i,k}} = \{\mathbf{0_{k-1}}, d_{i,1}, \dots, d_{i,T-k}\}$ and $\mathbf{u_{i,k}} = \{\mathbf{0_{k-1}}, u_{i,1}, \dots, d_{i,T-k}\}$ are shifted versions of the original requirements, where we used bold fonts to identify vectors and $\mathbf{0_k}$ is a null vector of size k.
• Interruption time[1] (or lateness) $l_{i,j} \in [0, q_M]$ is the missing data to fulfill the minimum content requirement $\overrightarrow{d_{i,j}}$:

$$l_{i,j} = [\overrightarrow{d_{i,j}} - b_{i,j} - a_{i,j}r_{i,j}]_0^{\overrightarrow{d_{i,j}}} \tag{1}$$

where $[x]_a^b = \min\{\max\{x, a\}, b\}$ is a bounding operator that forces the undelivered quantity to be greater than zero and smaller than the requirement in the slot.
• Extra quality outage $e_{i,j} \in [0, q_M]$ is the amount of data missing to obtain the content at the maximum bitrate $\overrightarrow{u_{i,j}}$,

$$e_{i,j} = [\overrightarrow{u_{i,j}} + \overrightarrow{d_{i,j}} - l_{i,j} - b_{i,j} - a_{i,j}r_{i,j}]_0^{\overrightarrow{u_{i,j}}}. \tag{2}$$

Figure 2(a) provides a graphical example of the buffer usage for a single user over two subsequent slots. Starting from an empty buffer, the obtained rate $a_{i,j}r_{i,j}$ is used to satisfy the current requirements and to buffer content for the next slot. The light area of the second slot highlights the fraction of content that has been previously buffered. Whether the buffer contains data to guarantee continuous streaming or extra quality is a key decision in the system and plays a critical role in the following optimization.

Figure 2(b) shows a two slot example where the user does not obtain a rate sufficient to satisfy the requirements: in the first slot this is compensated by the buffer, but this is not possible in the second slot resulting in an interruption of the streaming. Thus, the figure shows in light red the quality outage and in light green the missing minimum requirements in the second slot.

Figure 2(c) shows the cumulative download size and requirements according to the second user of the example of Figure 1: a waiting time $w_2 = 3$ moves the original requirements (red dashed line) towards the right by 3 slots (green dot-dashed line), avoiding streaming interruptions in the first six slots (red area between the original requirements and the obtained rates, blue solid line). Since content duration can be longer than T, a non-empty buffer is required at the end of the optimization window: in particular, we require the buffer to contain the minimum between the initial amount and the remaining size of the content.

In each slot j user i receives $a_{i,j}r_{i,j}$, which can be used either to satisfy the requirements in the current slot or to fill the buffer for later use. Thus we can write the following equation that describes the next buffer state:

$$b_{i,j+1} = b_{i,j} + a_{i,j}r_{i,j} - \overrightarrow{d_{i,j}} + l_{i,j} - \overrightarrow{u_{i,j}} + e_{i,j} \tag{3}$$

which means the buffer of user i in slot $j+1$ is obtained from the previous buffer $b_{j,i}$ by adding the received data $a_{i,j}r_{i,j}$ and subtracting the minimum requirements $\overrightarrow{d_{i,j}} - l_{i,j}$ and the extra quality $\overrightarrow{u_{i,j}} - e_{i,j}$[2]. Finally, we define $b_{i,0}$ as the initial status of the buffer of user i.

In addition, we introduce two KPIs that we will use to build the objective function for our problem. Namely, we define the fraction of continuous streaming time $\lambda_i \in [0,1]$ and the fraction of the extra quality obtained $\theta_i \in [0,1]$ as:

$$\lambda_i = \frac{1}{T} \sum_{k \in \mathcal{T}} \left(1 - l_{i,k} \overrightarrow{d'_{i,k}}\right) \tag{4}$$

$$\theta_i = \frac{1}{T} \sum_{k \in \mathcal{T}} \left(1 - e_{i,k} \overrightarrow{u'_{i,k}}\right), \tag{5}$$

where

$$\overrightarrow{d'_{i,j}} = \begin{cases} 1/\overrightarrow{d_{i,j}} & \overrightarrow{d_{i,j}} > 0 \\ 0 & \overrightarrow{d_{i,j}} = 0 \end{cases}$$
$$\overrightarrow{u'_{i,j}} = \begin{cases} 1/\overrightarrow{u_{i,j}} & \overrightarrow{u_{i,j}} > 0 \\ 0 & \overrightarrow{u_{i,j}} = 0 \end{cases}. \tag{6}$$

[1]Since receiving less data than the minimum requirement causes an interruption in the streaming, we use the effect instead of the cause to define this quantity. However, the actual interruption time is the ratio between missing and minimum requirement in a slot.

