JurCast: Joint User and Rate Allocation for Video Multicast over Multiple APs

Hui Wang, Wei Tsang Ooi and Mun Choon Chan
School of Computing, National University of Singapore
Email:{wanghui,ooiwt,chanmc}@comp.nus.edu.sg

***Abstract*—Wireless multicast has been exploited to bridge the gap between the limited wireless bandwidth and the rapidly increasing mobile video traffic demand. Multicast of videos to a set of heterogeneous users over multiple wireless access points, however, is challenging because of the trade-offs between high transmission rate, load balancing, and multicast opportunities.**

In this paper, we present JurCast, a joint user and rate allocation scheme for video multicast over multiple APs. Our approach balances the trade-off between these factors by determining user to Access Points (APs) association, the video resolution version (quality) to be delivered for each session, and the transmission link rate for each video version. The aim of our solution is to maximize the overall received video quality over all users. We have implemented and evaluated our solution on a WiFi testbed as well as the simulation of a large scale deployment. The results indicate that our method considerably outperforms the baseline schemes and achieves up to 3dB and 55% improvements in terms of peak signal-to-noise ratio (PSNR) and goodput, respectively.

I. Introduction

Mobile video streaming has became tremendously popular in recent years. The growth of mobile video traffic is driven by a rapid increase in the number of hand-held devices (e.g., smartphones, tablets). According to the latest Cisco Visual Index report [6], mobile video traffic will increase 13-fold from 2014 to 2019 and will account for 72% of the total mobile data traffic by the end of 2019, up from 55% in 2014. Moreover, high-definition (HD) videos have become universally available and will be the dominant form of video content, contributing to the increasing amount of video traffic.

A recent study [1] demonstrates that the majority of mobile video traffic is offloaded to WiFi. Over the past decade, WiFi networks have been widely deployed, and its adoption is still growing. The current WiFi systems, however, are still inadequate in providing satisfactory quality when streaming video to a large number of users, especially for HD videos. To bridge the gap between the rapidly increased video traffic and the limited bandwidth provided by 802.11 networks, a considerable research effort has been devoted to improve the performance of wireless streaming systems ([3], [4], [14], [18], [20]).

To increase capacity, multiple access points (APs) are typically deployed ([3], [14]). To utilize the deployed multiple APs more effectively, the mobile clients have to make intelligent decision about which AP to associate with. Apart from deploying more APs, exploiting wireless multicast is another efficient way to improve the system utilization while minimizing the wireless resource usage ([3], [18], [20]). Wireless multicast is a natural operation for delivering traffic to multiple clients simultaneously. This work allows each client to subscribe multiple multicast sessions, which can arise in many scenarios, such as broadcast live sport events (e.g., World Cup, Olympic Games) and game streaming (e.g., Twitch Streams [7], [17]).

Video broadcast over 802.11 wireless networks with multiple deployed APs, however, is challenging because of the conflicts between high transmission rate (associated AP), load balancing, and exploiting multicast opportunities. More specifically, if a client simply chooses the AP with the highest receiving Received Signal Strength Indicator (RSSI), this could result in severe unbalanced workload between APs and reduced multicast opportunities.

In this paper, we present JurCast, a joint user and rate allocation scheme for video multicast over multiple APs. In particular, the following problem is considered: given a set of clients in the system, the set of videos interested by each client (multiple subscriptions are allowed), and the estimated link condition between each client and each AP, how to determine (i) the client to AP association, (ii) the resolution level of each interested video to be delivered to each client, and (iii) the multicast transmission rate for each video version. Our goal is to maximize the overall perceived video quality of all clients. To this end, we first build a novel model to characterize this maximization problem, and then propose a heuristic algorithm to effectively solve the formulated problem.

To summarize, our key contributions are:

- A model that jointly characterizes the association schedule and the multicast resource allocation to maximize the overall system utility;
- A methodology to simplify the model and based on the simplified model, we suggest an effective heuristic to solve the maximization problem;
- Evaluation of the proposed heuristic, including both testbed implementation and large scale simulation, to show that, compared to the baseline schemes, our approach significantly improves the video quality (PSNR) and goodput by up to 3dB and 55%, respectively.

The rest of this paper is organized as follows. We discuss the related work in the next section. In Section III, we build a model to capture the allocation problem and present our JurCast solution. Section IV describes the implementation details of our testbed. The evaluation is presented in Section V. We conclude in Section VI.

II. Related Work

A large body of prior work explores adaptive wireless multicast streaming system. This section outlines the most relevant pieces of work, which falls in the following categories.

Application-layer rate adaptation. A tremendous amount of early work dynamically adapts application layer data rate (including video frames and FEC) to erroneous channel conditions for wireless multicast transmissions ([19], [13], [9], [10]). Wu et al. [23] present an adaptive framework that consists of scalable video representation, network-aware end systems, and adaptive services. Under this framework, the streaming system is able to provide smooth change of perceptual quality to clients as wireless channel condition change. Subsequently, adaptive layered FEC-based control mechanisms are investigated ([19], [13]). Most recently, Choi et al. [5] exploit FEC from multiple APs to achieve reliable video multicast. Most of these works adapt application layer data rate with fixed underlying multicast transmission rate, which may severely under-utilize the network resource.

Multicast data rate adaptation. Recently, to exploit the limited wireless channel resource and alleviate rate anomaly problem, adaptive multicast link rate mechanisms have been studied [22], [16], [3], [18]. Instead of transmitting at the basic rate, a relatively high broadcast rate is used for packet delivery [22], [16], [3]. Inspired by the intuition that high link rate typically leads to high loss rate, Medusa [18] prioritizes the frames according to their importance and transmits the less important frames at higher link rates. By utilizing this frame level rate assignment heuristic, Medusa achieves higher video quality with limited resources.

To further improve streaming performance, numerous recent approaches jointly adapt video data rate and multicast link rate. Deb et al. [21] investigate the utility optimization problem of scalable video multicast and prove that this problem is NP-Hard. A greedy algorithm is then proposed to schedule the transmissions of layers and determine the corresponding modulation and coding scheme (MCS) assigned for each transmission. Li et al. [11], [12] suggest a pseudo-polynomial algorithm with dynamic programming to solve the optimization problem. These solutions, however, fail to effectively solve the optimization problem with multiple multicast sessions. Most recently, Wang et al. [20] design an optimal algorithm that substantially reduces the computational complexity for the case of multiple sessions supported by a single AP.

