# FALCON: Joint Fair Airtime Allocation and Rate Control for DASH Video Streaming in Software Defined Wireless Networks

Miguel Catalan-Cid, Daniel Camps-Mur, Mario Montagud, August Betzler
i2CAT Foundation & Universitat de València

## ABSTRACT

Software Defined Wireless Networks offer an opportunity to enhance the performance of specific services by applying centralized mechanisms which make use of a global view of the network resources. This paper presents FALCON, a novel solution that jointly optimizes fair airtime allocation and rate recommendations for Server and Network Assisted DASH video streaming, providing proportional fairness among the wireless DASH clients. Since this problem is NP-hard and its resolution in dense scenarios is too computationally expensive to be applied in practice, FALCON introduces a novel heuristic algorithm that is proved to achieve almost optimal results in a significantly lower time. The performance of FALCON is also evaluated when used in conjunction with three referent Adaptive Bit Rate strategies (PANDA, BOLA and RobustMPC) in a simulated realistic ultra-dense scenario based on an In-flight Entertainment System with up to 175 clients. The obtained results show that FALCON provides significant benefits by minimizing instability and buffer underruns, while obtaining a fair video rate and airtime allocation among clients, thus contributing to an enhanced Quality of Experience.

## CCS CONCEPTS

• **Information systems → Multimedia streaming**; • **Networks → Wireless access networks**; **Programmable networks**
.

## KEYWORDS

DASH, SAND, SDWN, QoE, airtime, fairness, rate control

**ACM Reference Format:**
Anonymous Author(s). . FALCON: Joint Fair Airtime Allocation and Rate Control for DASH Video Streaming in Software Defined Wireless Networks . In *Proceedings of* . ACM, New York, NY, USA, 7 pages.

## 1 INTRODUCTION

Recent advances in wireless communications enable access densification by increasing the capacity and availability of the networks. For instance, 5G technologies are expected to provide a wireless access alternative to extremely dense scenarios, such as the In-Flight Entertainment & Connectivity (IFEC) systems. In these scenarios, Software Defined Wireless Networks (SDWNs) can contribute to guarantee fairness among clients and a satisfactory Quality of Experience (QoE), by enabling innovative network control mechanisms [9]. In particular, centralized association control solutions can make use of a global view of the wireless access network to enhance its efficiency, by intelligently allocating clients across the available Access Points (APs) and wireless channels. Although significant efforts have been devoted to devising client steering strategies to improve network wide performance, none of the proposed schemes consider DASH video streaming demands [28].

Video streaming is one of the most popular and bandwidth-demanding services in current networks, accounting for more than half of global traffic, and this is even expected to increase in the upcoming years [6]. In this context, Dynamic Adaptive Streaming over HTTP (DASH) [25] has become the dominant technology for online video content delivery. In DASH, the media content is encoded into multiple representations (e.g. with different resolutions and/or bitrates), and each representation is split into short segments (typically with a duration of 2-10s). Then, clients can dynamically request the segments of the most appropriate representation, based on the current network and end-system conditions. The algorithms that decide which representation to download for each segment are typically implemented at the client side, and are commonly known as Adaptive Bit Rate (ABR) strategies. Up to date, many ABR strategies based on different criteria have been devised to meet the targeted requirements and maximize the QoE in specific scenarios [4].

Even though the capacity of networks is continuously increasing and ABR strategies have the potential of adapting the video quality based on the current network conditions, meeting the streaming demands from massive clients in dense Radio Access Networks (RAN) is still very challenging. This can be for example the case of IFEC systems in airplanes. In these scenarios, performance issues arise, like network resources underutilization, QoE unfairness among clients, quality instability due to frequent bitrate switching, and video stalls [2, 11, 14, 15]. These issues have a significant impact on the clients' perceived QoE, leading to disengagement and abandonment [10, 24], and thus negatively affecting content and service providers.

As an alternative to client-driven quality adaptation strategies, network elements (e.g. proxies, gateways, SDN controllers. . . ) can be used to assist the clients in selecting an appropriate representation. Network elements can obtain more easily an overall and accurate view of the network conditions, like the available bandwidth and its capabilities. This is the approach followed in Server and Network Assisted DASH (SAND), an extension to the DASH standard to introduce DASH Assisting Network Elements (DANEs) with the above features, by introducing new messages and mechanisms [27], which has been standardized in ISO 23009-5 [13]. The SAND proposal is especially interesting in 5G architectures, where the DANE can be placed near the RAN. One of the most adopted and evaluated mechanism of SAND is the recommended rate assistance to DASH clients, which can significantly overcome the performance issues in dense and massive environments (e.g. [11, 15, 17]).

In this paper, we present FALCON, a novel association control mechanism for SDWNs, which jointly considers the wireless conditions experienced by each DASH client and their traffic demands, in order to allocate clients across APs and to recommend a video rate

, ,
©

Pre-print version: Final publication available in https://dl.acm.org/doi/10.1145/3386290.3396931

for each client. To the best of our knowledge, this is the first solution joining rate assistance and association control in the wireless access. FALCON aims to provide proportional fairness among wireless clients, which has been proven to be an NP-hard problem [19]. Hence, we present an heuristic algorithm that obtains almost optimal results, while decreasing the running time up to five orders of magnitude compared to an optimal solver. Finally, we demonstrate the benefits of FALCON applied to DASH video streaming in a realistic IFEC scenario, which is a paradigmatic example of ultra-dense wireless access. The obtained results show that FALCON provides significant improvements in terms of key QoE related metrics (e.g. fairness, instability and playout interruption) when using along three represantive ABR strategies of different nature: rate-based (PANDA), buffer-based (BOLA), and hybrid (RobustMPC).

