M^3: Practical and Reliable Multi-Layer Video Multicast over Multi-Rate Wi-Fi Network

Menghan Li*, Dan Pei*, Xiaoping Zhang*, Beichuan Zhang†, Zhi Wang‡, Hailiang Xu§, Zihan Wang‡
*Tsinghua National Laboratory for Information Science and Technology (TNList)
Department of Computer Science and Technology, Tsinghua University
†Department of Computer Science, University of Arizona
‡Graduate School at Shenzhen, Tsinghua University
§Department of Computer Science, Beijing University of Posts and Telecommunications
{limenghan12, wangzh12}@mails.tsinghua.edu.cn, {peidan, zhxp}@tsinghua.edu.cn
bzhang@cs.arizona.edu, wangzhi@sz.tsinghua.edu.cn, harlenxu@gmail.com

***Abstract*—IEEE 802.11-based wireless LAN, commonly referred to as Wi-Fi, has become a universal solution for the last-hop network access. In large and public assembly places, people may use their mobile devices to view the video of the same popular events via the same wireless access points (APs). However, current 802.11 APs transmit the same video stream multiple times via separate unicast sessions due to the well-known poor reliability and low data rate of the legacy Wi-Fi multicast. Besides, in traditional single-layer-coded video streams, all clients have to settle with the lowest video bitrate limited by the client with the worst channel quality. To address these problems, we propose M^3, a practical and reliable multi-layer video multicast solution over multi-rate Wi-Fi networks. The aims of our system are, in the premise of no change to APs, not only to ensure that all clients can smoothly watch the video at least with the lowest quality, but also to maximize the overall video quality received by all clients. To meet these design goals, the video server selects certain clients as unicast receivers to transmit different SVC video layers, and other clients listen for the packets in the promiscuous mode. It is challenging to select specific unicast receivers and allocate different SVC layers to fully utilize the available bandwidth because of dynamic network conditions. To overcome this challenge, we use a periodical feedback mechanism to collect necessary statistics from clients, and use them to derive an optimal SVC-layer allocation strategy to maximize the video quality. We implemented a prototype in a real Wi-Fi testbed consisting of one AP and one M^3 server and 8 clients. Compared with the single-layer video multicast, our M^3 system can improve the total received video rate by up to 200%.**

I. Introduction

Wi-Fi has been a universal solution for the last-hop access networks in large and public assembly places [1]. People can use their mobile devices to view live video whenever and wherever possible via Wi-Fi connections. Dynamic Adaptive Streaming over HTTP (DASH) [2] is a popular solution for Internet video streaming to dynamically adapt to varying network conditions. However, DASH-based end-to-end solution can hardly support multiple users viewing the same high-definition live video smoothly via a same wireless AP. Even a moderate number of users would lead to a severe channel congestion for one AP and further degrade QoS of all applications via that AP.

*Xiaoping Zhang is the corresponding author.

Multicast is a potential solution for supporting multiple users to view the same video via the same AP. However, the high-definition video streaming cannot be directly streamed over the legacy Wi-Fi multicast due to its poor reliability and low data rate. Besides, for a multi-rate WLAN scenario, different clients may have different channel qualities (*i.e.*, SNRs). Multicasting a traditional single-layer-coded video stream to multiple clients requires the AP to transmit at the lowest bit rate supported by their channels. This reduces all clients to the video quality of the client with the worst channel.

H.264/SVC [3] (hereafter referred to as SVC) standard is a multi-layer-coded video codec based on H.264/AVC. The latter has been the dominator in Internet video streaming. Although there is no open-source real time decoder for SVC, Polycom has issued a product for SVC-based video conferencing [4], which indicates that SVC is a practical solution for video streaming. SVC video stream has a mandatory base layer, which has the lowest temporal, spatial and quality representation of the video stream, and several optional enhancement layers that can increase video frame rate, resolution or picture quality. Therefore, it is natural to use SVC and enable the AP to multicast the mandatory base layer at the lowest bit rate supported by the client with the worst channel, and multicast optional enhancement layers at higher bit rates supported by those clients with better channel qualities.There have been a number of solutions [5]–[7] proposed for SVC video multicasting in Wi-Fi networks. However, these solutions need to modify 802.11 MAC protocol, making them impractical to be deployed in existing Wi-Fi networks.

Pseudo-broadcast [8] requires no change to the AP or the 802.11 MAC protocol, thus is deployable. Specifically, the multicast sender explicitly selects a wireless client as the unicast receiver, and other wireless clients listen for the packets opportunistically in the promiscuous mode. Compared with the legacy Wi-Fi multicast, pseudo-broadcast can benefit from Wi-Fi unicast's rate adaptation and Distributed Coordination Function (DCF) mechanisms. Solutions such as [9] use pseudo-broadcast for video multicasting over Wi-Fi networks, and the client with the worst channel quality is selected as the unicast receiver to increase the probability with which other clients successfully receive the packets. Compared to rate-limited and unreliable legacy Wi-Fi multicast, pseudo-broadcast is more reliable (due to unicast MAC ACK and retransmission) and can in theory achieve higher transmission rate (no rate-limiting). However, the transmission rate is still limited by the worst channel quality among all the clients.

In summary, existing approaches suffer from either *being impractical to deploy* and *low overall video quality limited by the worst channel* in existing multi-rate Wi-Fi networks. In this paper, we present M^3, a practical and reliable **m**ulti-layer video **m**ulticast solution over **m**ulti-rate Wi-Fi networks. In order to be practical to deploy in existing Wi-Fi networks, we constraint M^3 design so that it cannot modify or control APs and it cannot modify 802.11 MAC protocol. Our design goal is then to maximize the overall video quality received by all clients, while all clients can smoothly watch the video at least with the lowest quality.

M^3's core idea is to take advantage of the multi-rate nature of Wi-Fi unicast and make it work with SVC's multi-layer-coded video codec. Instead of selecting the client with the worst channel quality as the *single* receiver, M^3 would select *multiple* clients as the pseudo-broadcast receivers, therefore clients are not limited by the worst channel quality any more. Each receiver has a different channel quality, is automatically assigned a different PHY rate by the AP according to 802.11 protocol, and each serves a set of listener clients with better channel qualities. M^3 uses a proxy server called M^3 server to fetch SVC DASH chunks. Based on the latest network conditions, the M^3 server dynamically allocates and sends the suitable SVC layers for a receiver and the listeners it serves.

The design challenges faced by M^3 and our solutions are as follows. First, *how to improve pseudo-broadcast' reliability?* Pseudo-broadcast alone cannot guarantee the reliability for each client, especially for those *listeners*. M^3 uses a combined application-layer Forward Error Correction (FEC) and Automatic Repeat-reQuest (ARQ) mechanism to reduce the data loss in Pseudo-broadcast.

Second, *how to choose receivers and assign SVC layers to maximize overall video quality with limited bandwidth in an 802.11 LAN?* M^3 estimates the received signal strength indicator (RSSI) value and goodput of each client, and uses these two metrics as the inputs to the formulated optimization problem. Then M^3 uses binary integer linear programming to derive an optimal SVC-layer allocation strategy to maximize the total video quality.

