SimCast: Efficient Video Delivery in MU-MIMO WLANs

Guanhua Wang*, Kaishun Wu*†‡, Qian Zhang* and Lionel M. Ni*§
*Department of Computer Science and Engineering
§Guangzhou HKUST Fok Ying Tung Research Institute
Hong Kong University of Science and Technology
†College of Computer Science and Software Engineering, Shenzhen University
‡Corresponding Author
{gwangab, kwinson, qianzh, ni}@cse.ust.hk

Abstract—Wireless video stream delivery is choppy. This problem becomes much severer in MU-MIMO's simultaneous video transmission within the same band. Conventional schemes achieve graceful video delivery by harnessing from high data redundancy. However, with concurrent transmission in the same band, the leverage of high data redundancy leads to high probability of collisions and packets loss, which limits the performance. In concurrent video transmission, achieving efficiency over varied link condition is the main issue. To address this issue, this paper presents <u>Sim</u>Cast (<u>Sim</u>ultaneous), a cross-layer design for achieving efficient concurrent video uploading/downloading in MU-MIMO WLANs. The key idea of SimCast is to harness frequency diversity of the channel and spatial similarity of users. We implemented SimCast on USRP2 and conducted extensive simulations. Result shows that SimCast achieves higher throughput than traditional schemes by $1.2\times$ on average. Video quality of SimCast outperforms competitive schemes, which is up to 5 dB in PSNR.

I. INTRODUCTION

Mobile video delivery is becoming the major workload in wireless traffic. According to Cisco visual index [1], the video-on-demand traffic load will be tripled by 2015. However, the quality of mobile video streaming is unsatisfactory because of unstable wireless links. This issue attracts lots of research interests which aim to improve the performance of mobile video delivery. State-of-the-art literatures propose several novel techniques in improving mobile video delivery [16] [7] [19]. The basic idea of these schemes is to achieve graceful video delivery by harnessing from high data redundancy.

Nowadays, MU-MIMO (Multi-User Multiple Input Multiple Output) system can support concurrent video uploading/downloading within the same band. However, nearly none of prior works can achieve acceptable performance for simultaneous video transmission in MU-MIMO WLANs (Wireless Local Area Networks). The main reason is that the high data redundancy can lead to high probability of collisions and packets loss, which limits the transmission performance.

How many times have you endured bad video quality experience on your mobile devices when people around you watching the same event, like the Super Bowl game? How often have you got stuck at uploading an interesting incident you shot while others also uploading nearly the same content? Nearly none of recent approaches can solve these two problems, which are frequently happened in our daily life. This motivates us to design a more efficient video delivery scheme for concurrent uploading/downloading in MU-MIMO WLANs. Additionally, nowadays most of mobile devices can shoot High-Definition (HD) videos. And many recent released professional cameras are Wi-Fi enabled. This heavier workload trend makes it more urgent to realize such an efficient scheme.

The performance of concurrent video uploading/downloading in MU-MIMO WLANs is unacceptable. The culprits are twofold: 1) frequently varied link conditions, and 2) shared bandwidth. Video codecs are designed for working at a relatively fixed bitrate [7]. This fixed bitrate is based on the estimation of current channel conditions. However, it is untenable to adjust bitrate according to the measured channel condition. The key reason is that since wireless link varies frequently, we cannot get the instantaneous channel quality information. Additionally, given the shared bandwidth, concurrent uploading/downloading HD-videos with high redundancy will bring much heavier workload and lead to high probability of collisions or packets loss. In a nutshell, how to achieve efficiency in simultaneous video uploading/downloading with continuously fluctuated link condition is the main difficulty.

In this paper, we present SimCast (<u>Sim</u>ultaneous), an efficient cross layer design for simultaneous video uploading/downloading in MU-MIMO WLANs. SimCast addresses the above difficulties by leveraging frequency diversity of the wireless channel and spatial similarity of users. We briefly deliver the key components of SimCast step by step.