[2]Normalization between rates in a slot and amount of data is not required, because we assumed the slot length $t = 1$.

Note that when $\overrightarrow{d_{i,j}} = 0$ ($\overrightarrow{u'_{i,j}} = 0$) the interruption time $l_{i,j}$ (the extra quality outage $e_{i,j}$) is necessarily equal to 0, hence the substitutions of Eq. (6) are consistent.

In order to guarantee a given QoS we consider two constraints, the minimum continuous play time λ_i^* and the minimum average quality θ_i^*, defined so that $\lambda_i \geq (T-W_i)\lambda_i^*/T$ and $\theta_i \geq (T-W_i)\theta_i^*/T$.

These constraints can be seen as contractual agreements that must be enforced while the content is being streamed and they change the optimization problem from a best effort resource allocation solutions where the KPIs are maximized to a joint admission control and resource allocation approach where quality of service can be guaranteed.

Finally, we build our objective function to, in order of decreasing importance, (*i*) minimize the aggregate waiting time of the system ($\sum_{k\in\mathcal{N}} W_k$), (*ii*) maximize the total continuous streaming time ($\sum_{k\in\mathcal{N}} \lambda_k$) and (*iii*) maximize the total extra quality ($\sum_{k\in\mathcal{N}} \theta_k$). Consequently, we obtain the following **MILP formulation**:

$$\begin{aligned}
\underset{A,B,L,E,W}{\text{maximize}} \quad & \sum_{k\in\mathcal{N}} (K(\lambda_k - W_k) + \theta_k) \qquad (7)\\
\text{subject to:} \quad & a_{i,j} \geq 0; \ \sum_{k\in\mathcal{N}} a_{k,j} \leq 1\\
& \lambda_i \geq (T-W_i)\lambda_i^*/T; \ \ \theta_i \geq (T-W_i)\theta_i^*/T\\
& l_{i,j} \geq 0; \ \ e_{i,j} \geq 0; \ \ b_{i,j} \leq b_M\\
& l_{i,j} \geq \overrightarrow{d_{i,j}} - a_{i,j}r_{i,j} - b_{i,j}\\
& e_{i,j} \geq \overrightarrow{u_{i,j}} - a_{i,j}r_{i,j} - b_{i,j} + \overrightarrow{d_{i,j}} - l_{i,j}\\
& \forall i \in \mathcal{N}; j \in \mathcal{T}\\
& \text{Eqns. (3), (4) and (5).}
\end{aligned}$$

Eqns. (1-2) have been properly replaced by linear form. Note that the objective function is a linear combination of three components: $W_k \in \{0, 1, \ldots, T\}$, $\lambda_k \in [0, 1]$ and $\theta_k \in [0, 1]$, of which the first two are multiplied by $K > 1$. Since $\sum_{k\in\mathcal{N}} W_k \in \{0, \ldots, NT\}$, while $\sum_{k\in\mathcal{N}} \lambda_k \in [0, 1]$ and $\sum_{k\in\mathcal{N}} \theta_k \in [0, 1]$, the minimization of the waiting time is always addressed first in the problem.

Thus, the solver assign resources so that as many users as possible obtain the required λ_i^* and θ_i^*. The weight K ensures that the solver's second priority is the continuous streaming time: ideally for $K \to \infty$ the solution would never choose quality over continuous streaming, but in practice it is sufficient to set $K \gg 1$ as $\max\{\lambda_i\} = \max\{\theta_i\} = 1$.

Having the three quantities in the objective function accommodates all possible scenarios: for instance, if the sum of the achievable rates is very large compared to the sum of requirements, the solution is likely to obtain no waiting time and continuous streaming for all users and the objective function will assign resources to maximize the extra quality.

When all users need some pre-buffering, the objective function will first use resources to reduce the waiting time and then to improve the continuous streaming.

The granularity of the waiting times W_i may leave unused resources between the best solution and the next, unfeasible, value of the objective function. These saved resources can be used to either improve users' λ or θ, whereas they cannot decrease the total waiting time.

4. ONLINE ALGORITHM

A few preliminary tests showed that the MILP formulation of Eq. (7) is too complex (i.e. solvers need too much time) for online operations. The reasons are mainly two: MILP formulations are inherently combinatorial and the dimensionality of the problem is proportional to T^2N due to the three-dimensional matrices D and U, introduced to account for requirements shift. In this section we reduce the formulation complexity in two steps:

- first, we decrease the problem dimensionality from T^2N to TN by replacing waiting times with admission control variables;
- subsequently, to remove the combinatorial aspect of the MILP formulation, we approximate it with a simpler LP approach;
- finally, we perform a binary search over a sorted list of the users to find the largest set of users for which the LP formulation is feasible.