Third, *how to adapt to the wireless dynamics in practice?* During the dynamic network conditions, it is challenging to select the appropriate clients as the receivers and allocate different SVC layers to them to fully utilize the available bandwidth in real time. To overcome this challenge, we use a periodical feedback mechanism to collect some necessary statistics from the clients, such as the RSSI of each client, the goodput of each selected receiver and the packet delivery ratio (PDR) of each client versus each selected receiver, *etc*. Using these statistics, M^3 server can adjust the SVC-layer allocation to meet the variation of network conditions.

The major contributions of our paper are summarized as follows. 1) We propose a practical and reliable video multicast approach that can be deployed today. 2) To the best of our knowledge, M^3 is the first multi-layer video multicast approach via *pseudo-broadcast with multiple receivers*. 3) We propose an effective algorithm that can dynamically adjust to the dynamics of the WiFi network. 4) We built a prototype of the M^3 system and evaluated it in a testbed consisting of 8 clients and one commercial off-the-shelf (CoTS) wireless AP. Our results show that M^3 can improve the total received video rate by up to 200%. M^3 does not excessively consume bandwidth, and fairly share bandwidth with other applications.

II. M^3 System Design

In this section, we describe a detailed M^3 system design. First, we introduce some practical constraints to the M^3 system.

A. Constraints

C1 : Wireless access points (APs) should be off-the-shelf and cannot be modified or controlled by M^3.

C2 : M^3 should be TCP-friendly.

Constraint **C1** is critical when it comes to easy deployment and cost-efficiency. For already deployed Wi-Fi networks in large-scale public places, it is expensive to replace APs. And updating AP's software or firmware is also risky and hardly allowed by administrators. Constraint **C2** is critical when it comes to the fairness in practical application scenario. In real public Wi-Fi networks, there must be different users and applications via the same APs at the same time. M^3 should share available bandwidth with other applications fairly.

B. Overview

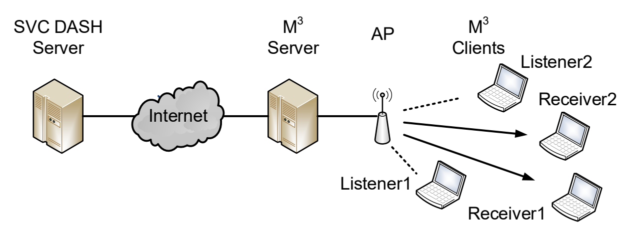


Fig. 1. M^3 System Architecture

As illustrated in Fig. 1, the M^3 system consists of one M^3 server, multiple M^3 clients and a SVC DASH server, which provides encoded SVC DASH chunks. Our focus is not the transmission of SVC video stream over the wired network, thus we assume that the wired network is not the bottleneck.

The M^3 server plays two roles: one is the SVC DASH client, fetching SVC DASH chunks from the SVC DASH server; another is the live video multicasting relay server, maintaining the membership of the multicast group and "*multicasting*" SVC video chunks to M^3 clients. Here, we use pseudo-broadcast to perform video multicasting. Therefore, there are two types of M^3 client: one is the M^3 receiver, selected by the M^3 server as the unicast receiver; another is the M^3 listener, receiving packets by monitoring the channel in the promiscuous mode. It is worth noting that one M^3 client can be both a M^3 receiver and a M^3 listener*.

Due to Wi-Fi's built-in multi-rate nature, we utilize SVC video to provide differentiated services for clients with different channel qualities. In particular, the M^3 server unicasts different SVC layers to different receivers with different channel qualities. Listeners can receive part of SVC layers according to the channel qualities of themselves.

Because of the variation of wireless environment and the different orders of clients' arrival, the M^3 server needs to periodically re-evaluate the selection of receivers and the resource allocation for SVC video layers (*e.g.* unicasting which SVC layer to which receiver, or FEC overhead); it also performs re-evaluation upon clients' arrival.

In the following, we will first introduce a multi-receiver pseudo-broadcast mechanism. Then we will present an optimized SVC-layer allocation for video multicasting over Wi-Fi network. Finally, we will describe how to adapt to the optimized target in dynamic network environment.

C. Multi-receiver Reliable Pseudo-broadcast

In pseudo-broadcast, a listener can receive packets with a certain probability which is related to the comparison between its channel quality and the channel quality of the selected receiver. 802.11 rate adaptation mechanism can select the transmission rate according to the receiver's channel quality. If the listener's channel quality is worse than the receiver's channel quality, this listener may not support the transmission rate allocated to that receiver and therefore may have a bad PDR. Here, we propose a definition called the receiver's *Service Set* (SS), consisting of the listeners which can receive the packets sent to that receiver with a high probability. For a listener in one receiver's SS, We say that the receiver can *cover* that listener.

The principles behind the multi-receiver reliable pseudo-broadcast are: 1) Selecting multiple receivers with different channel qualities; 2) The receiver R_1 with the worst channel quality can cover all listeners. However, R_1 has the lowest TCP throughput (TP), and the M^3 server at least sends the SVC base layer to R_1; 3) The receiver R_2 with the channel quality which is better than R_1's channel quality can cover less listeners than R_1. However, R_2 has a higher TP than R_1, and the M^3 server may send more SVC enhancement layers to R_2. Similarly, we may select more receivers (R_3, R_4 and so on). The set of receivers is determined by the number and bitrate of SVC layers and the available bandwidth of Wi-Fi network;

The main challenge of pseudo-broadcast-based SVC video multicast is, for each selected receiver, how to guarantee that all listeners in its SS can also successfully decode SVC layers sent to that receiver. Considering the structure of SVC DASH dataset, the video stream is chopped into chunks with the equal duration (*e.g.*, two seconds). For a specific SVC DASH chunk, it contains one base layer and multiple enhancement layers. One layer can be segmented into multiple packets for transmission purpose. To decode a specific layer in one SVC DASH chunk, all packets of this layer and other lower layers should be received. Since listeners in one receiver's SS may not receive all packets sent to that receiver, the M^3 server uses FEC mechanism to recover lost packets for listeners. In the following, we will first introduce some details of FEC code used by the M^3 system.

We use a powerful digital fountain code called RaptorQ [10], which is the most recent member of Raptor codes family, providing exceptional protection performance and enhanced encoding parameters. RaptorQ divides the whole input into **blocks**. Each block is *encoded and decoded independently* and will be divided into **symbols**. Each symbol must be transmitted separately in its own packet. Because we use TCP (Constraint **C2**: TCP-friendly) to transmit video stream, the symbol size cannot be larger than 1448 bytes. Due to the additional FEC header and the alignment limitation†, the symbol size is set to 1440 bytes. For a (n, k) block (containing k source packets and $(n-k)$ overhead packets), if the client receives more than k packets (source or overhead packets), k source packets can be successfully recovered.