To cope with frequently varied channel condition, SimCast incorporates rateless codes to ensure the sender transmit video stream in a relatively fixed bitrate [25]. However, state of the art rateless codes (i.e. spinal codes [21]) cannot be directly implemented into MIMO-OFDM (orthogonal frequency division multiplexing) system. It is because spinal codes [21] does not harness frequency diversity. Based on our experimental observations (details in Section II-B), because of frequency selective fading, the decoding capability of each sub-carrier varies. Thus we can achieve higher channel capacity by harnessing frequency diversity. By modifying spinal codes, SimCast employs *F-spinal*, a <u>F</u>ine-grained spinal codes.
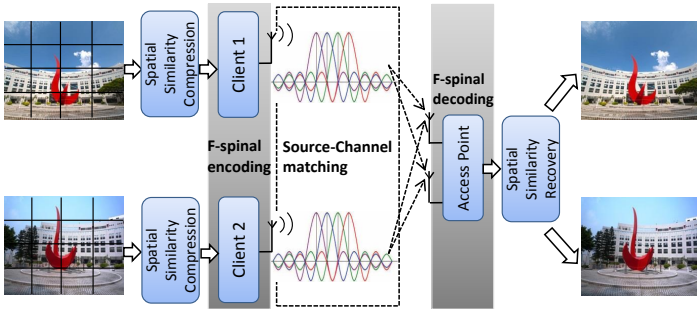
Fig. 1. An example of SimCast framework on concurrent uploading process.

Nevertheless, only by implementing rateless codes cannot ensure efficient performance. It is because that rateless coding regards all data bits as the same importance level. However, each chunk of one video frame has different contribution to the frame's recovery at the receiver side. Given this, we cannot regard each chunk as the same and transmit them as equal.

Another key component of SimCast is its source and channel matching process. This source-channel matching design is motivated by the fact that both the sub-channels' condition and the entropy level of source components are non-uniformly distributed. If we use more reliable sub-channels to transmit high-entropy components and use less reliable sub-channels to transmit low-entropy ones, we can achieve unequal error protection (UEP) by utilizing frequency diversity. Note that conventional approaches [16] [7] leverage high data redundancy to achieve UEP, our source-channel matching scheme achieves the same goal without additional redundancy.

To improve the efficiency one step further, we harness data redundancy in spatial diversity of users. Besides only focusing on transmission links as conventional literature did, we also concern about the clients' behavior. We observe that, in a specific area, most concurrent uploaded/downloaded video streams have high degree of similarity (details in Section II-B). Given this, we refine our basic scheme to further compress data by using this spatial domain similarity.

Given all these key components, as shown in Fig.1, we improve transmission efficiency all the way from senders, through channel, to receivers. We implemented SimCast on USRP software defined radio platform [3]. We also investigated SimCast's scalability with extensive simulations. Evaluations show that SimCast outperforms competitive digital video delivery schemes by nearly 5 dB in PSNR (Peak Signal-to-Noise Ratio) with low SNR situation. Further, due to our efficient design, SimCast achieves a higher throughput gain of nearly 120% on average compared with traditional schemes.

To sum up, our main contributions are as follows:

- To the best of our knowledge, SimCast is one of the first schemes to solve concurrent video uploading/downloading problem.

- SimCast incorporates *F-spinal*, which refines spinal codes by harnessing frequency diversity to achieve higher channel capacity.

- SimCast matches video source's entropy level with OFDM sub-channels condition to achieve UEP without additional data redundancy.

- SimCast leverages users' spatial similarity to increase video compression rate.

The rest of this paper is organized as follows. We illustrate our motivation and experimental observations in Section 2. Section 3 presents the detailed design of SimCast. Experiments and evaluation are discussed in Section 4. Related work is described in Section 5. We conclude our work in Section 6.

II. MOTIVATION AND OBSERVATION

In this section, we discuss about the motivation of SimCast and some observation results which support our design.