The rest of the paper is organized as follows. Section 2 briefly reviews the state-of-the-art. Section 3 presents FALCON, describing the network model, the definition of the problem as a MIQCQP and the novel heuristic algorithm. Section 4 presents the evaluation framework and the obtained results. Finally, Section 5 provides the conclusions and discusses ideas for future work.

## 2 RELATED WORK

The study in [4] surveys existing client-driven ABR strategies, which can be mainly categorized as buffer-based (i.e. the buffer occupancy level is the key factor that determines the representation to select), rate-based (i.e. the throughput / bandwidth estimation is the key factor), power-based (i.e. the power consumption is the key factor) and hybrid approaches (i.e. strategies that use a combination of the above factors and metrics). The defined ABR strategies highly depend on the targeted requirements and scenarios, but mainly aim at enhancing the DASH streaming adaptability and continuity, thus maximizing the QoE.

From the huge variety of existing ABR strategies, three representative ones have been considered in this work, each of them belonging to a different category. PANDA is a rate-based ABR strategy that implements a probe-and-adapt mechanism which works similar to TCP's congestion control, applying an additive-increase/multiplicative-decrease (AIMD) strategy to the video rate according to estimations of the available throughput [20]. BOLA implements Lyapunov optimization techniques to minimize rebuffering and maximize video quality, adapting the video quality and the segment downloading rate based on the the buffer level[26]. Finally, RobustMPC uses a Model Predictive Control hybrid approach which combines throughput prediction for a determined number of future chunks and buffer occupancy information, by selecting the video rate which maximizes a set of QoE metrics (average quality, quality variations and rebuffering) [29].

Despite the significant progress in the design and adoption of best-fitted ABR strategies, previous studies have reported performance issues with DASH in shared networks in which the DASH clients compete with other type of traffic or with other DASH clients sharing the same network resources. In [2], it is shown how off-the-shelf DASH players react to changing network conditions and how the video quality becomes unstable when two DASH players compete for bandwidth. In [3], it is reflected that the bandwidth estimation by the individual DASH clients is a challenging task, identifying the on-off download traffic patterns as a cause of instability. It is also demonstrated in that work that the number of active clients has a significant impact in the number of quality switches. In [12], it is confirmed that the joint TCP and DASH adaptation algorithms to dynamically react to changing network conditions can originate video quality instability.

To overcome the detected instability, scalability and fairness challenges in the mentioned scenarios, many network-assisted solutions have been devised up to date. In [16], a proxy-based implementation of a DANE is proposed. The HTTP traffic is routed through this proxy server, which can be placed between the clients and the server, and inspects the traffic to detect DASH streams and their characteristics. Rate adaptation assistance is provided by altering the content of the HTTP request headers. An evolved version of that proxy-based network-assisted solution in [16] is presented in [15]. It relies on the SDN paradigm, by proposing a DANE that works out-of-band via a WebSocket connection from the DASH players to the DANE. Evaluation results in a Wi-Fi network show that this DANE-based proposal contributes to the delivery of a stable and high quality DASH streams. The work in [7, 8] investigates network-assisted streaming approaches that rely on active cooperation between video streaming applications and the network in SDN environments to contribute to a better quality fairness, by also solving a max-min fairness optimization problem at runtime. In [22], it is demonstrated that SAND provides an improved performance in terms of fairness and QoE, compared to the sole use of ABR strategies, when multiple DASH players compete for shared network resources.

Related work analysis shows that rate adaptation assistance contributes to a better quality, although just few studies considered wireless scenarios when evaluating the presented proposals. FALCON goes one step beyond, since it is pioneering in jointly considering a SAND-based solution to recommend video rates to the clients according to the available wireless resources and conditions of dense scenarios, while properly allocating clients across the APs using centralized association control.

## 3 FALCON DESIGN

The objective of FALCON is to obtain a fair allocation of resources among DASH clients, according to the properties of their active video streams and the characteristics of the wireless channel. This objective can be stated as the problem of providing proportional fairness in a multi-rate multi-AP network, which has been proven to be NP-hard [19], even without considering the specific traffic demands of each DASH client.

Let us consider a network where $N$ is the set of clients, $M$ is the set of APs in orthogonal channels, and $K_i$ is the number of possible Representation Indexes (RI) in the Media Presentation Description (MPD) of each client. Then, $x_{i,j}$ is defined as a binary variable that represents the selection of the AP $j$ where a client $i$ is attached, and $y_{i,k}$ as a binary variable that represents the selection of the RI $k$ in the MPD for a given client $i$, corresponding to a video rate of $r_{i,k}$. Consequently, the problem of achieving proportional fairness according to DASH video streaming traffic demands can be formulated using the following cost function:

$$\underset{\vec{x_{i,j}},\vec{y_{i,k}}}{\operatorname{argmax}} \sum_{j\in M}\sum_{i\in N} x_{i,j} \sum_{k\in K_i} y_{i,k} \log r_{i,k} \quad (1)$$

Subject to the following constraints:

$$x_{i,j} \in \{0,1\} \quad (2)$$

$$y_{i,k} \in \{0,1\} \quad (3)$$

$$\forall i \in N, \quad \sum_{j\in M} x_{i,j} = 1 \quad (4)$$

$$\forall i \in N, \quad \sum_{k\in K_i} y_{i,k} = 1 \quad (5)$$

$$\forall j \in M, \quad \sum_{i\in N} x_{i,j} \sum_{k\in K_i} y_{i,k} \frac{r_{i,k}}{b_{i,j}} = \sum_{i\in N} x_{i,j} \sum_{k\in K_i} y_{i,k} AT_{ij,k} =\leq C_j \quad (6)$$

, where constraint (4) ensures that each client is associated to only one AP, and constraint (5) ensures each client selects only one RI. Finally, constraint (6) ensures that the airtime consumed by all the clients assigned to an AP does not exceed the available airtime $C_j$. The airtime of a client is defined as its current video rate $r_{i,k}$ divided by its available bandwidth $b_{i,j}$, i.e. its amount of channel utilization. $C_j$ can also consider the unavailable airtime in this channel due to external interference, background traffic or reserved resources in a multi-tenant architecture [23].