In practice, a specific SVC layer in one SVC DASH chunk can be seen as one block. To guarantee that all listeners can receive this block, the M^3 server should tune the amount of overhead packets to be more than the highest packet loss in one block of those listeners. When the M^3 server sends a specific SVC layer to one receiver, it knows the PDR in one block (denoted by $BPDR$) for each listener, then it can determine a proper threshold of $BPDR$ (denoted by $\overline{BPDR}$) to only keep listeners whose $BPDR$ is higher than $\overline{BPDR}$ in this receiver's SS.

The challenge of multi-receiver reliable pseudo-broadcast is how to determine the value of $\overline{BPDR}$ for each receiver. To overcome this challenge, we conduct a verification test. In our Wi-Fi testbed, RSSI values of clients vary in the range of (-46 $\sim$ -85) dBm. We divide this range into 8 equal subranges of 5dBm bin and respectively put one client into one subrange based on its RSSI. In each run of our test, the M^3 server selects each client as the receiver and sends 2000 packets via TCP, and listeners always listen for the packets sent to each receiver. The TCP payload size is 1448 bytes. We record the receiver's TP, and also record a bitmap of received packets in 2000 packets for each listener. In our synthetic SVC DASH dataset (will be later introduced in Section III-B), the smallest block size is 120 packets. Because $BPDR$ of a small block size can be easily affected by the bursty loss, we use the block size of 120 packets to calculate $BPDR$. To evaluate the stability, we record the minimum $BPDR$ for each listener.

*In the rest of the paper, we will omit "M^3" before client, receiver and listener unless specified otherwise.

†The alignment size is 4 bytes, and the symbol size must be a multiple of the alignment size.

We randomly set 5 different layouts of 8 clients. Each layout guarantees that each client is located in different RSSI value subranges, and we repeat the verification test of each layout for 20 times.

TABLE I
MEAN VALUES AND STANDARD DEVIATIONS OF TP FOR RECEIVERS

RSSI|dBm	-50	-55	-60	-65	-70	-75	-80	-85
AVG_DR|KBps	6136	6062	5989	6017	4677	3096	1989	633
STD_DR|KBps	345	474	447	394	217	300	132	93

Table I lists the mean values and standard deviations of TP for all receivers in different RSSI value subranges. When RSSI is higher than -65dBm, the TP is close to 6000KBps. When RSSI is lower than -65dBm, the TP is getting smaller until it reaches approximately 600KBps. For different tests, the TP is relatively stable and the maximum standard deviation among all receivers is not higher than 15%.

TABLE II
MEAN VALUES OF MINIMUM $BPDR$ FOR CLIENTS

AVG_MIN	-50	-55	-60	-65	-70	-75	-80	-85
-50	100	82.2	67.8	56.2	0	0	0	0
-55	81.2	100	78.1	65.3	0	0	0	0
-60	86.5	91.4	100	81.8	0	0	0	0
-65	85.7	94.3	71.9	100	0	0	0	0
-70	95	98.8	90.2	96.1	100	0	0	0
-75	98	95.8	96.8	98.8	94	100	3.9	0
-80	99.9	98.8	98	97.4	99.6	86.4	100	0
-85	100	99.6	98.1	100	99	99.4	93.5	100

TABLE III
STANDARD DEVIATIONS OF MINIMUM $BPDR$ FOR CLIENTS

STD_MIN	-50	-55	-60	-65	-70	-75	-80	-85
-50	0	22.3	27.4	26.5	0	0	0	0
-55	21.4	0	19.4	23.4	0	0	0	0
-60	17.1	9.3	0	17.9	0	0	0	0
-65	18.3	11.5	32.2	0	0	0	0	0
-70	7.7	3.2	18.9	8	0	0	0	0
-75	7.6	15.2	4.2	4.9	14	0	5.2	0
-80	0.5	3.6	4.4	3.1	1.1	17.1	0	0
-85	0.2	1.8	4	0	0.9	0.8	7.4	0

Table II and III list the mean values and standard deviations of $BPDR$ (%) for all clients. The first column lists the RSSI value subranges (dBm) in which the receivers are located and the first row lists the RSSI value subranges in which the receiver itself and other listeners are located. Because we have one client in each subrange, the diagonal in Table II lists $BPDR$ of each receiver and the values are all 100%. In Table II, all values below this diagonal are higher than 70%. Therefore, when RSSI values of listeners are higher than the RSSI value of the receiver, the receiver can cover those listeners with a high probability. However, as illustrated in Table III, $BPDR$ for those listeners are not all very stable (when the standard deviation of minimum $BPDR$ is higher than 15%) especially when the RSSI value of the receiver is higher than -65dBm. In other words, when located in the RSSI value range of (-46 $\sim$ -65) dBm, the receiver may not cover other listeners very stably.

From the above results, we propose a RSSI-based SS hierarchy. Before formally describing the hierarchy problem, we first provide some symbols used in our formulation in Table IV.

TABLE IV
SYMBOLS FOR THE FORMULATION OF HIERARCHY PROBLEM

N	The number of M^3 clients
h	The Level $h (1 \leq h \leq N)$ in the hierarchy
R_h	The selected M^3 receiver in Level h
SS_h	The service set of Level h
$\overline{BPDR}_h$	The threshold of $BPDR$ for SS_h
TP_h	The TP of R_h
$RSSI_h$	The RSSI value of R_h
O_h	FEC overhead ratio needed to recover lost packets in one block for SS_h. $O_h = (1 - \overline{BPDR}_h)/\overline{BPDR}_h$
GP_h	The goodput of R_h. $GP_h = TP_h/(1 + O_h)$

Our goal is to find a hierarchy consisting of multiple SSs. There are two constraints: (1) SS_1 contains all M^3 clients; (2) For Level i and Level j, if $i < j$, then $SS_i \supset SS_j$ and $GP_i < GP_j$. The formalized hierarchy is presented in Table V.

TABLE V
THE FORMALIZED HIERARCHY OF M^3 CLIENTS

Level 1: $R_1 \rightarrow (RSSI_1, SS_1, TP_1, \overline{BPDR}_1, O_1, GP_1)$
...
Level h: $R_h \rightarrow (RSSI_h, SS_h, TP_h, \overline{BPDR}_h, O_h, GP_h)$
...

Specifically, we assume that one receiver's SS contains all listeners whose RSSI values are higher than the RSSI value of that receiver. To build the hierarchy, the M^3 server needs to first get RSSI values and TP of all clients, and then determine $\overline{BPDR}$ for each SS. Based on the statistics in Table II, we limit the minimum $\overline{BPDR}$ to 70%. If the lowest $BPDR$ in one SS is higher than 70%, $\overline{BPDR}$ is set to that lowest $BDPR$. Otherwise, $\overline{BPDR}$ is set to 70%. For each SS_h, we can use TP_h and $\overline{BPDR}$ to calculate O_h and GP_h, then all feasible hierarchies can be derived by a depth first search (DFS) algorithm.