A. Motivation: Efficiency in Simultaneous Video Delivery

Mobile video transmission is choppy. *The initial problem is the contradiction between unstable wireless links and the fixed rate requirement for video coding.* It attracts lots of research interests to improve the performance. Previous literatures have studied this issue for long time and some recent works [8] [9] [19] can achieve decent performance. The basic idea is to leverage raw data redundancy or artificially adding redundancy to help the decoder for frame recovery.

With increasing usage of mobile devices, it leads to a new trend of concurrent video uploading/downloading (details in Section II-B). However, *with utilization of MU-MIMO technology for this concurrent video uploading/downloading, data redundancy indeed limits the performance.* Nowadays mobile devices are ubiquitous. There exists high probability that several people share the same AP (Access Point) and some of them uploading/downloading videos simultaneously. There are many common scenarios: 1) many people sitting around in a cafeteria watch the NBA finals or other events (concurrent downloading). 2) A bunch of people shoot video of a famous star or other intriguing incidents (concurrent uploading). However, if we naively implement traditional approaches for MU-MIMO's concurrent video transmission, the performance will be unsatisfiable. *The key reason behind is that with concurrent transmission in the same band, the leverage of high redundancy leads to high probability of collisions and packets loss, which limits the performance.*

Another point worth mentioning is that for concurrent downloading, one may achieve efficient transmission by multicasting the event like a single stream [16] [24]. However, this multicast can only be used for the same stream. More precisely, multicast cannot achieve efficiency with similar but not the same video streams. It is quite common for concurrent downloading similar but not the same videos. Any big news such as 9.11 terrorist attacking incident, variety of news media report the event with similar but not the same video frames. And it is natural that people may choose to watch the news from different media, like CNN or NBC. It is obvious that we can not use multicast to enhance efficiency.

With all these in mind, it seems that nearly no conventional approach can properly solve this current uploading/downloading issue, due to their high data redundancy. And this simultaneously uploading/downloading happens quite frequently in our daily life. This motivates us to design such a scheme SimCast, in order to achieve efficiency all the way from transmitters, through channel, to receivers.
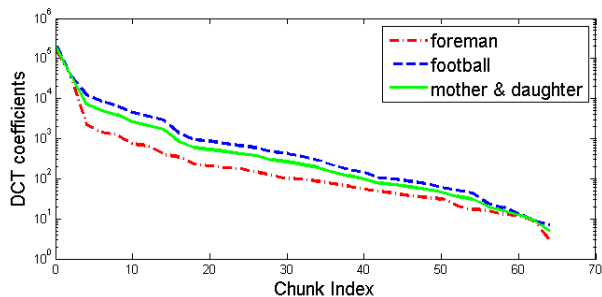
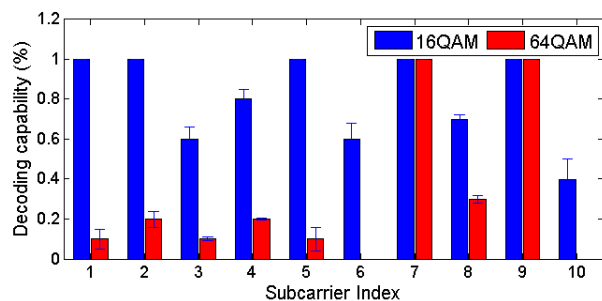Fig. 2. Non-uniform distribution of video chunk's entropy level.

Fig. 3. Decoding capability of the first 10 subcarriers.

(a) Video similarity of concurrent downloading. (b) Video similarity of concurrent uploading.

Fig. 4. Spatial similarity of concurrent video delivery.

B. Observation Results

In this part, we discuss some results we observed from experiments and tracking wireless traffic logs from 10 users for 1 month across our university.

B1. Non-uniform Distribution of Entropy Level

We evaluate some standard video samples, namely *foreman*, *mother and daughter*, *football* [6]. We first divided each frame into 8×8 chunks. After traditional encoding process DCT (Discrete Cosine Transform), we can get their DCT coefficients which is proportional to the chunk's entropy level. Fig.2 plots this non-uniform distribution clearly.