Reduced MILP formulation: to reduce the dimensionality of the problem caused by shifting the requirement sequences according to the waiting time W_i, we introduce a binary variable s_i, representing whether a user is admitted or not in the current optimization windows: $s_i \in \{0, 1\}$, $i \in \mathcal{N}$, where $s_i = 1$ if user i is admitted. Users who are admitted start streaming the content immediately (i.e. $W_i = 0$) and must fulfill both QoS conditions (λ_i^* and θ_i^*) for the whole content duration. Users that are not immediately admitted can only pre-buffer data if resources are still available. We obtain the following reduced MILP formulation:

$$\begin{aligned}
\underset{A,B,L,E,S}{\text{maximize}} \quad & \sum_{k\in\mathcal{N}} (K(\lambda_k + s_k) + \theta_k) \qquad (8)\\
\text{subject to:} \quad & a_{i,j} \geq 0; \ \sum_{k\in\mathcal{N}} a_{k,j} \leq 1\\
& \lambda_i \geq \lambda_i^* s_i; \ \ \theta_i \geq \theta_i^* s_i\\
& l_{i,j} \geq 0; \ \ e_{i,j} \geq 0; \ \ b_{i,j} \leq b_M\\
& l_{i,j} \geq d_{i,j} - a_{i,j}r_{i,j} - b_{i,j}\\
& e_{i,j} \geq u_{i,j} - a_{i,j}r_{i,j} - b_{i,j} + d_{i,j} - l_{i,j}\\
& \forall i \in \mathcal{N}; j \in \mathcal{T}\\
& \text{Eqns. (3), (4) and (5),}
\end{aligned}$$

where we replaced the shifted requirements with the original ones (Eq. (3-5) should be modified accordingly). We observe that the constraints on λ_i and θ_i are only activated if $s_i = 1$. In fact, if user i is not admitted ($s_i = 0$) the constraint becomes $\lambda_i \geq \lambda_i^* - (1 - s_i) = 0$, thus the problem accepts any value for λ_i, which means users that are not admitted can still obtain resources, but they can only pre-buffer data without playing the actual content.

In addition, the term $\lambda_k + s_k$ in the objective function has a discontinuity in $\lambda_k = \lambda_k^*$, as $\lambda_k \in [0, 1]$ varies continuously, while $s_k \in \{0, 1\}$ is discrete. Thus the solver will try to have as many admitted users as possible first ($\lambda_k > \lambda_k^*$). Then, after the largest set of users is admitted with guaranteed QoS, the remaining resources are distributed to either improve the QoS for already admitted users or to other users according to what requires fewer resources.

This allows us to estimate the time a non-admitted user has to wait before starting consuming the requested content:

$$W_i = T - \left\lfloor \frac{\sum_{k\in\mathcal{T}} a_{i,k}r_{i,k}}{\lambda_i^* \sum_{k\in\mathcal{T}} d_{i,k} + \theta_i^* \sum_{k\in\mathcal{T}} u_{i,k}} \right\rfloor, \qquad (9)$$

where the ratio between the total rate obtained $\sum_{k\in\mathcal{T}} a_{i,k} r_{i,k}$ and the needed rate to meet the requirements $\lambda_i^* \sum_{k\in\mathcal{T}} d_{i,k} + \theta_i^* \sum_{k\in\mathcal{T}} u_{i,k}$ approximates the number of slots where the content could be streamed at the agreed quality. After this time, a user is not immediately admitted into the system, but the solution is computed again to consider the impact of (*i*) requirement shift and (*ii*) prediction update.

In addition, since non-admitted users might start with a larger buffer state than new users, they will be required to maintain the same buffer state at the end of the optimization window (if the media is longer) or the remaining content size (if this is smaller than the starting buffer). Conserving the buffer between consecutive optimization windows is particularly useful when the content duration is longer than the optimization window and it is thus not possible to guarantee the QoS over its whole duration. Instead, the buffer conservation takes care of maintaining the quantity of resources that were lacking in the first round of optimization.