Considering the instability of pseudo-broadcast, we use a per-block ARQ mechanism to further improve the reliability of video transmission. For one listener, if the amount of received packets in one block are not enough for successfully decoding and more than 70% of the block size, it would send an ACK to the M^3 server to request the short amount of packets for this block. The M^3 server will unicast additional FEC overhead packets to that listener, which potentially avoids repeated retransmissions if the M^3 server still unicasts FEC overhead packets to the receiver. To limit the overhead of retransmission, in our M^3 system, we just use this ARQ mechanism to protect SVC base layer.

D. Optimized SVC-layer Allocation

For a specific hierarchy, the M^3 server must send SVC base layer to R_1 to guarantee that all clients can play the video with the lowest quality. Besides, the M^3 server can send other SVC enhancement layers to R_1 or other receivers. The goal of SVC-layer allocation is to maximize the total video rate (denoted by VR) received by all clients. In the following, we will describe the allocation problem formally.

We use $l(1 \leq l \leq L)$ to denote a SVC layer, and the video rate of layer l is VR_l. The formulation of clients hierarchy is described as above. Because one SVC layer is at most sent to one receiver, we give a limit to the number of levels ($H \leq L$) in the hierarchy. The problem is to find the optimal map $\{m^1, m^2, \cdots, m^l, \cdots, m^L | m^l \in (1, 2, \cdots, h, \cdots, H)\}$ between L SVC layers and H hierarchy levels that will maximize $\mathbb{U}$, sum of VR for all clients. $m^l = h$ means that the M^3 server sends SVC layer l to R_h. Note that some SVC layers can be transmitted to the same receiver and also some layers may not be transmitted ($m^l = 0$).

This optimization problem can be solved by binary integer linear programming (BILP) [11]. We first define an allocation matrix

$$\Lambda = \begin{bmatrix} \lambda_{1,1} & \cdots & & \\ \vdots & \ddots & & \\ & \lambda_{h,l} & & \\ & & \ddots & \vdots \\ & & \cdots & \lambda_{H,L} \end{bmatrix} \tag{1}$$

$\lambda_{h,l} = 1$ means that the M^3 server sends SVC layer l to R_h and $\lambda_{h,l} = 0$ otherwise.

For hierarchy level h, we can get effective goodput for video transmission

$$GP_h = TP_h/(1 + O_h) \tag{2}$$

If the M^3 server sends SVC layer l to R_h, then the channel occupancy ratio is

$$\rho_{h,l} = VR_l/GP_h \tag{3}$$

When transmitting multiple SVC layers, the sum of channel occupancy ratio should be less than 1‡. Thus, the optimization problem can be stated as follows:

The objective is to maximize

$$\mathbb{U} = \sum_{h=1}^{H} |SS_h| \cdot \sum_{l=1}^{L} \lambda_{h,l} \cdot VR_l \tag{4}$$

The constraints are

$$\sum_{h=1}^{H} \sum_{l=1}^{L} \lambda_{h,l} \rho_{h,l} \leq 1 \tag{5}$$

$$\lambda_{h,l} - \sum_{k=1}^{h} \lambda_{k,l-1} \leq 0, \forall h \in H, 2 \leq l \leq L \tag{6}$$

$$\sum_{h=1}^{H} \lambda_{h,l} \leq 1, \forall l \in L \tag{7}$$

$$\sum_{l=1}^{L} \lambda_{1,l} \geq 1 \tag{8}$$

The constraints are described in the following: constraint (5) guarantees smooth playback; constraint (6) ensures that a layer l is sent only if the layer $l-1$ has been sent in a same or lower level; constraint (7) ensures that any layer can be sent only once; constraint (8) ensures that all clients can at least receive the base layer.

The above BILP can be solved using GLPK (GNU Linear Programming Kit). For each feasible hierarchy, we calculate an optimal allocation matrix Λ and a maximum $\mathbb{U}$. Then we can choose the maximum $\mathbb{U}$ and get the optimal SVC-layer allocation.

E. Periodical Client Feedback

As described in the Section II-C, to build the hierarchy, the M^3 server should get TP and RSSI value of each client and $BPDR$ of each listener corresponding to each receiver. In practice, before providing live video service, not all clients have joined the multicast group. The M^3 server has indeed no chance to perform a complete test to get TP and $BPDR$ of all clients. Whenever system state or network condition changes, the M^3 server has just a little time to re-evaluate system settings to approach the optimized target. In fact, the real application scenario is very complicated. For the M^3 system, we mainly consider the following two cases: 1) One client successively joins the multicast group every once in a while. Whenever a new client arrives, the M^3 server may make necessary adjustments to SVC-layer allocation. 2) Whenever the bitrate of background traffic fluctuates, the M^3 server would adjust SVC-layer allocation to adapt to the available bandwidth.

We design a step-by-step method to adaptively adjust SVC-layer allocation for the process of clients joining the multicast group and a periodical client feedback mechanism to adapt to the changing network conditions.

- The first client arrives.
  - The M^3 server selects the first client as R_1 and begins to send all SVC layers.
  - The client sends a regular feedback containing average $RSSI$, average TP and $BPDR$ of each block to the M^3 server every 10 seconds.
  - The M^3 server re-allocates SVC layers whenever receiving a regular feedback.
- Other clients arrive.
  - The newly-arriving client first sends its $RSSI$ to the M^3 server. If this client has the lowest $RSSI$ in all current clients, the M^3 server immediately sends the "test chunk". Otherwise, the M^3 server waits for the first regular feedback from this client, and then sends the test chunk.
  - For the next video chunk, the M^3 server sends base layer to the current R_1, and sends all other enhancement layers to this newly-arriving client. This special

‡The measurement of GP_h should consider the background traffic

video chunk is referred to as "test chunk". After sending this test chunk, the M^3 server respectively unicasts a *test ending signal* to each client.

  - Once having received the test ending signal, the client immediately returns a *test feedback*. Then the client resets the regular feedback timer and still sends a regular feedback every 10 seconds.
  - From these test feedbacks, the M^3 server can update the TP of the newly-arriving client and $BPDR$ of other listeners corresponding to this new client. Once having received all test feedbacks, the M^3 server re-allocates SVC layers.
- The set of clients is stable.
  - Each client returns a regular feedback every 10 seconds. The M^3 server updates $RSSI$ of each client, TP of each receiver and $BPDR$ of each listener corresponding to each current receiver.
  - Once having received all regular feedbacks, the M^3 server re-allocates SVC layers.

III. Performance Evaluation

In this section, we first define several performance metrics and describe the implementation and experiment setups. Then we evaluate the system stability when clients successively join the multicast group and the background traffic fluctuates. At last, we compare the improvement of total video quality for different client layouts against single-layer video multicast.