B2. Frequency Diversity

Many prior works have studied frequency diversity on OFDM sub-carriers [10] [15]. Based on previous literature, we conduct experiments on sub-carriers' decoding capability of rateless codes. We implement a prototype of spinal codes [21] on USRP2 [3]. The channel bandwidth is 20MHz in 802.11 with a 64-point FFT modulation. We use 48 sub-carriers to transmit data with 4 sub-carriers severing as pilots. For the limitation of pages, we only present the first 10 sub-carriers' decoding capability results, which is shown in Fig.3.

In Fig.3, the decoding capability value refers to the percentage of correctly decoded symbols. It shows that the decoding capability of different sub-carriers varies drastically because of frequency selective fading.

B3. Spatial Similarity

Traditionally, the MPEG4 standard [4] only leverages time-domain residual to compress data. It is because adjacent video frames will not change much in short periods of time. Since our scenario has the attribute of spatial domain similarity, we can leverage this to further compress data.

The data we collected consists of three parts, namely location, time and wireless traffic information. Here we ignore all the non-video traffic. With the tracking result of 10 individuals over 1 month, it shows that the concurrent video downloading (more than 5 people in one area) happens more than 30 times a week. Concurrent uploading exists 14 times in a month. *These numbers show that concurrent video uploading/dowloading happens quite frequently in our daily life.*

We evaluate the video frame similarity by the metrics of correlation, chi-square, intersection and bhattacharyya, which are commonly used for frame similarity measurement. Then we normalized the value as shown in Fig.4(a) and Fig.4(b). *And it shows that this spatial similarity is high.*
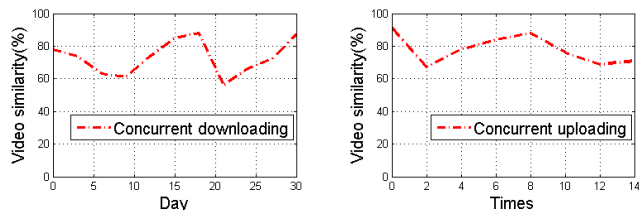
III. SIMCAST DESIGN

Based on MPEG4 standard [4], SimCast is implemented into USRP software defined radio platform. We have given a general view of SimCast in introduction. However, how to ground SimCast into realization is non-trivial. We first state the design challenges of our approach. Then we elaborate the key components' design in detail. For the ease of understanding, we illustrate our design based on the concurrent uploading process. The downloading process is just an inverse version of uploading. And we will discuss the differential part of concurrent downloading in Section III-E.

A. Design Challenges

For the spatial similarity compression, there are mainly two challenges. First, how and when to share the same base frame among mobile users efficiently is challenging. Second, to measure the similarity of different users' video frames usually introduces high computational complexity. How to simplify this measurement with negligible deviation is difficult.

With compressed data bits sending to *F-spinal* encoding process, two issues remain to be concerned. First, as illustrated in Section III-C, there exists transmission redundancy when directly implementing spinal codes [21] into OFDM sub-channels, which limits the overall performance. We need to modify it in order to remedy this limitation. Second, sub-channels with high decoding capability can finish transmission sooner and become available. How to achieve a sub-channel level transmission scheduling process remains to be done.

With encoded bits transmitting into wireless channel, there exists two problems. First, how to accurately measure the entropy level of different video chunks is non-trivial for source-channel matching procedure. Second, how to measure each sub-channel's channel condition efficiently is challenging.

With all these challenges in mind, the following states SimCast design to solve above problems one by one.

B. Spatial Similarity Compression/Decompression

MPEG4 contains three different frame kinds, namely I-frame, P-frame and B-frame. I-frame is the base frame which contains complete frame information. P-frame and B-frame only contain the residual information (i.e. the changes comparing with predecessor (P-frame) or successor (B-frame) frame). More precisely, MPEG4 first divides each I-frame into 8×8 macroblocks. Then it compares with adjacent frames and store the difference between the matching macroblocks into P or B frame. After that it sends the I-frame together with correlated P and B frames as a GOP (Group of Picture) [4] to the encoder.