LP formulation: starting from the reduced MILP formulation and fixing the set of admitted users $\tilde{\mathcal{N}}$ for which $\tilde{s}_i = I(i \in \tilde{\mathcal{N}})$, a LP formulation is obtained from Eq. (8) setting $s_i = \tilde{s}_i$ and replacing the objective function with:

$$\underset{A,B,L,E}{\text{maximize}} \sum_{k\in\mathcal{N}} \left(K\lambda_k + \theta_k\right), \tag{10}$$

where $I(x)$ is the indicator function and is 1 if x is true and 0 otherwise. This formulation requires all users in $\tilde{\mathcal{N}}$ to satisfy the quality constraints. However, the set of admitted users is given as a parameter. The selection of such set is critical, since it may also lead to unfeasible problems.

Admission and Resource Control (ARC): Hereafter we propose a binary search to approximate the best feasible set of admitted users. To evaluate the set of admitted users we propose a greedy utility function to sort the users and then we define the set of admitted users of size $\tilde{N} = |\tilde{\mathcal{N}}|$ as the set composed of the first $\tilde{N}$ users. By means of a binary search over the size of the admitted set $\tilde{N}$, we find the largest size $\tilde{N}$ for which the problem of Eq. (10) is feasible.

The sorting function has to weight how efficiently resources are used to satisfy users' requirements. This efficiency depends on almost all the input parameters of our problem and, in particular, it is related to the sequence of achievable rates: high rates in the early slots allow a user to fill its buffer and avoid to use low rates slots, but a high rate in a slot where many users have high rates means that many users will try to use resources in the same slots.

Since evaluating all these parameters for every combination of users would be as complex as solving the original problem, we follow an indirect approach: we compute the schedule that maximizes $\sum_{k\in\mathcal{N}} (K\lambda_k + \theta_k)$ if no QoS is enforced ($\tilde{\mathcal{N}} = \emptyset$). In such a case, no user is required to meet any condition on the QoS and resources are assigned, first, to maximize the overall continuous streaming time and, then, the average quality. Thus, the solution of Eq. (10) is certainly feasible and obtains the resource allocation $\tilde{A}$.

According to the scheduling $\tilde{A}$, each user i is characterized by the two KPIs $\tilde{\lambda}_i$ and $\tilde{\theta}_i$. Consequently, the least efficient user i is the one that has the lowest $\tilde{\lambda}_i$. In case of equal $\tilde{\lambda}_i$ we choose over $\tilde{\theta}_i$. In case of both equal $\tilde{\lambda}_i$ and $\tilde{\theta}_i$, we consider the amount of used resources. Therefore, we propose the following sorting function:

$$\phi_i = \frac{T(K\tilde{\lambda}_i + \tilde{\theta}_i)}{\sum_{k\in\mathcal{T}} \tilde{a}_{i,k}}, \tag{11}$$

where $\sum_{k\in\mathcal{T}} \tilde{a}_{i,k}/T$ is the total fraction of resources used.

Once that the sorting function has been defined, we can apply a binary search over the size of the set of admitted users. We call the algorithm Admission and Resource Control and its pseudocode is given in Algorithm 1. The convergence of the binary search is ensured by the sorting of the users: in fact any given set $\tilde{\mathcal{N}}$ always includes all the elements of the smaller sets, thus, if it makes the problem unfeasible, no larger sets can be feasible.

Algorithm 1 Admission and Resource Control (ARC)

Input: R, D, U, b_M, λ_i^*, θ_i^*.
Output: $\tilde{A}$, $\tilde{\mathcal{N}}$

```
N_min = 0, N_max = N
Compute A, from Eq. (10) with Ñ = N_max
if Problem feasible then
  ã_{i,j} = a_{i,j}, Ñ = N
else
  Compute Ã, λ̃_i, θ̃_i, from Eq. (10) with Ñ = ∅
  Compute φ_i from Eq. (11) ∀i ∈ N
  Sort N in descending order of φ_i
  while (N_max − N_min) > 1 do
    Ñ = (N_max + N_min)/2
    Solve Eq. (10) with Ñ = {i ∈ N | i ≤ ⌊Ñ⌋}
    if Problem feasible then
      N_min = Ñ
    else
      N_max = Ñ
    end if
  end while
  ã_{i,j} = a_{i,j}
end if
```

In what follows we provide a few **practical considerations** about its realization in cellular networks. With reference to current LTE, Fig. 3 shows a high level diagram of an eNodeB where only the relevant functionalities are drawn. The prediction and context information functionalities are drawn outside the eNodeB as they contain network wide information that are not specific to any eNodeB. However, it is possible to cache locally in the eNodeB the information that is more frequently used. Also, while the mobility prediction may be computed outside the eNodeB, the short term achievable rate variation might be computed internally as well. The input parameters of the problem ($r_{i,j}$, $d_{i,j}$, ui, j) are obtained by combining prediction, context information and admission control functionalities. The contractual agreement function governs the constraints of the problem and defines λ_i^* and θ_i^* for all users.