Formally, given that both the cost function (1) and constraint (6) are quadratic, this problem can be defined as a Mixed Integer Quadratically Constrained Quadratic Program (MIQCQP). However, as will be shown in Section 4, state-of-the-art MIQCQP solvers cannot cope with dense wireless networks, where hundreds of clients, tens of APs, and tens of qualities per video are possible. Therefore, heuristic solutions are needed to deliver close to optimal solutions with a limited computational cost.

**Algorithm 1:** FALCON algorithm

```
Inputs: list of $C_j$, list of $b_{ij}$, list of available $r_{i,k}$ per user,
        list H of current assignments of users $i$ to channels $j$
 1: //Populate $F$ with $k = 1\ \forall ij$:
    for each $b_{ij} \neq 0$ do $F(n) = \{AT_{ij,1}, r_{i,1}\}$ end for
 2: //Update $C_j$ according to $H$ and $k=1\ \forall j$:
    for each $i$ do
      if $(H(i) == j)$ then $C_j = C_j - AT_{ij,1}$ endif
    end for
 3: Sort $F$ in ascending order
 4: for each $F(n)$ in $F$ do
 5:   if $(F(n)\ != \{AT_{ij,Ki}, r_{i,Ki}\})$ then   //last RI $K_i$ already assigned to $i$
 6:     if $(H(i) == j)$ then                  //$i$ already in channel $j$
 7:       if $(C_j + AT_{ij,k} - AT_{ij,k+1} >= 0)$ then
 8:         $C_j = C_j + AT_{ij,k} - AT_{ij,k+1}$
 9:         $F(n) = \{AT_{ij,k+1}, r_{i,k+1}\}\ \forall j$
10:         goto 3
11:       end if
12:     else if $(H(i) == l)$ then             //$i$ in other channel $l$
13:       if $(C_j - AT_{ij,k+1} >= 0)$ then
14:         $C_j = C_j - AT_{ij,k+1}$, $C_l = C_l + AT_{il,k}$
15:         $F(n) = \{AT_{ij,k+1}, r_{i,k+1}\}\ \forall j$, $H(i) = j$
16:         goto 3
17:       end if
18:     end if
19:   end if
20: end for
Outputs: final list H of assignments, final list of $r_{i,k}$ according to $F$
```

FALCON proposes a heuristic based on the fact that, in the case of a single multi-rate AP, max-min airtime fairness provides proportional fairness [5]. Our solution is based on a progressive filling algorithm, which gradually increases the video rate of the DASH clients according to an ordering based on the airtime consumption of each client in the available APs. The FALCON heuristic is presented in Algorithm 1 and described next.

- Initially, a list $F$ is constructed, where each element in the list is a tuple $F(n) = \{AT_{ij,k}, r_{i,k}\}$. The first element of $F(n)$ represents the airtime of client $i$ when associated with the AP in channel $j$, and using RI $k$. The second element represents the video rate when using RI $k$. Thus, list $F$ can have a maximum size of $N * M$ in case that all the $N$ clients can be associated to all the $M$ APs. In addition, a list $H$ is defined to store the initial client-AP associations.
- In line 1, the elements in $F$ are initialized, assuming the lowest RI for each client.
- In line 2, according to the initial client-AP associations provided in list $H$, the available airtime of each channel is updated with these initial airtimes.
- In line 3, $F$ is sorted in ascending order, using the airtime component $AT_{ij,k}$ of the $F(n)$ entries as the primary sorting key, and the video rate component $r_{i,k}$ as the secondary one. Thus, in case of having multiple entries with the same airtime, the client with the lower video rate will be prioritized.
- In line 4, the algorithm selects the first entry in the list, and, in case that an RI increase is possible (i.e. channel $j$ has enough airtime to serve to client $i$ a video stream with an RI of $k+1$), all the $F(n)$ entries associated to this client are updated according to the value of $r_{i,k+1}$. Consequently, the available airtime of each channel and the list of client-AP associations are updated according to the selected channel for the RI increase of this client, and the algorithm restarts again from line 3. On the contrary, if an RI increase for this entry was not possible, the algorithm continues with the next one in the ordered list.

The algorithm finalizes after all clients have been able to allocate their maximum RI $K_i$ or when any additional increase of the RI of a client leads to a saturated channel (i.e. $C_j < 0, \forall j$), obtaining a final list of client-AP associations and a list of recommended video rates for each client. By following this algorithm, resources are allocated gradually to the different clients and APs in an efficient way: prioritizing lower airtimes gives indeed preference to clients with low RIs, leading to load fairness, but also to clients using higher link rates, leading to a higher aggregate throughput in the network.