As shown in Fig.1, on the mobile clients side, before encoding video source, we first compress the source by harnessing spatial similarity. It compares the current frame with the pre-shared base frame. The pre-shared base frame is sent by the receiver (AP) with calculating the similarity of two received video streams. If the similarity in a short period is relatively high (empirical threshold is 60%), AP will send back a shared base frame for mobile clients' further compression. After comparing with the pre-shared base frame in one GOP period, mobile clients calculate the residual and send this residual frame as P or B frames to the source encoding process.

On the AP side, how and when can we send the shared base frame to clients is challenging. First, we receive the concurrent video streams and decode them as traditional MPEG4 decoder do. After recovery from Inverse-DCT (IDCT), Inverse quantization and motion estimation based on received clients' frames, we evaluate the similarity of received frames by calculating Mean Absolute Difference (MAD), which is as equation 1.

$$MAD = \frac{1}{M^2} \sum_{x=1}^{M} \sum_{y=1}^{M} |I_{ori}(x,y) - I_{err}(x,y)| \quad (1)$$

As in equation 1, M represents for $M \times M$ pixels in one macroblock. x, y refer to the pixel's position in this macroblock. $|I_{ori}(x,y) - I_{err}(x,y)|$ depicts this x, y position pixel's entropy value.

MAD function is very similar to another common function MSE in equation 7. We choose to use MAD because it has less computational complexity. MAD only calculates the absolute value with a subtraction whereas MSE needs to do multiplication operation. Even though MSE can calculate the entropy or distortion effect more accurately, noting that we only need to get an overall average distortion value, we choose MAD as our evaluation function to enhance efficiency.

We calculate MAD value and evaluate frames based on this standard. If these concurrent received frames present high similarity, we broadcast an ACK and then send back a frame's bits stream as the shared base frame for a GOP period. In addition, to reduce this calculation overhead further, we decrease the times of this measurement. Instead of calculating MAD value from recovered frames all the time, we only measure it for the next GOP's I-frame compression. If the similarity is still high, we do the same process of sending back base frame. If the frames similarity is low (less than 60%), we separate the base frames to be independent for each video stream and perform as traditional MEPG 4 decoder.
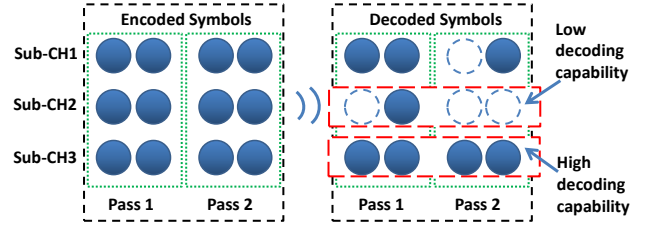
(a) Transmission redundancy in spinal codes on sub-channel 3 (i.e. Sub-CH3).
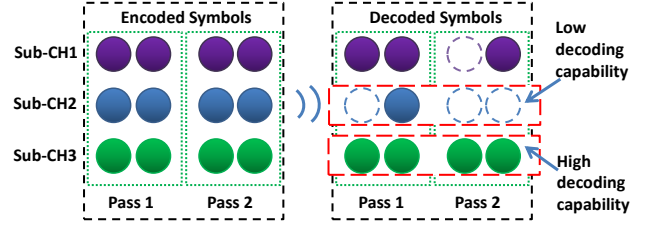
(b) Sub-channel independent transmission in *F-spinal*.

Fig. 5. Modification of spinal codes to avoid transmission redundancy on high decoding capability sub-channels. Note that on the decoded side of Sub-CH3, Fig.5(a) are repeated symbols whereas Fig.5(b) are subsequent new symbols.

C. F-spinal Source Encoding/Decoding

With the frame's macroblocks generated previously in spatial similarity compression, frame encoding process of traditional MPEG4 mainly includes three parts:

1) Discrete Cosine Transform (DCT): It is used to compute the frequency content of different macroblocks. Natural frames do not have sharp changes in any specific areas. Thus the low frequency components' values are very high whereas the high frequency components' values are nearly zero.