The admission control function is placed in parallel to the scheduler in order for the former to provide input to the latter without changing the main scheduling logic. These two functions operate at different time granularity: while the scheduler makes decisions every few milliseconds, the admission control time slots are in the order of seconds. The admission control should be able to modulate the user weights used by the scheduler. This allows the system to enforce admission control indirectly: the weight of a user which is

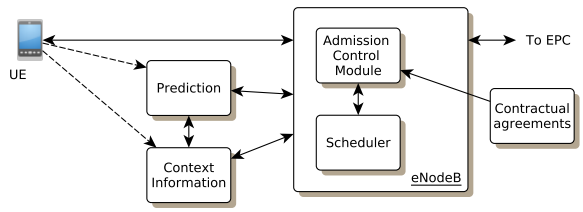


Figure 3: eNodeB high level diagram highlighting the relationship among the different modules.

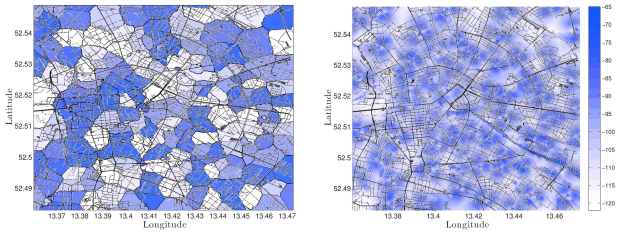


Figure 4: Coverage and pathloss maps of Berlin as measured by the MOMENTUM project [13].

not admitted in the current admission time slot is set to zero, while admitted users receive weights proportional to the fraction of resources assigned by the admission control.

In practice, whenever the admission control solution is re-evaluated, the admitted status of users that still have to complete their stream should be preserved. This can be achieved using an additional equality constraint requiring s_i to be larger or equal than the value obtained in the previous evaluation. New user arrivals can be managed either synchronously if the admission control time slots are smaller than 1 second or asynchronously if longer. In this last case, the users already admitted must preserve their condition.

5. SIMULATION RESULTS

This section presents the results of our evaluation campaign, which can be grouped in three parts: (*i*) the first part analyzes the computational complexity; (*ii*) the second evaluates how far the solution obtained by our approximation is from the original problem; (*iii*) the third part discusses the benefits of the combined admission control and resource allocation technique with respect to the baseline solution and an anticipatory technique that does not enforce QoS.

In particular we consider the following problems:

- *Original*: problem formulation of Eq. (7),
- *Simple*: mixed integer linear formulation of Eq. (8),
- *ARC*: online iterative approach of Algorithm 1,
- *RA*: anticipatory resource allocation without QoS (e.g. [8]),
- *Baseline*: plain proportionally fair scheduling.

Our evaluation campaign considers an LTE network scenario based on the pathloss data provided by the MOMENTUM project [13]. For each evaluation round we generate a random mobility trace in a 12×6 square kilometer area of Berlin (centered at latitude 52.52° North and longitude 13.42° East). Fig. 4 shows a map of the cell topology (left) in the considered area. From the mobility trace, we generate a pathloss trace computed on the pathloss map (right). Finally, we account for fast fading as in the model discussed in [21] to obtain the achievable rates and we average results over 200 repetitions of 5-minute scenarios.

The requirement traces are constant and equal for all the users to simplify the discussions of the results. However, all the formulations support any type of requirements. In particular, we set $d_{i,j} = 0.4$ Mbps and $u_{i,j} = 4.6$ Mbps to represent the different qualities available for video streams of resolution ranging from 360p ($\sim$ 400 Kbps) to 1080p ($\sim$ 5 Mbps). Unless specified otherwise, $\lambda_i^* = \lambda^* = 1$ for all users. This means that in all the following results it is required for the streaming to have no interruption. To prioritize continuous streaming time over extra quality we chase $K = 100TN$ for all the simulations.

The first tests aim to understanding which of the three formulations can be used to implement a real-time admission control and resource allocation mechanism based on system state prediction. The main challenge of such a module is to obtain a solution within the validity time of the prediction. To this end, we evaluate the three formulations over repeated instances with varying problem size, i.e., number of optimization variables involved in the specific instance.

Eq. (7) has dimensionality proportional to T^2N, while the simpler formulation of Eq. (8) has a size proportional to TN. However both include integer variable, while Algorithm 1 consists of at most $\log_2 N$ iterations of a simple LP program of size proportional to TN.