Multicast association control. To significantly improve the wireless system capacity, a dense deployment of access points (APs) is used instead of a single AP [14]. Chen et al. [4] propose approximation algorithms to *maximize the number of users*, *balance the load among the APs*, or *minimize the load of the APs*. In their work, however, each user is only allowed to subscribe to a single multicast video session. To address this shortcoming, DirCast [3], which is designed to support multiple multicast subscriptions, is proposed. DirCast, however, has an exponential running time respect to the number of available multicast sessions. Finally, none of these works on multicast association control takes adaptive video streaming into account. Our work is unique in that we consider a problem whereby all the above mentioned adaptive techniques (application as well as link layers) are integrated and applied in the settings with multiple APs.

III. JurCast Design

A. Preliminaries and Assumptions

The problem studied in this paper is as follows. There are N_{AP} access points in the system, and we assume that the neighboring APs operate on non-overlapping channels, the same assumption that has been made in [3], [4], [14]. The available capacity of each AP is T in terms of the available time slots for multicast transmissions. The number of distinct link rates is N_r; and the number of video sessions is N_v. We have a set of n clients, along with the interested video sessions of each client, and the estimated link rates between each client and each AP. The objective of our algorithm is to allocate resource for multicast video frames to maximize the total system utility of all clients.

Video Encoding. A video sequence is partitioned into Group of Pictures (GOP) with a certain number of frames. Each GOP consists of I, P, and B frames. For simplicity, we assume that the number of frames in a GOP is fixed to be J for any video sequence and the duration of each GOP is 1 second, which can be easily generalized to videos with different frame rates. In adaptive video streaming system, each video is encoded into M resolution versions (or levels). The average frame size of video v at resolution m is denoted by s_v^m. In our model, the average analysis technique is applied to simplify the resource allocation problem. More specifically, transmitting a GOP of frames could be regarded as transmitting a single frame with the average frame size. Thus, the network capacity T of each AP is set as the number of 802.11 slots in $1/J$ second.

Utility Assignment. Let $\mathbf{V}(i)$ be the set of video sessions interested by client i. The resolution of video v ($v \in \mathbf{V}(i)$) requested by this client is denoted by R_{iv}. Due to the dissimilarity of videos in bandwidth consumption, popularity, and priority, the requested video resolution levels of a client could be different. Note that if video v is not in video set $\mathbf{V}(i)$, we have $R_{iv} = 0$. Since the available bandwidth is rather restricted, we may not be able to meet the requests from all clients. As a result, some videos may be streamed with resolution levels lower than the desired resolution levels. To avoid significant quality degradation, we have the lowest resolution level guaranteed to be received, which is L_{iv}. Receiving video v^m at client i yields utility u_{iv}^m. It is clear that $u_{iv}^m = 0$ for all $v \notin \mathbf{V}(i)$. For video session $v \in \mathbf{V}(i)$, the utility function should follows the following rules:

$$\begin{aligned}
&u_{iv}^m = -\infty, \text{if } m < L_{iv}, \\
&u_{iv}^m < u_{iv}^{m'}, \text{if } L_{iv} \le m < m' \le R_{iv}, \\
&u_{iv}^m = u_{iv}^{R_{iv}}, \text{if } m > R_{iv}.
\end{aligned} \tag{1}$$

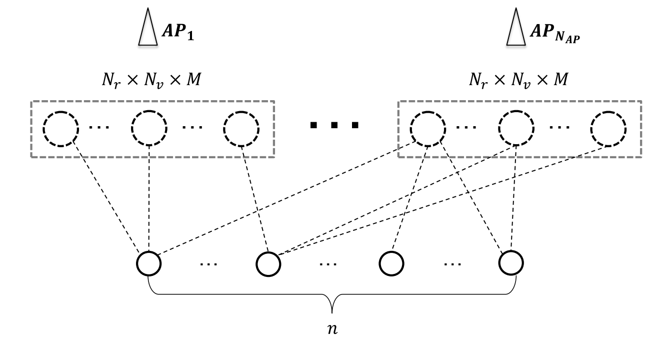


Fig. 1. The network model. We have $N_r \times N_v \times M$ transmission states for each AP station and n client nodes. Each state is represented by the dash circle and the client node is represented by the circle with solid line.

The utility function can be any general function subject to the above constraints. Here, we use the estimated PSNR as the utility function, where the highest achievable utility $u_{iv}^{R_{iv}}$ is the coding PSNR of video v at resolution level R_{iv}. For a resolution level m lower than R_{iv}, the frame with requested resolution level R_{iv} is used as the reference frame to calculate the received PSNR.

TABLE I
KEY NOTATIONS IN THE ALGORITHM

Notation	Definition
T	AP capacity in terms of time slots (802.11 slots)
n	Number of clients
N_{AP}	Number of APs
r_{ij}	The estimated link rate of between client i and AP_j
N_r	Number of distinct link rate levels
N_v	Number of video sessions
M	The number of available resolution levels
$\mathbf{V}(i)$	The set of videos interested by client i
R_{iv}	Resolution level of video v requested by client i
L_{iv}	Lowest resolution level of video v guaranteed to client i
v^m	Video v at resolution level m
s_v^m	The size (in bytes) of video v at resolution level m
u_{iv}^m	Utility of video v at resolution level m assigned to client i
$x_{ij} = 1$	Client i is associated with AP_j
$y_{jk} = 1$	Transmission state k is scheduled by AP_j for delivery
$z_{ijk} = 1$	Client i attains utility from transmission state k of AP_j

B. Problem Formulation

For each AP station, a possible *transmission state* is identified by *multicast link rate*, *video session id*, and *video resolution level*. Therefore, there are $N = N_r \times N_v \times M$ distinct transmission states in total for each AP. We build a network model to characterize our problem, which is shown in Figure 1. In the figure, each transmission state is represented by a dash circle. For state k ($1 \leq k \leq N$) of AP_j, the corresponding multicast link rate level, video id, and resolution level are k_r, k_v, and k_m, respectively. If the link rate between client i and AP_j is greater than k_r (i.e., $r_{ij} \geq k_r$) and this client is interested in video k_v (i.e., $k_v \in \mathbf{V}(i)$), a dash line is added between this client and state k of AP_j. The attainable utility is $u_{ik_v}^{k_m}$.