A. Performance Metrics

As described in literature [12] and [13], QoE of the video streaming is better measured by average video rate, video rate variation, buffer ratio *etc.*, because they are more relevant to user viewing experience than traditional PSNR metric. Based on this, we define the following metrics to characterize the performance of SVC video multicast over Wi-Fi network:

- **Buffering Time**: Total buffering time by a client, starting from when the client joins the multicast group, and including initial buffering and later re-buffering. In all tests, the initial buffering length and re-buffering length are all 6 seconds, which means that the player must buffer 6-second-long video chunks before resuming the video.
- **Total Video Rate**: Because our design goal is to optimize the overall system performance, we record total video rate of all current clients in the multicast group.
- **Skipped Chunks**: If not enough packets are received to decode SVC base layer of one video chunk and the amount of received packets do not meet the threshold of retransmission, the client will skip this video chunk. We record the number of skipped chunks for each client after successfully decoding its first video chunk.

B. Implementation and Experiment Setups

We have built a prototype implementation of M^3 System. The SVC DASH server and the M^3 server are built on two Ubuntu PCs. The M^3 client is built on Raspberry Pi, which runs Raspbian, a free operating system based on Debian. We use a CoTS TP-LINK wireless router as the AP, which has an 802.11a/n wireless adaptor running at 5GHz channel 40 with 20MHz frequency span. This channel is a non-overlapping channel and there is no other wireless AP to use this channel in our environment. The available unicast bandwidth ranges from 6.5Mbps to 144.4Mbps.

The M^3 server software is implemented as a user-level application. It first fetches video chunks from the SVC DASH server using HTTP, then segments each video chunk into multiple FEC source symbols with the size of 1440 bytes. The M^3 server does not perform real FEC encoding, it just produces fake overhead symbols with the same size as source symbols and adds an 8-byte header to source and overhead symbols, then unicasts them to selected receivers. The M^3 client software consists of two components: a packet filter and a simulated SVC video player. The client's wireless adaptor runs in the promiscuous mode, and the packet filter identifies FEC packets using the source IP and the TCP port number, then forwards these packets to the simulated player. When joining the multicast group, the simulated player sets up a TCP connection with the M^3 server to transmit feedbacks and retransmission requests. If the client is selected as the receiver, the M^3 server also uses this connection to send FEC packets. The simulated player acts as a video player with the exception of performing real decoding. It records a timeline of decodable SVC layers for each video chunk and buffer duration because the hardware of Raspberry Pi is not capable enough for real decoding.

A synthetic SVC DASH dataset with constant bitrate is used. In order to ensure the rationality of the synthetic dataset, we learn from the bitrate allocation of a 5-layer SVC dataset given by [14]. The size of each layer in a two-second-duration video chunk is (120, 300, 500, 350, 800) packets (the packet size is 1440 bytes). The corresponding VR of each layer is (84, 211, 352, 246, 563) KBps. Because our focus is the transmission protocol for reliable SVC video multicasting over Wi-Fi network, we believe these simplifications for FEC and SVC do not impact the effectiveness of transmission mechanisms and are acceptable.

To compare with single-layer video multicast, we also use 5 corresponding synthetic AVC DASH datasets with constant bitrate. Because SVC encoder introduces 10% overhead of bitrate for each enhancement layer [14], if we use 5 different AVC DASH datasets whose video quality is the same as that of SVC DASH dataset, the size of each AVC DASH chunk is (120, 382, 767, 977, 1479) packets. The video rate of each AVC video stream is (84, 269, 539, 687, 1040) KBps.

To avoid the impact of hidden terminals, we place 8 clients in a 90-degree quadrant area around the AP. The RSSI values of clients are in the range from -46dBm to -85dBm. For the sake of simplicity, in each layout used in our experiments, we number the clients in the order of their RSSI values, C_1 has the highest RSSI value, and C_8 has the lowest RSSI value.

C. System Stability When Clients Successively Arrive

As illustrated in Table II and III, receiver selection is the key factor to system stability. In this subsection, we consider the process of clients successively joining the multicast group to verify if our M^3 system can adapt well to newly-arrived clients.

TABLE VI
THE ARRIVAL ORDER OF CLIENTS

Case 1	C_1, C_2, C_3, C_4, C_5, C_6, C_7, C_8
Case 2	C_8, C_7, C_6, C_5, C_4, C_3, C_2, C_1
Case 3	C_4, C_1, C_6, C_2, C_8, C_5, C_3, C_7
Case 4	C_3, C_1, C_7, C_5, C_2, C_6, C_8, C_4

In order to consider the general situation, as we did in the previous verification test, we divide the RSSI value range into 8 subranges of 5dBm bin and put one client in each subrange. Besides, as listed in Table VI, we use 4 different client arrival orders. Case 1 and Case 2 are respectively in descending and ascending orders of RSSI values. Case 3 and Case 4 are two different random orders.

As described in Section II-E, whenever a new client arrives except for the first one, M^3 server will send a test video chunk. When this test is finished, M^3 server may need to adjust SVC-layer allocation. We would like to examine the impacts to performance metrics after a new client arrives and before the next client arrives. Besides, we also show the change of SVC-layer allocation in this duration.