In our evaluation we explore the following parameters: users $N \in [10, 50]$, slots $T \in [10, 50]$, quality requirements $\theta_i^* = \theta^* \in [0.5, 1], \forall i$ and we compare the average computational time[3] obtained by the three formulations using GUROBI [14]. In Fig. 5(a) we fix the number of slots $T = 30$ and we plot a solid curve for ARC, dashed ($\theta^* = 1$) and dot-dashed ($\theta^* = 0.7$) curves for Simple and a dotted curve for the Original approach for $N = [10, 50]$[4].

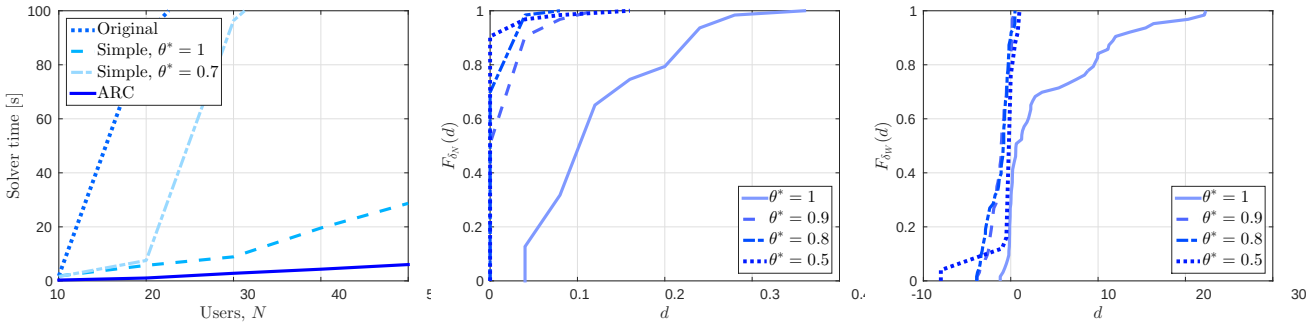
We do not plot curves for different θ^* for the original and ARC formulation as this parameter has minimal impact on the computation time. Instead, we plot two curves for the simple formulation for $\theta^* = 1$ and $\theta^* = 0.7$, because we observe that if the system does not require the full quality to be delivered, the resource allocation has more degree of freedom and decreases the solution speed.

The original formulation becomes too slow very rapidly, while the simple formulation can be computed in less than 10 seconds if $\theta^* = 1$. However, for lower θ^* the simple formulation is affordable for very small problem instances only. This is due to the fact that for small problem instances the solution becomes trivial as almost all users can be admitted. Finally, ARC obtains a solution in an affordable time for all the problem sizes.

In the second set of results we compare the solutions obtained by the simple MILP and the ARC approaches. In particular, we evaluate the number of admitted users $\hat{N}$ (MILP) and $\tilde{N}$ (ARC) and the average waiting time $\hat{W} = \sum_{k \in \mathcal{N}} \hat{w}_k (N - \hat{N})$ (MILP) and $\tilde{W} = \sum_{k \in \mathcal{N}} \tilde{w}_k / (N - \tilde{N})$ (ARC) computed with Eq. (9). We choose $N = 25$ and $T = 50$ and we vary $\theta^* \in \{1, 0.9, 0.8, 0.5\}$. Finally, for each repetition we compute $\delta_N = (\hat{N} - \tilde{N})/N$ and $\delta_W = \hat{W} - \tilde{W}$.

[3]In all cases we stop the computation after 100 seconds.

[4]We do not report the curves obtained for a fixed N varying the number of slots, because they show a similar trend.



(a) Complexity (b) eCDF of δ_N (c) eCDF of δ_W

Figure 5: Evaluation of the computational time and the optimality of the different approaches.

Fig. 5(b) and Fig. 5(c) plot the empirical cumulative distribution function (eCDF) of δ_N and δ_W respectively. Different constraints $\theta^* \in \{1, 0.9, 0.8, 0.5\}$ are plotted with solid, dashed, dash-dotted and dotted lines respectively. The former figure illustrates that the ARC approach closely approximates the number of admited users with respect to the MILP formulation for all but $\theta^* = 1$. In this case, the exact solution of the problem requires the maximum quality to be delivered in every slot to admit a user. Thus, the approximate formulation is less likely to find the exact combination of users. Similarly, Fig. 5(c) shows that for the average waiting time ARC obtains a good approximation. While in the previous figure the domain of the eCDF was limited to positive values, here δ_W can assume negative values, too: in fact, by admitting less user in the system, more resources remains for the non-scheduled users that can start the streaming earlier.