The network model shown in Figure 1 clearly demonstrates that solving our maximization problem is to optimally determine: (i) which client should be associated with which AP; (ii) which AP should schedule which transmission states subject to the bandwidth capacity constraints; and (iii) which client should attain utility from which scheduled and associated (dash lines from the associated AP that is determined by (i)) transmission states.

Binary variable $x_{ij} = 1$ indicates that client i is associated with AP_j. The constraint that *every client is associated to exactly one AP* can be formulated by the equation: $\sum_{j=1}^{N_{AP}} x_{ij} = 1$.

We define an indicator variable y_{jk}, which takes the value of 1 if the transmission state k is scheduled at AP_j for delivery. The time cost of transmitting this particular state equals to the video size over the link rate, which can be written as $s_{k_v}^{k_m}/k_r$, where $s_{k_v}^{k_m}$ is the frame size of video k_v at resolution level k_m. As the overall multicast traffic at each AP should be subject to the capacity limit T, we have the following expression:

$$\sum_{k=1}^{N} y_{jk} \cdot \left\lceil \frac{s_{k_v}^{k_m}}{k_r} \right\rceil \leq T, \quad \text{for all } 1 \leq j \leq N_{AP}, \tag{2}$$

where $N = N_r \times N_v \times M$.

We have another indicator variable z_{ijk}, which takes the value of 1 if client i attains utility from the transmission state k of AP_j. Regarding the value of the binary variable z_{ijk}, we have the following two restrictions:

(i) The value of $z_{ijk} = 1$ implies that client i should be associated with AP_j ($x_{ij} = 1$) and state k is scheduled to be delivered at AP_j ($y_{jk} = 1$). This dependence relationship could be formally written as:

$$z_{ijk} \leq x_{ij} \cdot y_{jk}, \quad \text{for all } 1 \leq i \leq n, 1 \leq j \leq N_{AP}, 1 \leq k \leq N; \tag{3}$$

(ii) Client i attains utility from exactly one state for any particular interested video session $v \in \mathbf{V}(i)$: $\sum_{k=1}^{N} z_{ijk} = 1$, where $k_v = v$.

In summary, we formulate our maximization problem as

$$\begin{aligned} \text{maximize } & \sum_{i=1}^{n} \sum_{j=1}^{N_{AP}} \sum_{k=1}^{N} z_{ijk} \cdot u_{ik_v}^{k_m}, \\ \text{subject to } & \sum_{j=1}^{N_{AP}} x_{ij} = 1, \\ & \sum_{k=1}^{N} y_{jk} \cdot \left\lceil \frac{s_{k_v}^{k_m}}{k_r} \right\rceil \leq T, \\ & z_{ijk} \leq x_{ij} \cdot y_{jk}, \\ & \sum_{k=1}^{N} z_{ijk} = 1, \forall v \in \mathbf{V}(i), k_v = v. \end{aligned} \tag{4}$$

which is a 0-1 integer programming problem with three variables: x_{ij}, y_{jk}, and z_{ijk}. As multiple binary variables are present in this integer programming formulation, it is generally unclear how to solve it efficiently. In the subsequent section, we first present a methodology to simplify the model and then suggest an efficient heuristic algorithm.

C. Greedy Algorithm

The previous section formulated the problem as a 0-1 integer programming problem with a considerable number of constraints and three binary variables, which is difficult to solve. In this section, we will show how to simplify these constraints.

1) Eliminate Dependency of States: At any point of time, a client only associates to exactly one AP, which leads to the dependency of scheduling transmission states. More specifically, one client only attains utility from transmission states that come from an identical AP (the associated AP), which is not captured by the model in Figure 1. To eliminate the scheduling dependency, we design a new network model to characterize this issue, where each state takes a set of videos (interested by a client) instead of a single video as an element.

We define $\mathcal{V} = \{\mathbf{V}(1), \mathbf{V}(2), \ldots, \mathbf{V}(n)\}$, where element $\mathbf{V}(i)$ is the set of videos interested by client i. For a particular element $\mathbf{V}(i)$, the total number of resolution level combinations is $M^{|\mathbf{V}(i)|}$, which is clearly too large to be practical as multiple subscription is allowed. Instead of enumerating all possible combinations, we use an uniform resolution level for all videos in $\mathbf{V}(i)$, and the number of combinations for element $\mathbf{V}(i)$ is reduced to M. As the request resolutions of videos in $\mathbf{V}(i)$ could be different, this design may reduce the cost efficiency. We will present how to alleviate the inefficiency in the following sections.

Now we have $N' = N_r \times |\mathcal{V}| \times M$ transmission states for each AP. Similarly, for state k ($1 \leq k \leq N'$), the corresponding video set is represented by $k_{\mathbf{V}}$ and the transmission cost is calculated by $(\sum_{v \in k_{\mathbf{V}}} s_v^{k_m})/k_r$. For a scheduled state k from AP_j, the set of clients that can benefit from this state is: $\{i|\mathbf{V}(i) \subseteq k_{\mathbf{V}}, r_{ij} \geq k_r\}$. With the pre-computed client list, the attainable utility for each state can be easily calculated.

2) Update Residual Utility: Apart from the above association restriction, another critical restriction is that client can only receive one resolution level of each interested video and attain the corresponding utility. In our new model, this restriction implies that a client can only attain utility from exactly one state that covers all interested video sessions. To avoid redundant utility counting, we calculate the residual utility for each state. Once a transmission state is scheduled for delivery, we have to update the residual utility for all the correlated unscheduled states.

3) Quantify Cost of AP: The last restriction is that the overall multicast traffic from each AP should not exceed the capacity. Although the new network model eliminates the states dependency, the benefit of wireless multicast is not taken into consideration. Therefore, integrating the cost of all scheduled transmission states at each AP will considerably overestimate the workload, which is mainly due to the following issues: (i) scheduling multiple states that partially overlap in video sessions could result in redundant transmissions of the overlapped videos; (ii) to reduce the number of states, an uniform resolution level is transmitted for the videos from each state. As a result, the video transmissions of this state may not be fully utilized when the request resolution level of some video is lower than the transmitted resolution level; and (iii) if a client attains utility from a new scheduled state at a different AP, this client will re-associate to this new AP. The occupied bandwidth consumption of previous AP, however, is not revoked.