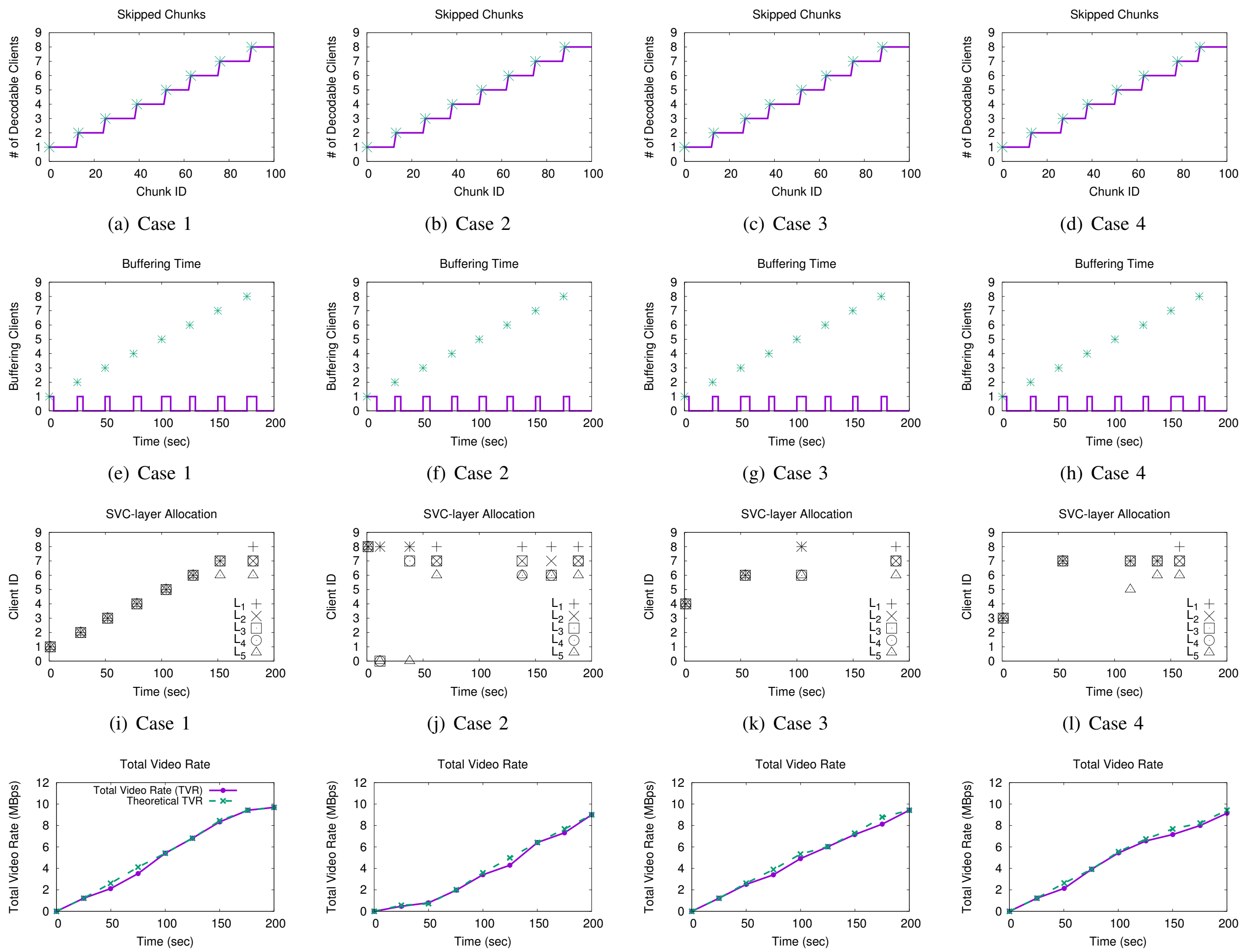
For live video streaming, the smoothness of video playback is the most important indicator of user experience. Users may be able to accept a lower video resolution, but it's hard for them to accept frequent buffering events and dropped frames [15]. We first observe skipped chunks and buffering time during the client arriving process. Note that these two metrics use different timelines. For skipped chunks in Fig. 2(a) to 2(d), x-axis represents all video chunks sent by the M^3 server from the first client arriving, y-axis represents the number of decodable clients for a specific video chunk. Here, a decodable client is the client who can decode the x-th video chunk. A star in the figure represents the first video chunk received by the newly-arriving client. For buffering time in Fig. 2(e) to 2(h), x-axis represents the time since the first client arriving, y-axis represents the number of buffering clients for a specific time. Here, a star represents the time of the new client arrival.

From the results of skipped chunks and buffering time, we can observe that system stability is not obviously impacted by different client arrival orders. As illustrated in Fig. 2(a) to 2(d), there is no skipped chunk for all clients after receiving their first video chunks, which means that the M^3 server can select proper receivers after each new client arriving and the combined FEC and ARQ mechanism can guarantee reliable transmission of SVC base layer for all clients. As illustrated in Fig. 2(e) to 2(h), for any time after the first client arrives, there is at most one client for buffering the video chunks. These buffer durations are the initial buffer durations for the newly-arriving clients. There is no re-buffering event occurring for all clients, which means that the M^3 server can provide smooth video streaming for all current clients whenever the new client arrives.

Combining the variation of SVC-layer allocation showed in Fig. 2(i) to 2(l), we will explain how the M^3 server adapts to different client arrival orders. Here, x-axis represents the time starting from the arrival of the first client, and y-axis represents the selected receiver to which each SVC layer is allocated whenever SVC-layer allocation changes. When the client ID is 0 this means that the corresponding SVC layers are not transmitted due to the bandwidth limitation. For Case 1, every newly-arriving client has the lowest RSSI value among the current clients in the multicast group. Therefore, whenever the new client arrives, the M^3 server will immediately send the test chunk. And after the testing process is finished for each newly-arriving client, the M^3 server will adjust SVC-layer allocation as illustrated in Fig. 2(i). In this situation, the newly-arriving client may not receive any video chunk before the testing process is finished, which leads to a longer initial buffering time. And a lower RSSI value may result in a longer testing time, *e.g.* C_8 in Case 1. For Case 2, the RSSI value of each newly-arriving client is higher than the RSSI value of the current R_1. Therefore, the newly-arriving client can be covered by the current R_1, and the initial buffering time is not extended by the testing process. After receiving the first regular feedback from the new client, the M^3 will start the testing process to re-evaluate the SVC-layer allocation. In this situation, the SVC-layer allocation may not be changed if it is still optimal for the new client set as illustrated in Fig. 2(j). Both processing methods are present in Case 3 and 4 as illustrated in Fig. 2(k) and 2(l).

The results in Fig. 2(a) to 2(l) show that, when a new client arrives, the M^3 server can adjust the SVC-layer allocation to provide the reliability for all current clients. Due to different orders of client arrival in these 4 cases, the variations of SVC-layer allocation show different trends. However, after all clients have arrived, the last SVC-layer allocation patterns are all the same. This means that our SVC-layer allocation algorithm can converge to the same result regardless of the clients arrival order.

Combining the results of skipped chunks and buffering time with the RSSI-based SS, we can utilize SVC-layer allocation patterns to calculate theoretical total video rate illustrated by the dash lines in Fig. 2(m) to 2(p). The solid lines represent the measured total video rate. We can observe that these two curves fit very well most of the time, which verifies the effectiveness of RSSI-based SS. For those periods in which the theoretical value is higher than the measured value, it is because the selected receivers cannot perfectly cover all listeners of which RSSI values are higher than the receivers' with only FEC mechanism due to the instability of pseudo-broadcast. For these 4 different cases, after all clients arriving, the total video rates are nearly the same, which also means that the M^3 server can select proper receivers according to the specific client layout.