The final set of results compares Baseline (red dashed line), RA (green dash-dotted line) and ARC (solid lines from darker to lighter shade of blue representing $\theta^* \in \{1, 0.9, 0.7, 0.4\}$) to investigate the improvements offered by our proposal over existing solutions. The results for RA is obtained using the formulation of Eq. (10) with no admitted users, hence no QoS is enforced. In this set of graphs we vary both $N \in [5, 50]$ and $\theta^* \in [0.1, 1]$.

Fig. 6(a) shows the average fraction of continuous streaming obtained by the three approaches. Baseline does not leverage prediction and thus cannot avoid streaming interruption. As the number of users increases, the average interruption time reaches 15%. Both RA and ARC show almost no interruptions for any number of user. They only differ if $N > 30$ for which ARC drops a few users to enforce QoS.

Fig. 6(b) shows the average fraction of obtained quality (1 means that all the streams obtain the maximum quality in every slot) for the three approaches. The overall quality obtained decreases with the number of users for all approaches to different degrees. RA and ARC always deliver higher quality than Baseline. In addition, we plot 4 curves for different quality constraints for ARC. The two predictive approaches, ARC and RA obtain the same quality as long as the number of users is small enough to sustain the required QoS, then RA starts violating the constraint, while ARC reduce the set of admitted users.

Finally, Fig. 6(c) shows the average fraction of admitted users $\tilde{N}/N$ for ARC. The comparison between the last three figures highlights the tradeoff intrinsic to our solution: the joint admission control and resource allocation is able to tradeoff the number of admitted users and the guaranteed QoS. For instance, to obtain a stream with no interruption at 40% of the maximum quality, only 30 of the 50 requesting users can be admitted at once.

6. CONCLUSIONS

In this paper we presented an admission control and resource allocation solution for multimedia streaming in mobile networks. The proposed solution exploits system state prediction to derive the set of users that can be admitted into the system with guaranteed Quality-of-Service and specifies the resource allocation for all users. Starting from a very general MILP formulation, we reduced the approach complexity by means of a simpler LP formulation and binary search and we obtained a very fast approximation with small performance degradation. Not only does our approach improve the state-of-the-art by combining guaranteed QoS and resource allocation, but also achieves this result within a short time. These two features make our proposed solution a good candidate for the realization of online admission control modules that, in coordination with the scheduler, will be able to enforce QoS in base stations. Although these results have been obtained with perfect prediction, we intend to extend the solution to imperfect forecast.

Acknowledgments

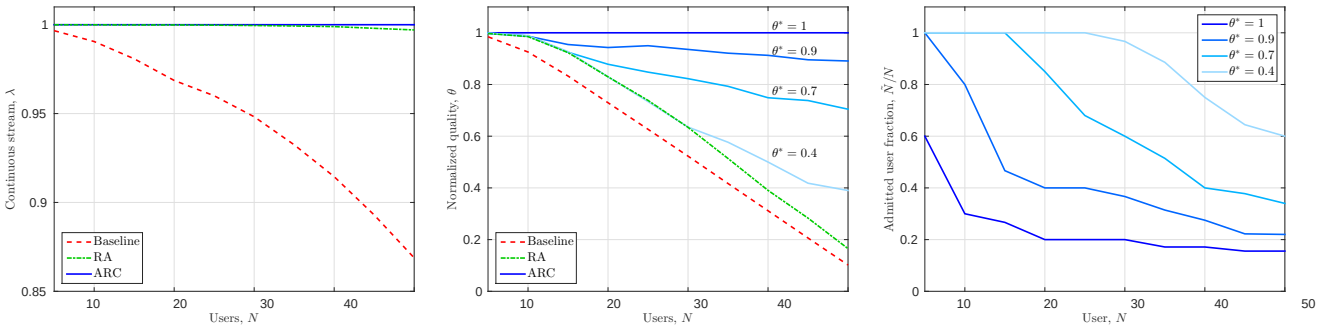
The authors thank Dr. Stefan Valentin for the fruitful discussions. The research leading to these results was partially supported by the PhD@Bell Labs Internship program, the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919), the Ramon y Cajal grant from the Spanish Ministry of Economy and Competitiveness RYC-2012-10788 and grant TEC2014-55713-R, and from the European Union H2020-ICT Gramt No. 644399 (MONROE).

7. REFERENCES

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014.

[2] H. Abou-zeid, H. Hassanein, and S. Valentin. Optimal predictive resource allocation: Exploiting mobility patterns and radio maps. In *Proc. IEEE GLOBECOM*, 2013.

[3] H. Abou-zeid, H. Hassanein, and S. Valentin. Energy-efficient adaptive video transmission:



(a) Continuous streaming (b) Extra quality (c) Admitted users

Figure 6: Evaluation of the performance of the joint admission control and resource allocation solution.