To address the afore discussed issues, we apply a dynamic programming algorithm to accurately predict the workload for each AP instead of integrating the cost of the scheduled states. Given the existing selected transmission states, the set of clients that are associated to a particular AP_j is denoted as $\mathbf{C}_j$. To simplify notations, we relabel the client ids in $\mathbf{C}_j$ as $1, 2, \ldots, |\mathbf{C}_j|$. Without loss of generality, we assume that the physical link rate sequence $r_{1j}, r_{2j}, \ldots, r_{|\mathbf{C}_j|j}$ is non-decreasing. The union of videos covered by these selected transmission states is represented by $\mathbf{V}_j = \cup_{i \in \mathbf{C}_j} \mathbf{V}(i)$.

For client i ($1 \leq i \leq |\mathbf{C}_j|$) and interested video session v ($v \in \mathbf{V}(i), v \in \mathbf{V}_j$), given the existing scheduled states, the highest effective resolution level expected to receive is denoted by h_{iv} ($h_{iv} \leq R_{iv}$). The remaining part of this section illustrates how to calculate the minimum required time slots for AP_j while satisfying the resolution level requirement h_{iv}, for all $i \in \mathbf{C}_j, v \in \mathbf{V}(i)$.

We separately analyze the transmission time slots for each video session v ($v \in \mathbf{V}_j$). The resolution level of video v expected to be received by client i can be transmitted at link rate r_{ij} or at lower link rate $r_{i'j}$, where $1 \leq i' \leq i$. Define $T_v(i,l)$ as the minimum required time slots satisfying the requirements from clients 1 to i and at least one transmitted resolution level should be greater than l (the level required from clients with indexes larger than i). The recursive equation for $T_v(i,l)$ can be written as

$$T_v(i,l) = \min\left\{T_v(i-1,H), T_v(i-1,0) + \left\lceil \frac{s_v^H}{r_{ij}} \right\rceil \right\}, \quad (5)$$

where $H = \max\{l, h_{iv}\}$. The minimum time slots required for delivering video v while satisfying the quality lower bound is $T_v(|\mathbf{C}_j|, 0)$, which could be easily calculated by leveraging recursion (5). The cost of AP_j is expressed as

$$T_j = \sum_{v \in \mathbf{V}_j} T_v(|\mathbf{C}_j|, 0). \quad (6)$$

4) Greedy Algorithm: By leveraging the simplified model in previous subsections, we develop a heuristic algorithm that greedily chooses the transmission state with the maximum cost efficiency in every iteration among the unselected states.

The greedy algorithm procedure is illustrated in Algorithm 1. Let $S_{j,k}$ be state k at AP_j. We define the initial set of the unselected states as $\mathcal{S} = \{S_{j,k}|1 \leq j \leq N_{AP}, 1 \leq k \leq N'\}$.

After this greedy iteration procedure, we obtain the "best" AP association arrangement for every client. With the obtained

Algorithm 1 Greedy Algorithm

1: Compute the cost $T(S_{j,k})$ and utility $u(S_{j,k})$ of state $S_{j,k}$.
2: **repeat**
3: Compute the cost efficiency $\frac{u(S_{j,k})}{T(S_{j,k})}$.
4: Let S^* be the unscheduled state with maximum cost efficiency, while $T_{j^*} \leq T$, update $\mathcal{S} = \mathcal{S} \backslash S^*$.
5: Update cost of T_{j^*} using Equations (5) and (6).
6: Update residual utility of all relevant states in $\mathcal{S}$.
7: **until** S^* cannot be found
8: Compute the optimal utility for each AP with the above determined association.

association, we run an allocation algorithm for each AP. The total number of states in our model is $N_{AP} \times N_r \times |\mathcal{V}| \times M$, where $|\mathcal{V}|$ ($|\mathcal{V}| \leq n$) is the number of unique elements in $\mathcal{V}$.

5) Optimal Allocation of Single AP: If we set $N_{AP} = 1$, the maximization problem (Equation (4)) is reduced to the optimal resource allocation problem of a single AP. To solve this maximization problem, we use the optimal allocation solution proposed by Wang et al. [20]. This algorithm employs dynamic programming to optimally determine which resolution of which video session should be transmitted at which link rate.

IV. Implementation

In this section, we describe our testbed setup and the implementation details.

Testbed Setup: Our testbed consists of the following components: (i) a video server that runs on a Mac Pro with a 3.2 GHz Quad-Core processor and 8GB memory; (ii) a gateway runs on a typical Linux machine with 3.4 GHz Quad-Core processor and 8GB memory; (iii) two APs with IEEE 802.11abg adapters featuring the Atheros AR5414 chipset and runs OpenWRT Kamikaze 7.09 with kernel version 2.6.25.16. The driver of the wireless adapter used in MadWifi (version 0.9.4); and (iv) the mobile devices, all LG Nexus 5.

The video server, gateway, and WiFi APs are all connected through wired Ethernet. The mobile devices communicate with the associated AP using IEEE 802.11a operating at 5GHz.

Multicast Rate Adaptation: In addition to transferring data between clients and the gateway, the primary modification at the AP is to support multicast link rate adaptation. To enable the packet level rate adaptation, we significantly extend and modify the Click modular router [8] (version 1.6.0), which is installed on each AP. The multicast transmission rate of each video packet is determined by the gateway and specified in the packet header. We extract the rate value from the header and pass the assigned value to the MadWifi driver.

The above assigned multicast link rate level closely depends on the wireless link conditions between each client and the associated AP. As the signal-to-noise ratio (SNR) of each received packet is not exposed at the smartphones, we used the measure received signal strength indicator (RSSI) to roughly estimate the initial link rate and use frame loss rate that is reported periodically from each client as a basis for rate adaptation. The History-Aware Robust Rate Adaptation Algorithm (HA-RRAA) [15] is implemented and employed in our testbed.

RaptorQ FEC: Since there is no MAC level retransmission for wireless multicast, the clients may not receive all packets. To overcome packet losses, the redundant FEC packets are transmitted in advance with the source video data. In our experiment, we leverage the library provided by OpenRQ [2] that implements the RaptorQ FEC scheme described in RFC 6330. The block size is set to 256. The number of FEC packets is adaptively determined by the thresholds in link rate adaptation and the history frame loss rate. The average encoding time cost measured over a block of 256 packets (1470 bytes per packet) is 61.27ms. As there is no coding overhead for the source symbols, we can transmit the source packets while generating the redundant FEC packets.