(a) Case 1 (b) Case 2 (c) Case 3 (d) Case 4

(e) Case 1 (f) Case 2 (g) Case 3 (h) Case 4

(i) Case 1 (j) Case 2 (k) Case 3 (l) Case 4

(m) Case 1 (n) Case 2 (o) Case 3 (p) Case 4

Fig. 2. System stability in the clients arriving process

D. The Impact of Background Traffic

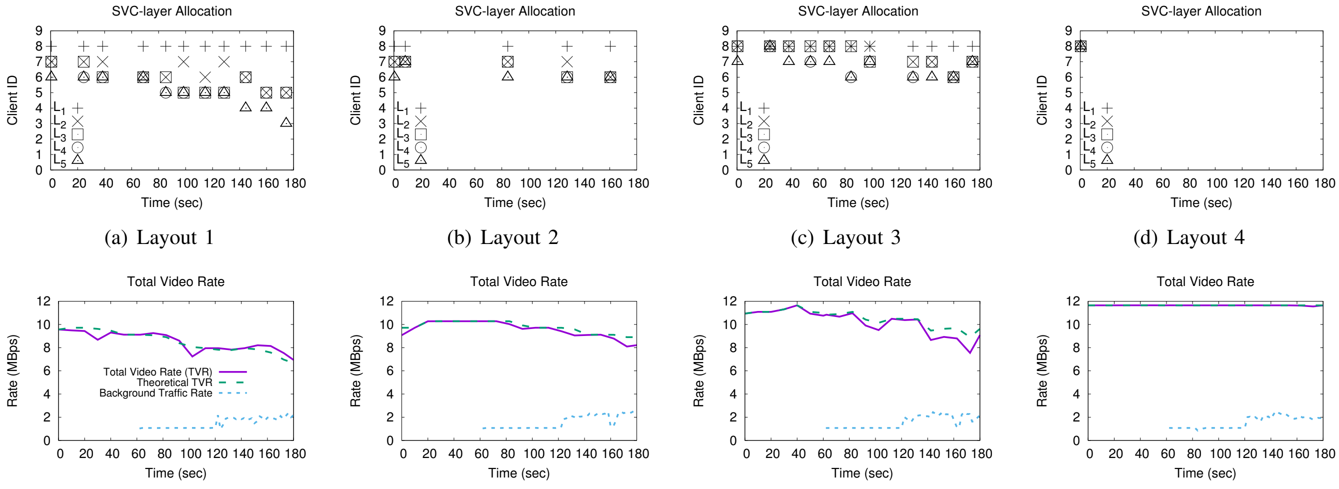
In this subsection, we consider system stability impacted by background traffic. We divide the experiment process into three phases: 1. After all clients have joined the multicast group and the new client test has been finished for the last client, M^3 system runs for 60 seconds with no background traffic; 2. A background traffic server which is connected to the AP via the wired network sends TCP packets to a background traffic client which is connected to the AP via Wi-Fi. RSSI value of this client is about -60dBm. The server sends a TCP packet with the payload size of 1448 bytes every 1200 microseconds for 60 seconds; 3. The same as phase 2 except that the time interval of sending a packet is set to 50 microseconds.

In order to evaluate system stability impacted by background traffic, we consider 4 different client layouts with different available bandwidths listed in Table. VII. The main differences between these layouts are RSSI values of those clients with the worst channel qualities, which largely determine the available bandwidth of M^3 system.

TABLE VII
CLIENT LAYOUTS

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Layout 1 (dBm)	-50	-55	-60	-65	-70	-75	-80	-85
Layout 2 (dBm)	-50	-55	-60	-60	-65	-70	-75	-85
Layout 3 (dBm)	-50	-55	-60	-60	-65	-70	-75	-80
Layout 4 (dBm)	-50	-55	-55	-60	-60	-65	-70	-75

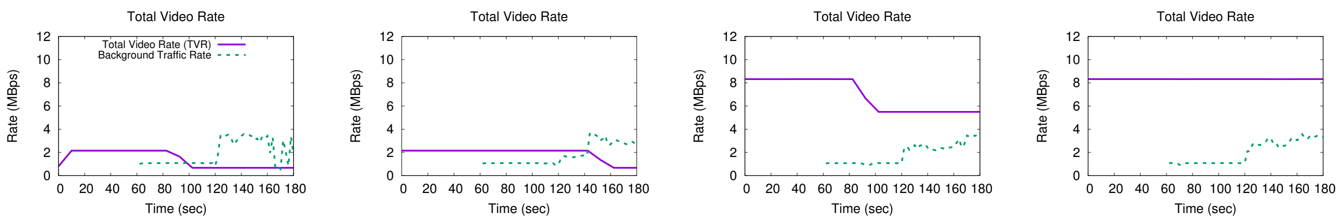
For all these tests, we observe no skipped chunk and buffering event for any client, which means that M^3 can provide a good reliability for SVC base layer and can adjust SVC-layer allocation according to the available bandwidth. Therefore, we only present variations of SVC-layer allocation and total video rate in Fig. 3. In Fig. 3(e) to 3(h), the solid lines



(a) Layout 1 (b) Layout 2 (c) Layout 3 (d) Layout 4

(e) Layout 1 (f) Layout 2 (g) Layout 3 (h) Layout 4

Fig. 3. System stability when background traffic fluctuates



(a) Layout 1 (b) Layout 2 (c) Layout 3 (d) Layout 4

Fig. 4. Total video rate of single-layer video multicast

and dash lines represent the same meanings as described in Fig. 2(m). The dotted lines represent the bitrate of background traffic. For Layout 1 $\sim$ 3, with the arrival and increasing intensity of background traffic, the total video rate gradually decreases. Especially for Layout 1 and 3, there are frequent fluctuations for SVC-layer allocation. Whenever the measured GP of selected receivers decreases and cannot support current SVC-layer allocation, the M^3 server would re-calculate SVC-layer allocation according to the existing measured GP. When a receiver is changed, the M^3 server needs to re-evaluate the GP of this receiver. If the new measured GP still cannot support the current SVC-layer allocation, this receiver would be changed again. However, for Layout 4, the GP of C_8 can still support all SVC layers even in the third phase. Therefore, SVC-layer allocation and the total video rate are always stable.

E. Comparison with Single-layer Video Multicast

Single-layer video multicast can be seen as a special case of multi-layer video multicast to transmit just one video layer. We implement this single-layer video multicast based on our M^3 system. The M^3 server selects the client with the lowest RSSI value as the receiver and transmits a single-layer-code video stream with proper bitrate selected from 5 available options described in Subsection III-B according to the GP of the receiver. We also use the same client layouts listed in Table VII to compare single-layer video multicast with our M^3 system. Our single-layer video multicast scheme is also a pseudo-broadcast-based video multicast solution, which can be seen as an enhanced Dircast version [9] with combined FEC and ARQ mechanisms.

As illustrated in Fig. 4, we can observe that the total video rate is completely limited by the GP of C_8. Although the results of all layouts are stable, the total video rate of single-layer video multicast is much lower than M^3's result when C_8 has a very low RSSI value. Specifically for Layout 2 with no background traffic, M^3 has a total video rate of over 10MBps. Considering the coding overhead of 40% for 4 enhancement layers, M^3 still improves the total video rate by over 200% compared with the single-layer video multicast. When C_8 has quite adequate bandwidth, these two approaches tend to have the same received video quality *e.g.* in Layout 4.

IV. Related Work

Reliability for multicast can be achieved by two ways: automatic repeat request (ARQ) and forward error correction (FEC). Nearly all of the solutions use one of them or both [16]. According to different feedback manners, multicast ARQ mechanisms can be classified into three main categories. All of them have some limitations: individual ACK mechanisms [17] may incur feedback implosion with large multicast group; leader-based ACK mechanisms [18]–[20] just provide the reliability to those leaders; leader-based NACK mechanisms [5], [21], [22] may ignore NACKs due to the capture effect. Pseudo-broadcast [8] is a special leader-based ACK mechanism with only one receiver to transmit ACKs. Compared with ARQ, FEC enables receivers to recover losses without contacting the sender. Therefore, FEC is more suitable for delay constrained applications. Packet-level FEC is widely used for streaming applications [19], [21], [23]. In contrast, M^3 propose a multi-receiver reliable pseudo-broadcast, which selects multiple receivers to transmit SVC video stream. To improve the reliability, M^3 mainly uses FEC mechanism to protect video stream, and uses an application-layer block-NACK mechanism to respectively recover lost packets for each client via unicast.