Exploiting rate predictions in wireless networks. *IEEE Transactions on Vehicular Technology*, 63(5):2013–2026, June 2014.

[4] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: predicting the evolution of popularity in user generated content. In *Proc. ACM WSDM*, 2013.

[5] A. Ashraf, F. Jokhio, T. Deneke, S. Lafond, I. Porres, and J. Lilius. Stream-based admission control and scheduling for video transcoding in cloud computing. In *Proc. IEEE/ACM CCGrid*, 2013.

[6] T. Braun, C. Castelluccia, G. Stattenberger, and I. Aad. An analysis of the diffserv approach in mobile environments. In *Proc. IQWiM-Workshop*, 1999.

[7] N. Bui, F. Michelinakis, and J. Widmer. A model for throughput prediction for mobile users. In *European Wireless*, 2014.

[8] N. Bui, S. Valentin, and J. Widmer. Anticipatory quality-resource allocation for multi-user mobile video streaming. In *Proc. IEEE CNTCV*, 2015.

[9] N. Bui and J. Widmer. Mobile network resource optimization under imperfect prediction. In *Proc. IEEE WoWMoM*, 2015.

[10] F. Dobrian, V. Sekar, A. Awan, I. Stoica, D. Joseph, A. Ganjam, J. Zhan, and H. Zhang. Understanding the impact of video quality on user engagement. *ACM SIGCOMM Computer Communication Review*, 41(4):362–373, 2011.

[11] M. Dräxler and H. Karl. Cross-layer scheduling for multi-quality video streaming in cellular wireless networks. In *Proc. IEEE IWCMC*, 2013.

[12] J. Froehlich and J. Krumm. Route prediction from trip observations. *SAE SP*, 2193:53, 2008.

[13] H.-F. Geerdes, E. Lamers, P. Lourenço, E. Meijerink, U. Türke, S. Verwijmeren, and T. Kürner. Evaluation of reference and public scenarios. Technical Report D5.3, IST-2000-28088 MOMENTUM, 2003.

[14] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2015.

[15] V. Joseph and G. de Veciana. NOVA: QoE-driven optimization of DASH-based video delivery in networks. In *Proc. IEEE INFOCOM*, 2014.

[16] P. Koutsakis, M. Vafiadis, and H. Papadakis. Prediction-based resource allocation for multimedia traffic over high-speed wireless networks. *AEU-International Journal of Electronics and Communications*, 2006.

[17] G. Majid, J. Capka, and R. Boutaba. Prediction-based admission control for DiffServ wireless internet. In *Proc. IEEE VTC-Fall*, 2003.

[18] R. Margolies, A. Sridharan, V. Aggarwal, R. Jana, N. Shankaranarayanan, V. A. Vaishampayan, and G. Zussman. Exploiting mobility in proportional fair cellular scheduling: Measurements and algorithms. In *Proc. IEEE INFOCOM*, 2014.

[19] A. K. Moorthy, L. K. Choi, A. C. Bovik, and G. De Veciana. Video quality assessment on mobile devices: Subjective, behavioral and objective studies. *IEEE J-STSP*, 6(6):652–671, 2012.

[20] A. J. Nicholson and B. D. Noble. Breadcrumbs: forecasting mobile connectivity. In *ACM MobiCom*, 2008.

[21] O. Østerbø. Scheduling and capacity estimation in LTE. In *Proc. IEEE ITC*, 2011.

[22] R. Pantos and W. May. HTTP live streaming. *IETF Draft, June*, 2010.

[23] U. Paul, A. P. Subramanian, M. M. Buddhikot, and S. R. Das. Understanding traffic dynamics in cellular data networks. In *Proc. IEEE INFOCOM*, 2011.

[24] Y. Qiao, J. Skicewicz, and P. Dinda. An empirical study of the multiscale predictability of network traffic. In *Proc. IEEE HDPC*, 2004.

[25] N. Sadek and A. Khotanzad. Multi-scale high-speed network traffic prediction using k-factor Gegenbauer ARMA model. In *IEEE ICC*, 2004.

[26] M. Z. Shafiq, L. Ji, A. X. Liu, and J. Wang. Characterizing and modeling internet traffic dynamics of cellular devices. In *Proc. ACM SIGMETRICS*, 2011.

[27] T. Taleb and A. Ksentini. QoS/QoE predictions-based admission control for femto communications. In *Proc. IEEE ICC*, 2012.

[28] S. Wang, Y. Xin, S. Chen, W. Zhang, and C. Wang. Enhancing spectral efficiency for LTE-advanced and beyond cellular networks [Guest Editorial]. *IEEE Wireless Communications*, 21(2):8–9, April 2014.