V. Evaluation

In this section, we perform two sets of measurements to evaluate the efficacy of our approach. In the first set, we evaluate our approach using a wireless testbed with up to 8 Android smartphones. In the second set, we evaluate large scale scenarios through simulation of up to 50 users.

Reference Schemes: We compare the performance of our JurCast with the following two adaptive approaches.

Best-RSSI based association (Best-RSSI): This scheme employs the traditional WiFi association control mechanism that each client chooses the AP with the highest RSSI value received. With the determined association, we run the same optimal allocation algorithm used in JurCast to obtain the allocation scheme for each AP separately.

Customized DirCast (DirCast$^+$): We choose DirCast approach [3] as another baseline algorithm because it is the prominent study that addresses the similar multicast problem over multiple APs. To make our comparison fairer, we incorporated adaptive video quality into DirCast using a heuristic that enumerates the resolution level for each state in DirCast.

Clusters: In real world networks, the clients are often unevenly distributed across all the available APs. Although every client is typically covered by many APs, the clients tend to be close to some particular APs. This clustering phenomenon is common in many scenarios, such class/conference room, concert, stadium. Here, a cluster refers to *a group of clients that associate with an identical AP at the initial status*, where the best RSSI association mechanism is employed. In the experiments, we vary the number of clusters by generating different initial statuses.

Metrics: The perceived video quality is measured by PSNR (peak signal-to-noise ratio), which is widely used by prior works [18], [24], [20]. In addition to measure video quality, we also characterized the network performance by goodput. Since multicast is enabled, multiple video versions may be received. For each client, every interested video with exactly one resolution level (minimum of the highest received and the requested levels) will contribute to the goodput.

Video Coding: Our video server encodes videos using the standardized FFmpeg tool (version 2.4.3) with H.264 codec.

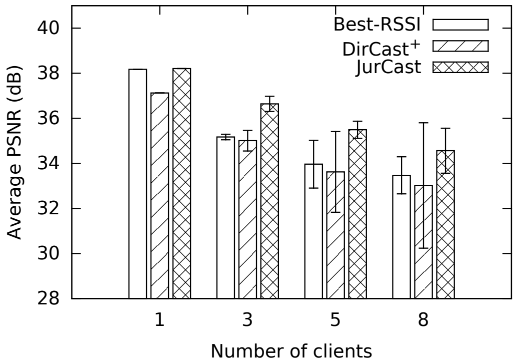


Fig. 2. Average PSNR per video session.

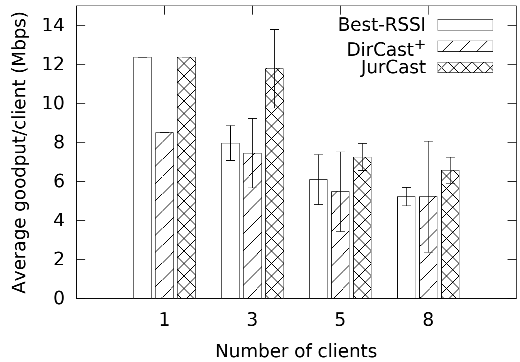


Fig. 3. Average Goodput per client.

In the experiment the full HD video sequences are encoded at 10Mbps with 25fps. We generate five resolution levels for each video sequence and the numbers of pixels at different resolution levels are: 1920×1080, 1600×900, 1280×720, 960×540, and 640×360.

A. Testbed-Based Evaluation

We evaluate the performance of JurCast and compare it to Best-RSSI and DirCast$^+$ schemes using up to 8 mobile devices. In particular, the benefits of *balancing workload* and *exploiting wireless multicast* are evaluated in the following two subsections, respectively.

1) Baseline Comparison: As only two APs are deployed in the testbed, we initially cluster all mobile devices to one AP by deploying them relatively close to one particular AP, while each client can also communicate with another AP at a lower transmission rate. On the other hand, to create distinct channel conditions between the clients and APs, these smartphones are placed in different locations.

In these experiments, we have 6 video sessions in total and each client randomly subscribes 2 video sessions. The request resolution levels of the interested video sessions are between 3 to 5.

The average PSNR values with error bars (standard deviation) across different clients are depicted in Figure 2. For each multicast scheme, we aggregate the PSNR values of all videos for each client and present the average of them from 10 runs. The result from the figure shows that JurCast significantly outperforms other two schemes when multiple clients are present in the system. On the average, JurCast improves the video quality by about 2dB over Best-RSSI and DirCast$^+$. As expected, the video quality reduces for all three schemes as the number of mobile devices increase. From the figure, we reveal that the highest video quality improvement between our approach and other methods is achieved when there are 3 or 5 users. As more workload is introduced by more devices, the enhancement is slightly reduced.

The figure also shows that the DirCast$^+$ performs slightly worse than Best-RSSI scheme, especially when only a single client is present. Best-RSSI employs the optimal resource allocation algorithm to intelligently determine the resolution version transmitted for each video session. By contrast, DirCast$^+$ takes the uniform resolution version to transmit for a set of videos interested by one client, where the transmissions may not be fully utilized. Moreover, DirCast$^+$ cannot benefit from the association control with one or few clients.

During the same experiments, we also measure the goodput for each client and present the results in Figure 3 with respect to the different number of clients. A similar goodput pattern is present as that of PSNR. Since goodput is also closely related to the video packet receptions, the more packets a client receives, the higher goodput and PSNR values are observed. In particular, the goodput improvements over other two schemes with 3 and 5 clients are about 40% and 30%, respectively.