With regard to the computational complexity, the sorting algorithm (line 3) can be called a maximum of $\sum_{i=1}^{N} K_i$ times and has a maximum complexity of $O(N * M * ln(N * M))$. However, since usually the network becomes overloaded before reaching the last RI for each client, the number of iterations will be considerably lower. Also, in large scenarios, not all clients will reach all the APs, leading to a size smaller than $N * M$ for $F$. In Section 4 we present an evaluation of the execution times experienced by FALCON.

Figure 1 depicts the components and interactions involved in FALCON. The main component is the FALCON Controller (FC), which can be part of the Radio Resource Management (RRM) subsystem of a RAN Controller in a SDWN architecture. The FC considers as input RAN and video service telemetry to run the FALCON algorithm. Then, it generates as output a list of client-AP assignments,

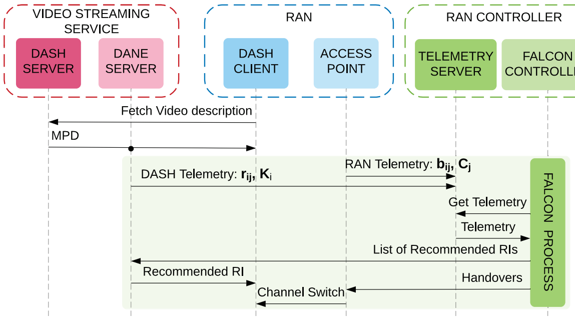

**Figure 1: FALCON's Network model and main procedures.**

which are checked by the FC to perform the necessary handovers, and a list of recommended RIs, which are sent to the DANE.

It is important to highlight that SAND does not enforce the utilization of the advised rate by the clients, since the algorithm of each particular DASH player can have local information (e.g. buffer status) recommending a different rate. Thus, in our model, after receiving the recommended RI or the equivalent representation rate from the DANE, each DASH client continues to rely on its built-in DASH rate adaptation algorithm, but using the RI recommended by FALCON as the maximum available for the current video stream.

Finally, note that FALCON requires network triggered handovers, which is a feature not available in standard Wi-Fi enterprise systems. For this purpose FALCON adopts a network model based on the BIGAP architecture [30], where all APs share the same Basic Service Set Identifier (BSSID) and their radio interfaces are assigned to $M$ orthogonal channels. Thus, association control relies on Dynamic Frequency Selection (DFS) signaling to provide seamless channel switching (i.e. handovers). However, FALCON could also be implemented in other SDWN architectures as, for instance, solutions based on creating one single virtual AP per station [9].

# 4 PERFORMANCE EVALUATION

In this section we evaluate the performance of FALCON by means of simulations. First, a standard MIQCQP solver and the FALCON algorithm are compared in terms of optimality and execution time. Then, the performance of the FALCON mechanism applied to DASH video streaming in an IFEC scenario is evaluated according to relevant metrics which have an impact on the QoE of the clients.

## 4.1 Comparison of the FALCON algorithm with the MIQCQP solver

In this section we evaluate the ability of the MIQCQP solver and of the FALCON algorithm to obtain an optimal solution that maximizes the cost function introduced in Equation (1); i.e. the assignment of video rates ($y_{ik}$) and APs ($x_{ij}$) to the DASH clients in order to optimize the proportional fairness of the system. The MIQCQP solver is implemented using OPTI[2], which is an optimization toolbox for Matlab, and SCIP [3][1], which is one of the fastest non-commercial solvers for mixed integer programming. In addition, the FALCON algorithm is implemented using Python.

For both cases, neither the wireless channel nor the DASH service are simulated, since the evaluation is focused on the performance of the algorithms given a set of inputs. In particular, we consider 802.11a link rates (i.e. from 6 to 54 Mbps, that are assigned randomly to each $b_{ij}$), and the same video stream with 10 quality levels (from to 50 kbps to 5 Mbps) for all the clients . We analyze the performance of both algorithms when varying the number of clients (up to 150) and APs (up to 5, each of them with different available airtimes $C_j$ between 1 and 0.5). Presented values are the average of 10 simulations, which were run in a laptop with the following characteristics: Intel Core i7-8550U 1.8 GHz and 16 GB RAM.

Figure 2a depicts the average utility per user achieved by the MIQCQP solver and the FALCON algorithm (i.e. $log(r_{i,k})$), showing that FALCON reaches a very close performance to the optimal solution (traces are almost overlapped). Indeed, Figure 2b depicts the obtained normalized Root Mean Square Error (RMSE) of the utility difference between FALCON algorithm and the MIQCQP solver, which is below the 0.025% in all the cases. Note that a decreasing utility per user implies that the solutions provided by the solvers are non-trivial, i.e. due to saturation all clients cannot be served with their maximum rates (i.e. 5 Mbps). In the scenario consisting of 5 APs, due to the complexity of the quadratic constraint, in the cases with a higher number of clients (50, 75, 100 and 150) the MIQCQP solver was not able to obtain an optimal solution in a reasonable time without exhausting the available memory resources of Matlab. Therefore, for these cases we cannot establish how far from the optimal value the solutions reported by MIQCQP or FALCON are (RMSE values for these cases are not depicted). Nevertheless, given the fact that other results lead to similar RMSE values regardless the number of clients or APs, we can expect similar performance for these cases as well.