SVC video streaming can be used to reduce the video distortion because of its multi-layer coding. In order to guarantee reliable SVC video multicasting over Wi-Fi, an efficient rate adaptation mechanism for SVC video layers should be specified. All of these rate adaptation mechanisms [5]–[7] create a correlation between Modulation and Coding Scheme (MCS) indexes and used metrics. However, adjusting MCS index for each transmitted packet needs to modify AP's wireless driver. In contrast, M^3 utilizes legacy rate adaptation mechanism of 802.11 unicast, and selects different unicast receivers to provide multi-level reliability for SVC layers instead of changing MCS index.

V. Conclusions

In this paper, we present the M^3 system for practical and reliable multi-layer video multicasting over multi-rate Wi-Fi network. To the best of our knowledge, this is the first multi-layer multicast approach via pseudo-broadcast with multiple receivers under the same AP. Unlike other previous schemes for SVC video multicasting over Wi-Fi network, M^3 does not make any changes to the AP's 802.11 driver and can be deployed in existing Wi-Fi networks. M^3 uses a combined FEC and ARQ mechanism to improve multicast reliability and uses a periodical feedback mechanism to adapt to the dynamics of the Wi-Fi network. Using a real testbed, we have demonstrated that M^3 can improve the total received video rate by up to 200% compared with traditional single-layer video multicast. In our future work, we plan to evaluate the impact of client mobility on M^3 and further demystify the instability of pseudo-broadcast.

Acknowledgement

This work was partly supported by the State Key Program of National Science of China under grant 61233007, the National Natural Science Foundation of China (NSFC) under grant 61472214 & 61472210, the National High Technology Development Program of China (863 program) under grant 2013AA013302, the National Key Basic Research Program of China (973 program) under grant 2013CB329105, the Tsinghua National Laboratory for Information Science and Technology key projects, the Global Talent Recruitment (Youth) Program, and the Cross-disciplinary Collaborative Teams Program for Science & Technology & Innovation of Chinese Academy of Sciences-Network and system technologies for security monitoring and information interaction in smart grid.

References

[1] "Cisco visual networking index: forecast and methodology, 2013-2018." [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/ip-ngn-ip-next-generation-network/white_paper_c11-481360.pdf

[2] T. Stockhammer, "Dynamic adaptive streaming over HTTP – standards and design principles," in *ACM conference on Multimedia systems*, 2011.

[3] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE TCSVT*, 2007.

[4] "RealPresence SVC-based Conferencing Solutions." [Online]. Available: http://support.polycom.com/global/documents/support/strategic_partner_solutions/realpresence_svc_based_conferencing_solutions_deployment.pdf

[5] W.-S. Lim, D.-W. Kim, and Y.-J. Suh, "Design of efficient multicast protocol for IEEE 802.11 n WLANs and cross-layer optimization for scalable video streaming," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 5, pp. 780–792, 2012.

[6] Y. P. Fallah, H. Mansour, S. Khan, P. Nasiopoulos, and H. M. Alnuweiri, "A link adaptation scheme for efficient transmission of H.264 scalable video over multirate WLANs," *IEEE TCSVT*, vol. 18, no. 7, pp. 875–887, 2008.

[7] W.-H. Kuo, R. Kaliski, and H.-Y. Wei, "A QoE Based Link Adaptation Scheme for H.264/SVC Video Multicast over IEEE 802.11," *IEEE TCSVT*, vol. 25, no. 5, 2015.

[8] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: practical wireless network coding," *IEEE/ACM Transactions on Networking (ToN)*, vol. 16, no. 3, pp. 497–510, 2008.

[9] R. Chandra, S. Karanth, T. Moscibroda, V. Navda, J. Padhye, R. Ramjee, and L. Ravindranath, "DirCast: a practical and efficient Wi-Fi multicast system," in *IEEE ICNP*, 2009.

[10] M. Luby, A. Shokrollahi, M. Watson, T. Stockhammer, and L. Minder, "RaptorQ Forward Error Correction Scheme for Object Delivery," RFC 6330, 2011.

[11] K. Deep Singh, K. Piamrat, H. Park, C. Viho, and J.-M. Bonnin, "Optimising QoE for scalable video multicast over WLAN," in *IEEE Personal Indoor and Mobile Radio Communications (PIMRC)*, 2013.

[12] V. Joseph and G. de Veciana, "Nova: Qoe-driven optimization of dash-based video delivery in networks," in *IEEE INFOCOM*, 2014, pp. 82–90.

[13] X. Yin, A. Jindal, V. Sekar, and B. Sinopoli, "A control-theoretic approach for dynamic adaptive video streaming over http," in *ACM SIGCOMM*, 2015, pp. 325–338.

[14] C. Kreuzberger, D. Posch, and H. Hellwagner, "A scalable video coding dataset and toolchain for dynamic adaptive streaming over HTTP," in *ACM Conference on Multimedia Systems*, 2015.

[15] A. Balachandran, V. Sekar, A. Akella, S. Seshan, I. Stoica, and H. Zhang, "Developing a predictive model of quality of experience for internet video," in *ACM SIGCOMM*, 2013, pp. 339–350.

[16] J. Vella and S. Zammit, "A survey of multicasting over wireless access networks," *IEEE Communications Surveys & Tutorials*, 2013.

[17] "IEEE 802.11aa standard, Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Amendment 2: MAC enhancements for robust audio video streaming," 2012.

[18] M.-t. Sun, L. Huang, S. Wang, A. Arora, and T.-H. Lai, "Reliable MAC layer multicast in IEEE 802.11 wireless networks," in *IEEE ICPP*, 2002.

[19] O. Alay, T. Korakis, Y. Wang, and S. Panwar, "Dynamic rate and FEC adaptation for video multicast in multi-rate wireless networks," *Mobile Networks and Applications*, vol. 15, no. 3, pp. 425–434, 2010.

[20] X. Wang, L. Wang, Y. Wang, Y. Zhang, and A. Yamada, "Supporting MAC layer multicast in IEEE 802.11n: Issues and Solutions," in *IEEE WCNC*, 2009.

[21] Z. Li and T. Herfet, "MAC layer multicast error control for IPTV in wireless LANs," *Broadcasting, IEEE Transactions on*, vol. 55, no. 2, pp. 353–362, 2009.

[22] A. Basalamah, H. Sugimoto, and T. Sato, "Rate adaptive reliable multicast MAC protocol for WLANs," in *IEEE VTC*, 2006.

[23] A. Majumda, D. G. Sachs, I. V. Kozintsev, K. Ramchandran, and M. M. Yeung, "Multicast and Unicast Real-time Video Streaming over Wireless LANs," *IEEE TCSVT*, vol. 12, no. 6, pp. 524–534, 2002.