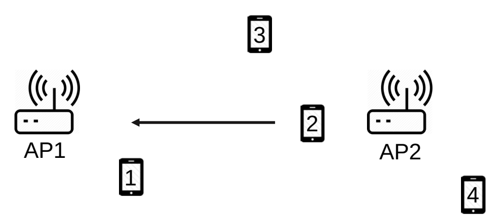
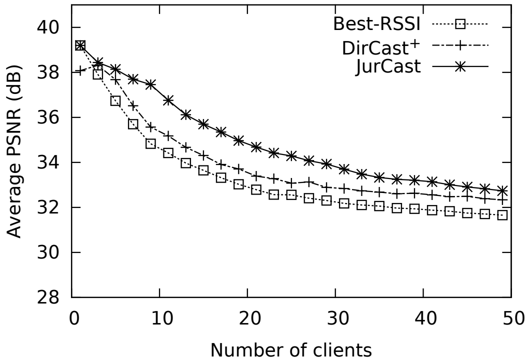


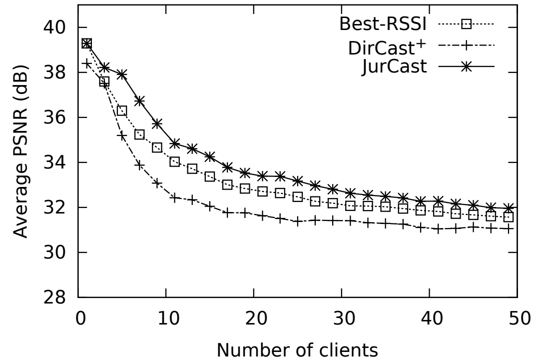
Fig. 4. Mobility experiment: the testbed consists of two APs and four clients. Client 2 is the mobile client, the moving direction is represented by the arrow. Clients 1 and 3 subscribe the same set of videos; and clients 2 and 4 subscribe another set of videos.

2) Client Mobility.: This section measures the performance with client mobility. The experimental testbed consists of two APs and four clients, with three stationary and the other moving at walking speed. The illustration diagram is present in Figure 4. The frame PSNR value is plotted in Figure 5, where the moving period is between frame 100 to 400.

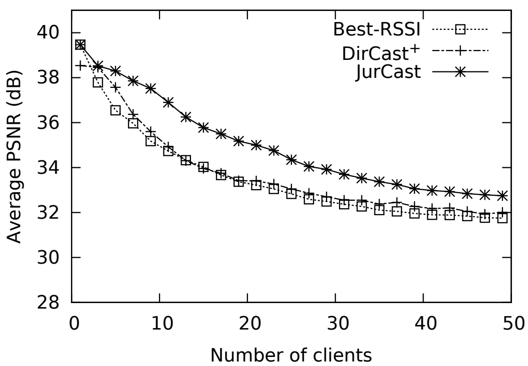
As depicted in the diagram, clients 3 and 4 are closer to AP_2 and client 1 is closer to AP_1, and these three clients are static. Client 2 is moving from AP_2 towards AP_1. Since clients 1, 2, and 3 are placed between two APs, they all observe fair wireless condition from the farther AP. According to this network condition, Best-RSSI scheme will group clients 2, 3, and 4 to AP_2 and associate client 1 to AP_1, in the initial stage. Since the network condition between client 3 and AP_1 is fair,



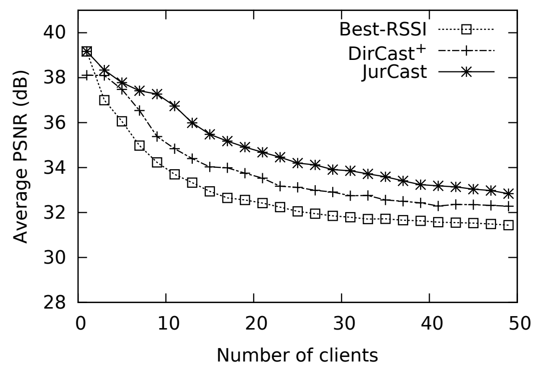
(a) 5 access points and 2 clusters



(b) 3 access points and 2 clusters



(c) 5 access points and 3 clusters



(d) 5 access points and 1 clusters

Fig. 6. Average PSNR per video with different configurations.

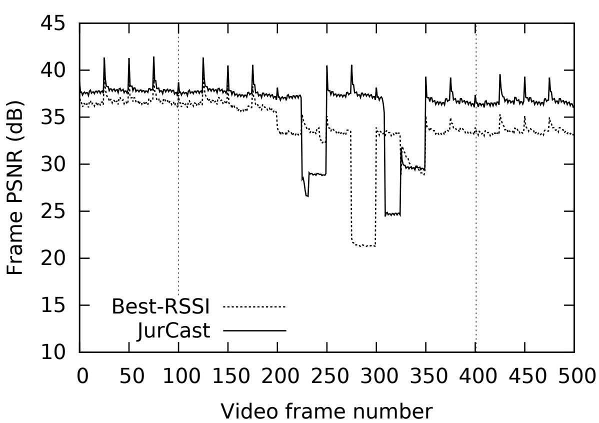


Fig. 5. Frame psnr value of the mobile client (client 2). The moving period is from frame number around 100 to 400.

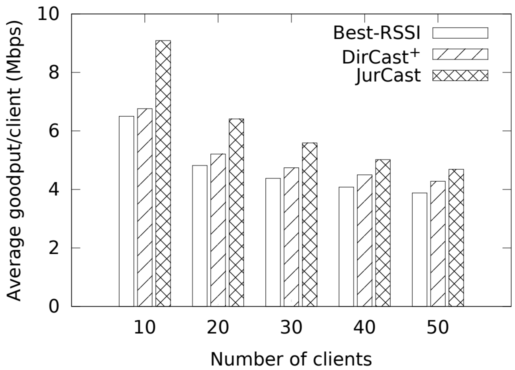
client 3 is associated to AP_1 to exploit wireless multicast in JurCast, which reduces the load of AP_2. As a result, the frame PSNR in JurCast is slightly higher than that of Best-RSSI before the movement.

During the mobility period (frame number 100 to 400), client 2 is moving towards AP_1, meanwhile, this client has been re-associated to AP_1 in Best-RSSI. In JurCast, client 2 is still associated with AP_2 as long as the supported link rate is fair, although it becomes lower as moving farther from AP_2. Figure 5 shows that after moving period (frame number 400), both JurCast and Best-RSSI suffer from video quality degradation. From the data after the movement, we can reveal the following two findings:

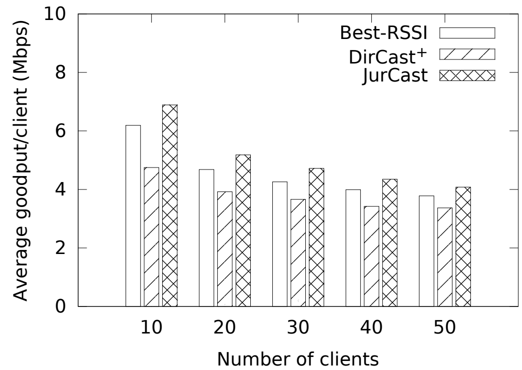
First, the frame PSNR of JurCast is slightly reduced as the supported multicast link rate is lower than the initial status. Second, client 2 in Best-RSSI experiences remarkably quality degradation because of the re-association. Although client 2 re-associates to the AP (AP_1) with lighter load and attains a higher link rate, which reduces the multicast opportunity. At the initial stage, clients 2, 3, and 4 are associated to AP_2, where clients 2 and 4 shares the identical interests. Benefiting from multicast and intelligent resource allocation, clients 2 and 4 will receive higher video resolution quality to maximize the overall utility. After the re-association, without multicast, client 2 suffers from considerably quality degradation as the load of AP_1 is substantially increased.