Figure 2c depicts the running time of the different algorithms in logarithmic scale. In all cases, the runtimes of the FALCON algorithm are several orders of magnitude lower than the ones obtained using the MIQCQP solver. In particular, increasing the complexity of the quadratic constraint (i.e. 3 and 5 APs) and the traffic demand of the network (i.e. the number of clients), leads to a very significant increase of the MIQCQP solver runtime, making impractical its utilization in real scenarios. Indeed, as aforementioned, the MIQCQP solver was not able to find the optimal solution in the cases with more than 30 users and 5 APs, being outperformed by FALCON with a runtime five orders of magnitude smaller. We also computed the 95% confidence interval of the runtime results. On the one hand, FALCON obtained an interval below the 15% of the average values, which are below 1 second for all cases with less than 150 clients. On the other hand, MIQCQP confidence intervals exceeded in some cases the 30% of the average, leading to very variable and high runtimes depending on the characteristics of the network (i.e. the inputs defining the quadratic constraint). In addition, in order to stress the FALCON algorithm, we also evaluated its runtime with a higher number of APs (10 and 20) and clients (300 and 500). Results show that, even in these very large problem instances, the measured FALCON runtime is below 10 seconds, proving the ability of the FALCON algorithm to achieve a close-to-optimal-proportional fairness in dense scenarios in a practical amount of time.

[2] https://www.inverseproblem.co.nz/OPTI/index.php/Probs/MIQCQP
[3] https://scip.zib.de/

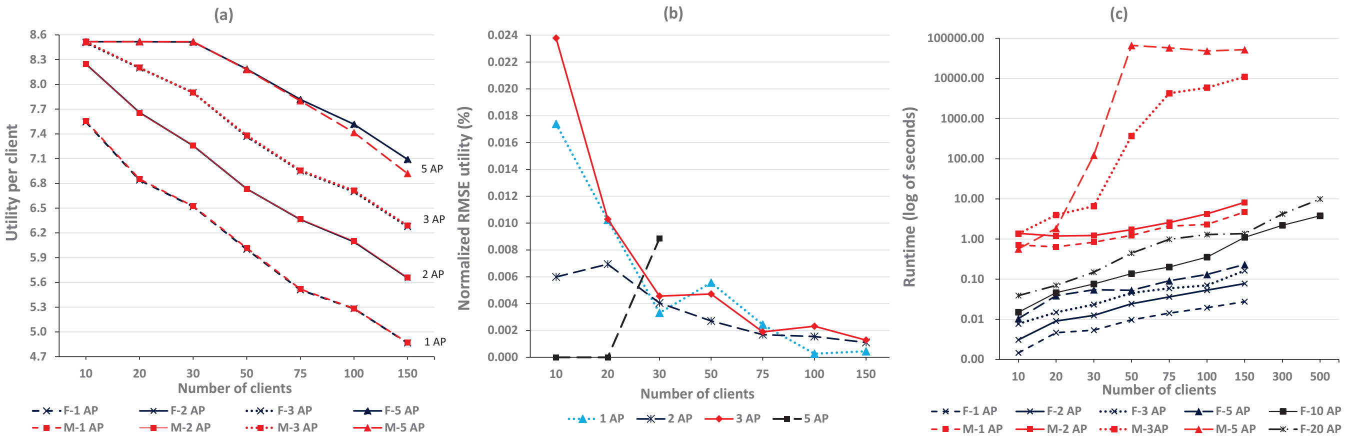

**Figure 2: FALCON's algorithm evaluation. The F- character indicates the utilization of FALCON while the M- character indicates the results of the MIQCQP solver. (a) Average utility per client, (b) Normalized RMSE of the utility difference between FALCON algorithm and the MIQCQP solver, and (c) Runtime of the algorithms**

## 4.2 Performance of FALCON in a realistic scenario

In this section, we evaluate the performance of FALCON by means of simulations using ns-3[4] in a realistic airplane IFEC scenario with dimensions of 36*4*2.5 meters. We consider up to 175 clients located according to an airplane seat distribution and three IEEE 802.11a APs with two radio interfaces placed uniformly along the airplane (i.e. at 9, 18 and 27 meters). Thus, six orthogonal channels are available and clients are statically assigned to the channel with the highest RSSI. Finally, the APs in the airplane are connected to the DASH server through a wired LAN of 1 Gbps.

The ns-3 DASH model is based on the work in [21][5] [6], where we extended the built-in PANDA, BOLA and RobustMPC ABR strategies to be able of using FALCON's recommended RIs as the maximum available RI for each particular DASH video stream. To ease the interpretation of the results, all DASH clients consume the same video stream and have the same characteristics (seat screens). The video stream is based on the "Big Buck Bunny" movie, using its real duration, segment sizes and qualities[18][7]. The movie consists of 299 segments of 2 seconds and 20 quality levels, but in the simulations we limited the available set of qualities to 9 (from 100 kbps to 8 Mbps) to reduce the computational cost of RobustMPC. For fairness, all players attempt to maintain a steady-state buffer of 28 seconds.

When using FALCON, the algorithm is run before starting the streaming, modifying the maximum RI of each client and triggering the necessary handovers according to its output. Table 1 summarizes the different parameters of the scenario. In each case, results are the average of 10 simulations.

Figure 3 depicts the results of this experiment according to the key performance metrics that influence QoE in video streaming [4]: media representation, fairness, quality instability and buffer underruns (i.e video stalls). The quality instability is computed as the number of RI levels dropped during the whole video reproduction divided by the total number of segments [11]. Results show that, compared to the original versions of the different ABR strategies, FALCON provides significantly better performance, maintaining or even increasing the average RI without compromising fairness between clients, and avoiding instability and buffer underruns, which may result in a significantly enhanced QoE.