2) Quantization: This procedure is used to separate continuous component value into discrete value level. Different quantization strategies fit for different resolution level.

3) Compression: Traditional MPEG4 uses Huffman or Run Length encoding to compress quantized components. And then it transmits packets of compressed frame bits to the networks.

Suppose the original source component matrix is called $\mathbf{X}$. After DCT and Quantization, we get the encoded matrix $\mathbf{Y}$ as equation 2:

$$\mathbf{Y} = \mathcal{Q}\mathcal{D}\mathbf{X} \quad (2)$$

In equation 2, $\mathcal{Q}$ is quantization process matrix and $\mathcal{D}$ is discrete cosine transform matrix. SimCast remains these two steps and changes the compression part with *F-spinal* encoding scheme for transmission. *F-spinal* achieves the compression gain by compressing data blocks' information into each round transmission symbols [21].

Before we encode quantized data with *F-spinal*, we first evaluate the entropy level of macroblocks based on DCT and quantization results, which are used for source-channel matching process (details in Section III-D). After entropy level estimation, we continue the coding process with *F-spinal*.

C1. F-spinal Sub-channel Independent Transmission

Rateless codes is a new class of coding scheme which is suitable for video transmission in wireless networks. On the encoder side, it can encode data in a fixed bitrate without knowing wireless link conditions [8] [9]. The main idea of spinal codes [21] is to use a pseudo-random hash function to encode source message. It outperforms most of existed rateless codes, such as Raptor codes [23], fixed-rate LDPC codes [14].

Algorithm 1 Sub-channel Independent Transmission

1: $i \leftarrow 0$;
2: Sort v by block tag;
3: **for** each block v in G_m **do**
4:    $i \leftarrow m$;
5:    Build a vertex set $V_i = \{v_j\}_{j=1}^n$ from G_i, where v_j is sort by transmission order in V_i;
6:    **if** D_i is not full **then**
7:       Push v into D_i;
8:    **end if**
9:    **loop**
10:       **if** V_i is allow to transmit **then**
11:          transmit v from D_i in V_i sequence;
12:       **else**
13:          **if** D_i is not full **then**
14:             Push v into D_i;
15:             sort v in V_i by transmission order;
16:          **else**
17:             **return** Di is full
18:          **end if**
19:       **end if**
20:    **end loop**
21: **end for**

However, as we mentioned before (in Section II-B), the reason for bad performance of spinal codes is frequency diversity. As depicted in Fig.5(a), spinal codes first divides the bit stream of one chunk of a frame into several message blocks. And then it transmits each block among all OFDM sub-channels in parallel. All the sub-channels transmit the block symbols repeatedly until this block is decoded, whereas the sub-channels with high decoding capability may have received the successfully decoded symbols for many times.

We present *F-spinal* to avoid this transmission redundancy, which is aiming to achieve a fine-grained data transmission. The fundamental idea is to leverage decoding capability diversity. More precisely, after the receiver decodes received symbols on the sub-channels with high decoding capability, it asks the sender to send subsequent symbols instead of receiving repeated symbols. However, in rateless codes, the encoded symbols are correlated through all sub-channels [25] [20]. To broke up this correlation, we need to decode symbols independently on each sub-channel. Thus, instead of transmitting one block through all sub-channels, we transmit each block on one particular sub-channel. Recall that the data bits of each chunk have correlation, to maintain this correlation, we transmit each chunk's blocks sequentially on a particular sub-channel. As shown in algorithm 1 and depicted in Fig.5(b), by avoiding the redundancy in high decoding capability sub-channels, it can achieve higher overall throughput.