B. Simulation-Based Evaluation

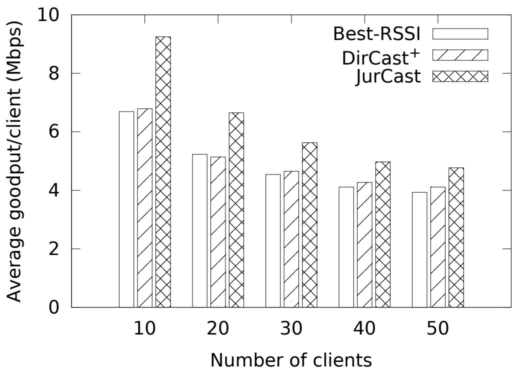
In this section, we extend our evaluation using simulations to determine the scalability of JurCast to larger deployments. Specifically, (i) we investigate the impact of workload by varying the number of clients and number of APs deployed in



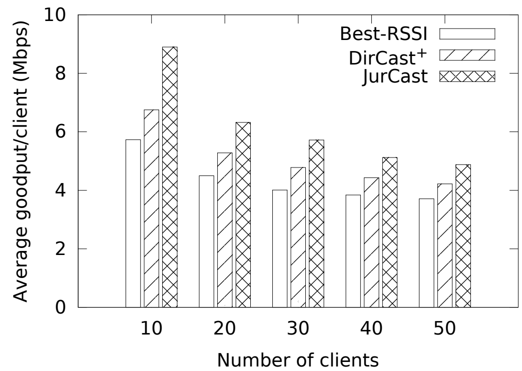
(a) 5 access points and 2 Clusters



(b) 3 access points and 2 clusters



(c) 5 access points and 3 clusters



(d) 5 access points and 1 clusters

Fig. 7. Average goodput per client with different configurations.

the system; (ii) we evaluate how the three methods perform with the different degrees of clustering (initial status); (iii) we present and discuss the algorithm computational overhead.

We implement these algorithms on ns-3 simulator (version 3.22) in C++. The neighboring APs operate on orthogonal 802.11a channels. To create the distinct link conditions, we randomly generate the received signal strength expected to be received at each client from different APs. Furthermore, we have 12 distinct video sessions, and each client randomly subscribes 3 videos.

The average PSNR and goodput values are presented in Figures 6 and 7, respectively. The results indicate that JurCast considerably outperforms the other two schemes under all configurations we evaluated. In particular, JurCast achieves up to 2.77dB and 3.04dB PSNR improvement, compared to DirCast$^+$ and Best-RSSI, respectively. Moreover, the corresponding goodput improvements are up to 45% and 55%.

1) Impact of Workload: In our measurement, we create different levels of workload by varying the number of clients (Figure 6(a)) and the number of APs (Figure 6(b)). It is clear that introducing more clients or deploying less number of APs leads to the heavier load. In particular, We examine the algorithm performance with up to 50 clients under the configuration with 5 or 3 APs.

Figure 6 clearly shows that all three schemes suffer from video quality degradation as more clients are deployed. JurCast achieves the highest improvement over other schemes when there are about 10 clients. When more than one clients are present, AP association control will be exploited by DirCast$^+$, especially for more APs. As a result, DirCast$^+$ achieves higher performance over Best-RSSI (shown in Figure 6(a)). Comparing Figures 6(a) and 6(b) confirms that deploying less number of APs also declines the PSNR value with the same number of clients.

2) Degree of Clustering: We evaluate the impact of the clustering degree at the initial stage, which relates to the decision of association control and therefore determines the transmission link rate. The degrees of clustering are distinct in all sub-figures of Figures 6. We make the following observations regarding these results.

First, the performance impacts of varying clustering degree on JurCast and DirCast$^+$ are almost unnoticeable when different number of clusters are configured (Figures 6(a), 6(c), and 6(d)). The observed consistent trends are mainly attributed to the employed association control mechanisms. With association control employed, to balance workload, most of clients are generally not associated with the AP that has the highest RSSI value.

Second, unlike other two schemes, the AP with the highest RSSI is associated to each client in Best-RSSI scheme. As a result, it is most sensitive to the changes of clustering degree. In particular, with the same number of APs deployed, the configurations with more clusters achieve a better video quality and goodput.

3) Algorithm Running Time: Apart from the measured performance, algorithm computational overhead is another paramount factor that determines the efficacy of the proposed solutions. In the previous experiments, the algorithm running time under the scenarios with 5 APs is recorded as well. Figure 8 plots the average running time values with respect to different number of clients.

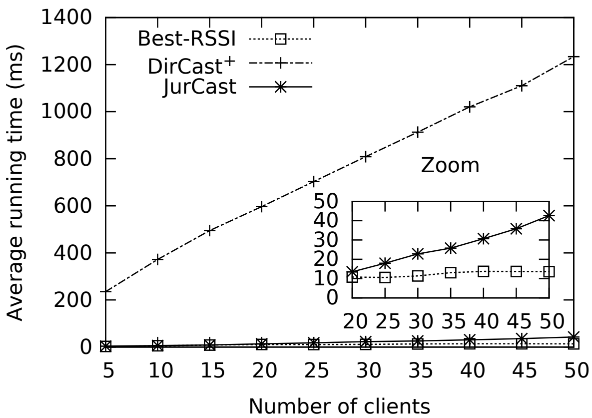


Fig. 8. Average algorithm running time. The number of available video sessions is 12 ($N_v = 12$).