**Table 1: Main simulation parameters**

| | |
|---|---|
| $N$ | 35, 70, 105, 140, 175 |
| $M$ | 6 |
| $K$ | 9 (0.1, 0.25, 0.5, 0.9, 1.2, 2, 3, 5 and 8 Mbps) |
| $C_j$ | $0.9\ \forall j$ |
| Target Buffer | 28 seconds |
| PANDA[20] | $\kappa = 0.14, \omega = 0.3, \alpha = 0.2, \beta = 0.2, \epsilon = 0.15$ |
| BOLA[26] | $\gamma p = 0.597133, V = 3.349338$ |
| RobustMPC[29] | FastMPC with h=5 |

PANDA recovers slower from the quality drops caused by congestion due to too high RI selections and, in some cases, to a suboptimal client distribution among APs (i.e. the default association based on RSSIs). Therefore, although obtaining better stability (Figure 3b) and larger buffers than the other ABR strategies (Figure 3d), it leads to lower average RIs (Figure 3a). Using FALCON's handover and RI recommendation mechanisms, quality drops are almost avoided, increasing the average quality across all DASH clients. Also, with an increasing number of clients, FALCON is able to totally avoid buffer underruns (Figures 3e and f) caused by the high congestion present in the wireless channels.

Although following different ABR strategies, results in Figure 3a show that BOLA and RobustMPC obtain similar performance in terms of average RI. Compared to PANDA, they are able to maintain higher RIs by considering the buffer levels. However, this leads

[4] https://www.nsnam.org/
[5] https://github.com/haraldott/dash
[6] https://github.com/tomlyko/ns3-dash-cmaf-model
[7] http://www-itec.uni-klu.ac.at/ftp/datasets/mmsys12/BigBuckBunny/

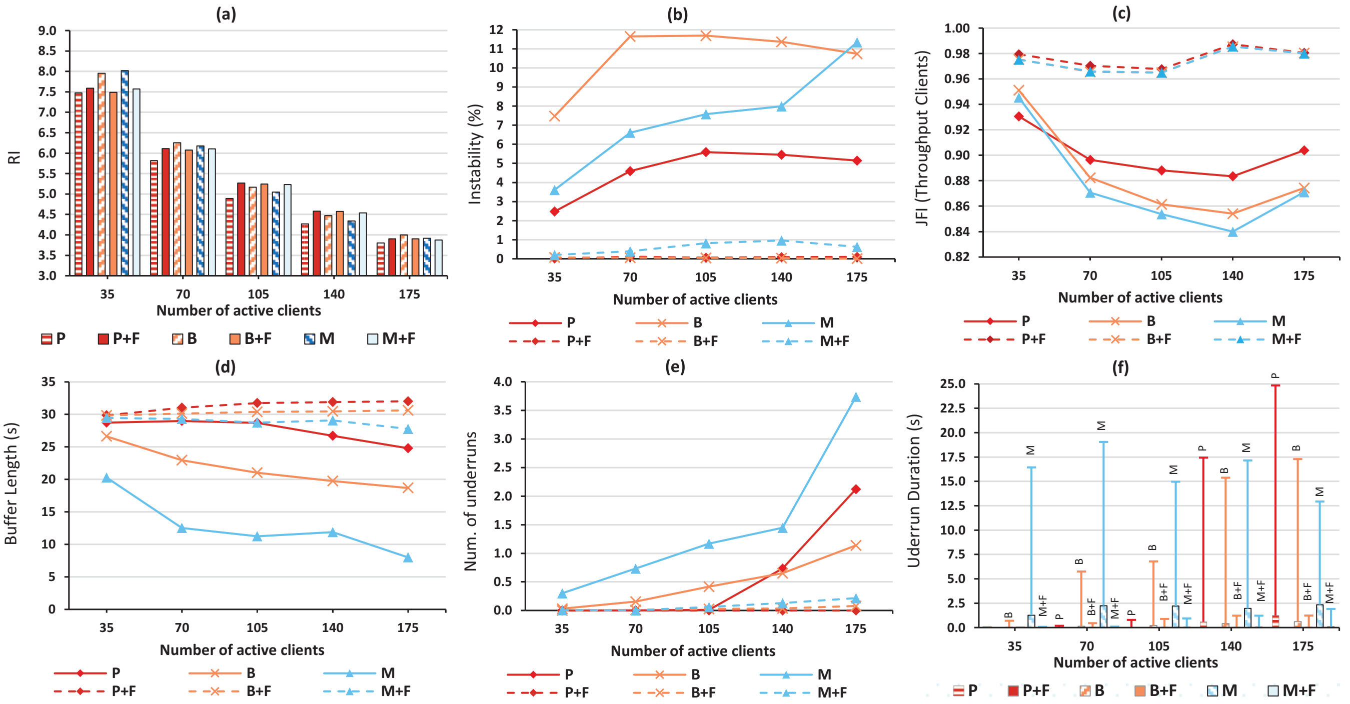

**Figure 3: IFEC scenario results using PANDA (P), BOLA (B) and RobustMPC (M). The $+F$ character indicates the utilization of FALCON. Values are the average per client. (a) RI of the clients. (b) Instability. (c) Jain Fair Index (JFI) of the throughput of the clients (B+F and M+F results are overlapped). (d) Buffer length. (e) Number of underruns. (f) Duration of the underruns, where error bars denote the maximum underrun suffered by one of the clients (accumulated during the whole reproduction)**

to higher instability (Figure 3b) and buffer underruns (in number, average duration and maximum duration) (Figures 3e and f), since some clients become significantly congested by the high downloading rates of other ones. In general, BOLA is better suited to avoid buffer underruns by decreasing RIs when needed (i.e. increasing the instability), while RobustMPC maintains RIs more stable but leads to lower buffers levels and numerous underruns. Nevertheless, in both cases the use of FALCON dramatically decreases instability and underrun levels while obtaining similar and, in some cases, better average video qualities.