F-spinal encoding process is extracted from spinal codes [21]. It consists of a hash function h and a random number generator (RNG). For every k bits message $\bar{m}_i$ ($\{0,1\}^k$), we encodes them into one rateless symbol according to equation 3 and 4:

$$h : \{0,1\}^k \times \{0,1\}^v \rightarrow \{0,1\}^v \qquad (3)$$

$$RNG : \{0,1\}^v \times \mathbb{N} \rightarrow \{0,1\}^c \qquad (4)$$

As in equation 3 4, $\{0,1\}^v$ refers to a v-bit state of spine value. $\{0,1\}^c$ is the modulated rateless symbol we need.

After this rateless encoding process, now we get current encoded matrix $\mathbf{Y}$ as in equation 5, where $\mathcal{R}$ represents *F-spinal* coding process.

$$\mathbf{Y} = \mathcal{R}\mathcal{Q}\mathcal{D}\mathbf{X} \qquad (5)$$

On the receiver side, we perform a similar Maximum-Likelihood (ML) decoding function as spinal codes [21], which is expressed as equation 6. H refers to the channel frequency response of all sub-channels. X_M refers to the encoded symbols of block M, whereas Y is the corresponding received symbols. Since ML algorithm is known with exponential complexity, here we also leverage bubble decoding method in spinal codes [21] to achieve polynomial complexity.

$$\widehat{M} = \arg \min_{M \in \{0,1\}^n} \|Y - HX_M\|^2 \qquad (6)$$

Another point worth explaining is that our sub-channel independent transmission design still has a bottleneck. More precisely, the data bits transmitted on low decoding capability sub-channels may not be decoded by the receiver for long time. These data bits have contributions to the whole frame recovery. However, after our source-channel matching process (in Section III-D), these data bits on low decoding capability sub-channels are almost low entropy level information. These bits have little contribution for frame recovery. Given efficiency as our chief goal, we just drop this low entropy information if we cannot decoded it in several transmission rounds. Thus, there is no bottleneck in *F-spinal* design.

C2. Micro-ACK Feedback

The whole rateless coding/decoding process is illustrated above. Since we achieve sub-channel independent transmission, sub-channels with high decoding capability can finish transmission sooner and become available. How to efficiently leverage these high throughput sub-channels is a vital issue.

We design a simple but efficient feedback mechanism. Assume that the sub-channel condition will not vary drastically within a few transmission rounds, which is reasonable in real-world wireless transmission. Then 802.11 preamble [11] can be used periodically for synchronization to reduce the MAC layer overhead. Note that the sender does not need to know channel condition in traditional rateless codes, we want to minimize this feedback overhead. On the receiver side, instead of sending back a traditional ACK packet after each transmission round, we only send back one modulated OFDM symbol. This OFDM symbol consists of N bits, where N is the number of sub-channels. We use BPSK (Binary Phase Shift Keying) to modulate 1 bit represents for 1 sub-channel, where 1 refers to successfully decoded and 0 refers to not decoded.

After every transmission round, the sender has to pause for this 1 symbol feedback from the receiver. After the sender

gets this Micro-ACK symbol, it knows which sub-channels are vacant and arranges them with subsequent data blocks. Thus, we implement a light-weight feedback and scheduling mechanism with negligible overhead.

D. MU-MIMO Source-Channel Matching Process

Due to the modification of sub-channel independent transmission in *F-spinal*, with rateless encoded bits, if we match the video source and sub-channels properly, it can enhance transmission reliability for high entropy components. Given this, we need to design the source-channel matching process, which is used to schedule different entropy level chunks with different decoding capability sub-channels.

D1. Entropy Level Estimation

Recall that before rateless coding, we need to use DCT and quantization information to assist our entropy level estimation process. Since MPEG4 uses threshold to quantize continuous components' value, we cannot distinguish distortion level of source components with the same quantized threshold value. Recall that DCT coefficients can be used for separating source components with the same quantization value, we leverage both former DCT coefficients and Mean Squared Error (MSE) value to do this entropy level estimation. Note that DCT coefficients of each macroblock are calculated by previous DCT process, it does not bring any computational overhead.