The results show that the algorithm running time of JurCast and DirCast$^+$ is linearly increasing with the number of clients (denoted by n). The number of distinct transmission states in JurCast and DirCast are both linear correlated with the value of n. The coefficients of the linear functions for JurCast and DirCast with respect to the number of video sessions are $\mathcal{O}(N_v)$ and $\mathcal{O}(2^{N_v})$, respectively. As the quantity of the transmission states in DirCast is exponentially increasing with the number of video sessions, the running time of DirCast is substantially greater than JurCast, which is impractical to be employed in real-time adaptive system, especially with large N_v.

The Best-RSSI scheme employs an allocation algorithm that has complexity that is independent of the number of clients. As a consequence, with more clients introduced in the system, the running time does not vary much.

VI. Conclusion

This paper presents JurCast, a joint user and rate allocation scheme for video multicast over multiple APs. JurCast effectively allocates resource for multicast video streaming by balancing the trade-off between high transmission rate, load balancing, and multicast opportunities. Our extensive measurements in both testbed and large scale simulation demonstrate that JurCast significantly outperforms other compared adaptive schemes with up to 3dB and 55% enhancements in item of PSNR and goodput, respectively.

References

[1] Ericsson mobility report, June 2014. http://www.ericsson.com/res/docs/2014/ericsson-mobility-report-june-2014.pdf.

[2] OpenRQ: an open-source RaptorQ implementation. http://www.lasige.di.fc.ul.pt/openrq/.

[3] Ranveer Chandra, Sandeep Karanth, Thomas Moscibroda, Vishnu Navda, Jitendra Padhye, Ramachandran Ramjee, and Lenin Ravindranath. Dircast: A practical and efficient wi-fi multicast system. In *ICNP*, pages 161–170. IEEE, 2009.

[4] Ai Chen, Dongwook Lee, and Prasun Sinha. Optimizing multicast performance in large-scale wlans. In *ICDCS*, pages 17–17. IEEE, 2007.

[5] Munhwan Choi, Weiping Sun, Jonghoe Koo, Sunghyun Choi, and Kang G Shin. Reliable video multicast over wi-fi networks with coordinated multiple aps. In *INFOCOM, 2014 Proceedings IEEE*, pages 424–432. IEEE, 2014.

[6] Cisco Visual Networking Index. Global mobile data traffic forecast update, 2014-2019. *White Paper, February*, 2015.

[7] Mehdi Kaytoue, Arlei Silva, Loïc Cerf, Wagner Meira Jr, and Chedy Raïssi. Watch me playing, i am a professional: a first study on video game live streaming. In *WWW*, pages 1181–1188. ACM, 2012.

[8] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti, and M Frans Kaashoek. The Click modular router. *TOCS*, 18(3):263–297, 2000.

[9] T-WA Lee, S-HG Chan, Qian Zhang, Wen-Wu Zhu, and Ya-Qin Zhang. Optimal allocation of packet-level and byte-level fec in video multicasting over wired and wireless networks. In *GLOBECOM*, volume 3, pages 1994–1998. IEEE, 2001.

[10] T-WA Lee, S-HG Chan, Qian Zhang, Wen-Wu Zhu, and Ya-Qin Zhang. Allocation of layer bandwidths and fecs for video multicast over wired and wireless networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, 12(12):1059–1070, 2002.

[11] Peilong Li, Honghai Zhang, Baohua Zhao, and Sampath Rangarajan. Scalable video multicast in multi-carrier wireless data systems. In *ICNP*, pages 141–150. IEEE, 2009.

[12] Peilong Li, Honghai Zhang, Baohua Zhao, and Sampath Rangarajan. Scalable video multicast with joint layer resource allocation in broadband wireless networks. In *ICNP*, pages 295–304. IEEE, 2010.

[13] Abhik Majumdar, Daniel Grobe Sachs, Igor V Kozintsev, Kannan Ramchandran, and Minerva M Yeung. Multicast and unicast real-time video streaming over wireless lans. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):524–534, 2002.

[14] Rohan Murty, Jitendra Padhye, Ranveer Chandra, Alec Wolman, and Brian Zill. Designing high performance enterprise wi-fi networks. In *NSDI*, volume 8, pages 73–88, 2008.

[15] Ioannis Pefkianakis, S Wong, Hao Yang, S Lee, and Songwu Lu. Towards history-aware robust 802.11 rate adaptation. *IEEE Transactions on Mobile Computing*, 12(3):502–515, 2013.

[16] Kandaraj Piamrat, Adlen Ksentini, J-M Bonnin, and César Viho. Q-DRAM: QoE-based dynamic rate adaptation mechanism for multicast in wireless networks. In *GLOBECOM*, pages 1–6. IEEE, 2009.

[17] Karine Pires and Gwendal Simon. Dash in twitch: Adaptive bitrate streaming in live game streaming platforms. In *Proceedings of the 2014 Workshop on Design, Quality and Deployment of Adaptive Video Streaming*, pages 13–18. ACM, 2014.

[18] Sayandeep Sen, Neel Kamal Madabhushi, and Suman Banerjee. Scalable wifi media delivery through adaptive broadcasts. In *NSDI*, pages 13–13. USENIX Association, 2010.

[19] Wai-Tian Tan and Avideh Zakhor. Video multicast using layered fec and scalable compression. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(3):373–386, 2001.

[20] Hui Wang, Mun Choon Chan, and Wei Tsang Ooi. Wireless multicast for zoomable video streaming. *arXiv preprint arXiv:1505.01933*, 2015.

[21] Mathias Wien, Renaud Cazoulat, Andreas Graffunder, Andreas Hutter, and Peter Amon. Real-time system for adaptive video streaming based on SVC. *IEEE Transactions on Circuits and Systems for Video Technology*, 17(9):1227–1237, 2007.

[22] Starsky HY Wong, Ramya Raghavendra, Yang Song, and Kang-Won Lee. X-wing: A high-speed wireless broadcasting framework for ieee 802.11 networks. In *SECON*, pages 344–352. IEEE, 2013.

[23] Dapeng Wu, Yiwei Thomas Hou, and Ya-Qin Zhang. Scalable video coding and transport over broadband wireless networks. *Proceedings of the IEEE*, 89(1):6–20, 2001.

[24] Jongwon Yoon, Honghai Zhang, Suman Banerjee, and Sampath Rangarajan. MuVi: a multicast video delivery scheme for 4G cellular networks. In *MobiCom*, pages 209–220. ACM, 2012.