It is important to remark the slight variations in performance between the different ABRs when using FALCON, since in such cases results are almost independent from the original algorithm. Indeed, in all the cases the obtained average RIs are very close to the FALCON's recommended ones (we obtain a normalized RMSE lower than the 1%), leading to the noticeable stable RIs and buffers levels shown in Figure 3b and Figure 3d, respectively. Also, note that in such a dense scenario, where a proper wireless coverage is assured by the 6 available channels, all clients should achieve a similar performance. As shown in Figure 3c, FALCON is able to obtain a fairer distribution of the throughput between the different clients by efficiently associating the clients among the available APs and recommending appropriate RIs according to the airtime consumption. Therefore, when using FALCON in this scenario all the clients achieve similar RIs in contrast to the original ABR strategies that lead to a worse fairness index.

## 5 CONCLUSIONS AND FUTURE WORK

In this paper we have introduced FALCON, a novel centralized RRM solution that jointly allocates wireless stations to APs and video rates to DASH clients in order to maximize their QoE. FALCON optimizes proportional fairness in a wireless network according to DASH traffic demands. We have first presented an MIQCQP solver that obtains optimal proportional fairness, but leads to very high runtimes in dense scenarios. Accordingly, we have proposed the FALCON heuristic algorithm, which obtains close to optimal results in all cases, while decreasing the runtime up to five orders of magnitude. Finally, we have evaluated FALCON in a realistic ultra-dense IFEC scenario. Results show that FALCON is able to avoid the main factors that impact the QoE when using three referent state-of-the art ABR strategies (one of a different category), including instability, buffer underrunning and fairness, without degrading the average video quality.

Our future work is to evaluate FALCON in a real testbed, combining and comparing it with other ABR strategies and network-assisted solutions. In addition, we plan to analyze the impact of network dynamics on FALCON, including the co-existence with traffic from other services and interference from external networks, the presence of simultaneous DASH video streams with different characteristics, and the effect of mobile and incoming/leaving clients.

**ACKNOWLEDGMENTS**

This work has been supported by the Spanish Ministry under grant TEC2016-76795-C6-2-R, and by the EC under grant N 857201 (5G VICTORI).

## REFERENCES

[1] Tobias Achterberg. 2009. SCIP: solving constraint integer programs. *Mathematical Programming Computation* 1, 1 (01 Jul 2009), 1–41. https://doi.org/10.1007/s12532-008-0001-1

[2] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C. Begen, and Constantine Dovrolis. 2012. What Happens When HTTP Adaptive Streaming Players Compete for Bandwidth? *(NOSSDAV '12)*. Association for Computing Machinery, New York, NY, USA, 9–14. https://doi.org/10.1145/2229087.2229092

[3] Saamer Akhshabi, Ali C. Begen, and Constantine Dovrolis. 2011. An Experimental Evaluation of Rate-Adaptation Algorithms in Adaptive Streaming over HTTP *(MMSys '11)*. Association for Computing Machinery, New York, NY, USA, 157–168. https://doi.org/10.1145/1943552.1943574

[4] A. Bentaleb, B. Taani, A. C. Begen, C. Timmerer, and R. Zimmermann. 2019. A Survey on Bitrate Adaptation Schemes for Streaming Media Over HTTP. *IEEE Communications Surveys Tutorials* 21, 1 (Firstquarter 2019), 562–585. https://doi.org/10.1109/COMST.2018.2862938

[5] A. Checco and D. J. Leith. 2011. Proportional Fairness in 802.11 Wireless LANs. *IEEE Communications Letters* 15, 8 (August 2011), 807–809. https://doi.org/10.1109/LCOMM.2011.060811.110502

[6] VNI Cisco. 2018. Cisco visual networking index: Forecast and trends, 2017–2022. (2018).

[7] Giuseppe Cofano, Luca De Cicco, Thomas Zinner, Anh Nguyen-Ngoc, Phuoc Tran-Gia, and Saverio Mascolo. 2017. Design and Performance Evaluation of Network-Assisted Control Strategies for HTTP Adaptive Streaming. *ACM Trans. Multimedia Comput. Commun. Appl.* 13, 3s, Article Article 42 (June 2017), 24 pages. https://doi.org/10.1145/3092836

[8] G. Cofano, L. De Cicco, T. Zinner, A. Nguyen-Ngoc, P. Tran-Gia, and S. Mascolo. 2016. Design and Experimental Evaluation of Network-Assisted Strategies for HTTP Adaptive Streaming *(MMSys '16)*. Association for Computing Machinery, New York, NY, USA, Article Article 3, 12 pages. https://doi.org/10.1145/2910017.2910597

[9] B. Dezfouli, V. Esmaeelzadeh, J. Sheth, and M. Radi. 2019. A Review of Software-Defined WLANs: Architectures and Central Control Mechanisms. *IEEE Communications Surveys Tutorials* 21, 1 (Firstquarter 2019), 431–463. https://doi.org/10.1109/COMST.2018.2868692

[10] Florin Dobrian, Vyas Sekar, Asad Awan, Ion Stoica, Dilip Joseph, Aditya Ganjam, Jibin Zhan, and Hui Zhang. 2011. Understanding the Impact of Video Quality on User Engagement *(SIGCOMM '11)*. Association for Computing Machinery, New York, NY, USA, 362–373. https://doi.org/10.1145/2018436.2018478

[11] O. El Marai, T. Taleb, M. Menacer, and M. Koudil. 2018. On Improving Video Streaming Efficiency, Fairness, Stability, and Convergence Time Through Client–Server Cooperation. *IEEE Transactions on Broadcasting* 64, 1 (March 2018), 11–25. https://doi.org/10.1109/TBC.2017.2781146

[12] Jairo Esteban, Steven A. Benno, Andre Beck, Yang Guo, Volker Hilt, and Ivica Rimac. 2012. Interactions between HTTP Adaptive Streaming and TCP *(NOSSDAV '12)*. Association for Computing Machinery, New York, NY, USA, 21–26. https://doi.org/10.1145/2229087.2229094

[13] International Organization for Standardization. 2018. ISO/IEC 23009-5:2017. Information technology – Dynamic adaptive streaming over HTTP (DASH) – Part 5: Server and network assisted DASH (SAND). Standard. (2018).

[14] Te-Yuan Huang, Nikhil Handigol, Brandon Heller, Nick McKeown, and Ramesh Johari. 2012. Confused, Timid, and Unstable: Picking a Video Streaming Rate is Hard *(IMC '12)*. Association for Computing Machinery, New York, NY, USA, 225–238. https://doi.org/10.1145/2398776.2398800

[15] Jan Willem Kleinrouweler, Sergio Cabrero, and Pablo Cesar. 2016. Delivering Stable High-Quality Video: An SDN Architecture with DASH Assisting Network Elements *(MMSys '16)*. Association for Computing Machinery, New York, NY, USA, Article Article 4, 10 pages. https://doi.org/10.1145/2910017.2910599

[16] J. W. Kleinrouweler, S. Cabrero, R. van der Mei, and P. Cesar. 2015. Modeling Stability and Bitrate of Network-Assisted HTTP Adaptive Streaming Players. In *2015 27th International Teletraffic Congress*. 177–184. https://doi.org/10.1109/ITC.2015.28

[17] Jan Willem Kleinrouweler, Britta Meixner, and Pablo Cesar. 2017. Improving Video Quality in Crowded Networks Using a DANE *(NOSSDAV'17)*. Association for Computing Machinery, New York, NY, USA, 73–78. https://doi.org/10.1145/3083165.3083167

[18] Stefan Lederer, Christopher Müller, and Christian Timmerer. 2012. Dynamic Adaptive Streaming over HTTP Dataset *(MMSys '12)*. Association for Computing Machinery, New York, NY, USA, 89–94. https://doi.org/10.1145/2155555.2155570

[19] L. Li, M. Pal, and Y. R. Yang. 2008. Proportional Fairness in Multi-Rate Wireless LANs. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*. 1004–1012. https://doi.org/10.1109/INFOCOM.2008.154

[20] Z. Li, X. Zhu, J. Gahm, R. Pan, H. Hu, A. C. Begen, and D. Oran. 2014. Probe and Adapt: Rate Adaptation for HTTP Video Streaming At Scale. *IEEE Journal on Selected Areas in Communications* 32, 4 (April 2014), 719–733. https://doi.org/10.1109/JSAC.2014.140405

[21] Harald Ott, Konstantin Miller, and Adam Wolisz. 2017. Simulation Framework for HTTP-Based Adaptive Streaming Applications *(WNS3 '17)*. Association for Computing Machinery, New York, NY, USA, 95–102. https://doi.org/10.1145/3067665.3067675

[22] Stefan Pham, Patrick Heeren, Daniel Silhavy, and Stefan Arbanowski. 2019. Evaluation of Shared Resource Allocation Using SAND for ABR Streaming *(MMSys '19)*. Association for Computing Machinery, New York, NY, USA, 165–174. https://doi.org/10.1145/3304109.3306227

[23] M. Richart, J. Baliosian, J. Serrat, and J. Gorricho. 2016. Resource Slicing in Virtual Wireless Networks: A Survey. *IEEE Transactions on Network and Service Management* 13, 3 (Sep. 2016), 462–476. https://doi.org/10.1109/TNSM.2016.2597295

[24] D. C. Robinson, Y. Jutras, and V. Craciun. 2012. Subjective video quality assessment of HTTP adaptive streaming technologies. *Bell Labs Technical Journal* 16, 4 (March 2012), 5–23. https://doi.org/10.1002/bltj.20531

[25] I. Sodagar. 2011. The MPEG-DASH Standard for Multimedia Streaming Over the Internet. *IEEE MultiMedia* 18, 4 (April 2011), 62–67. https://doi.org/10.1109/MMUL.2011.71

[26] K. Spiteri, R. Urgaonkar, and R. K. Sitaraman. 2016. BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[27] E. Thomas, M. O. van Deventer, T. Stockhammer, A. C. Begen, and J. Famaey. 2017. Enhancing MPEG DASH Performance via Server and Network Assistance. *SMPTE Motion Imaging Journal* 126, 1 (Jan 2017), 22–27. https://doi.org/10.5594/JMI.2016.2632338

[28] L. Yang, Y. Cui, H. Tang, and S. Xiao. 2015. Demand-Aware Load Balancing in Wireless LANs Using Association Control. In *2015 IEEE Global Communications Conference (GLOBECOM)*. 1–6. https://doi.org/10.1109/GLOCOM.2015.7417003

[29] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP *(SIGCOMM '15)*. Association for Computing Machinery, New York, NY, USA, 325–338. https://doi.org/10.1145/2785956.2787486

[30] S. Zehl, A. Zubow, and A. Wolisz. 2016. BIGAP — A seamless handover scheme for high performance enterprise IEEE 802.11 networks. In *NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*. 1015–1016. https://doi.org/10.1109/NOMS.2016.7502950