Conventional approaches may only use DCT coefficients for entropy level estimation [19] [9]. We argue that this DCT-based metric is not sufficient. The main reason is that DCT coefficients only represent the importance level of one macroblock in its own frame. It does not concern about this macroblock's distortion effect to the whole GOP frames, which MSE measures this effect. Given this, combining DCT and MSE will be more appropriate and accurate for entropy level estimation. MSE can be calculated as equation 7. Since the characters in equation 7 are similar to equation 1, we do not repeat the character explanation.

$$MSE = \frac{1}{M^2} \sum_{x=1}^{M} \sum_{y=1}^{M} [I_{ori}(x,y) - I_{err}(x,y)]^2 \quad (7)$$

We calculate the product of non-zero DCT coefficient and its corresponding MSE value. And then we rank the macroblocks distortion effect in descending order, which is as equation 8.

$$\text{Sort: From } \max \text{ to } \min$$
$$\widehat{E} = DCT_{coe} \times MSE_{coe} \quad (8)$$
$$\text{Subject to } \forall DCT_{coe} \neq 0$$

In equation 8, $\widehat{E}$ represents distortion effect value. And MSE_{coe} is the coefficient value calculated from equation 7. DCT_{coe} represents DCT coefficient of each macroblock.

D2. Channel Condition Estimation

After we encode our source components with *F-spinal codes*, we measure current MU-MIMO wireless link condition.
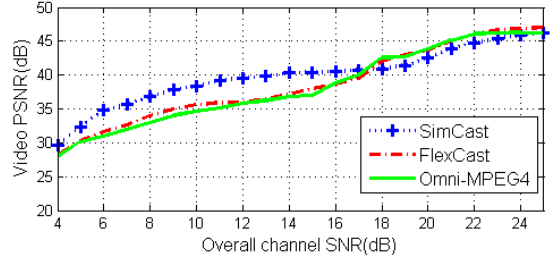

Fig. 6. SimCast performance with unknown SNR. It outperforms FlexCast and Omni-MPEG4 upto 5 dB in low SNR scenarios

There are many existed approaches for accurately estimating MIMO-OFDM sub-channels conditions [13] [12] [15]. Different from these methods, we estimate the sub-channel condition with our Micro-ACK feedback.

Recall that the sender schedules vacant sub-channels based on received Micro-ACK from the receiver, it contains decoding capability information of all sub-channels. Assume that the sub-channel condition will not vary drastically within a few transmission rounds. Here we use a sliding window method to evaluate each sub-channel's condition.

For the ith round we get one Micro-ACK symbol $\mathbb{H}_i$ as equation 9, where n represents the number of sub-channels.

$$\mathbb{H}_i = \{h_{1,i}, h_{2,i}, h_{3,i}...h_{n,i}\} \quad (9)$$
$$\text{Subject to } \forall h_{j,i} \in \{0,1\}, \text{ where } j \in \{1,2,...n\}$$

We collect the Micro-ACK symbols from i^{th} round to $(i+m)^{th}$ round. Here m can be adjusted by the variation frequency of channel conditions. Then we calculate the value $\widehat{H}_j$ with equation 10 as j^{th} sub-channel's channel condition value.

$$\widehat{H}_j = \sum_{k=i}^{k=i+m} h_{j,k} \quad (10)$$
$$\text{where } j \in \{1,2,...n\}$$

With concerning about efficiency, we just use the simplest one-to-one matching method. We sort sub-channel condition in descending order by $\widehat{H}$ value of each sub-channel. And then we match the highest entropy level component with the sub-channel maintaining the highest $\widehat{H}$ value.

To implement this process into our system, we need to add the matching matrix into encoded source components. So we get $\mathbf{Y}$ in equation 11, where $\mathcal{M}$ represents matching matrix, which contains the information of matching process.

$$\mathbf{Y} = \mathcal{M}\mathcal{R}\mathcal{Q}\mathcal{D}\mathbf{X} \quad (11)$$

On the receiver side, it also keeps Micro-ACK information from i^{th} round to $(i+m)^{th}$ round and calculates $\widehat{H}$ value in the same way. Thus it can decode the $\mathcal{M}$ matching matrix.
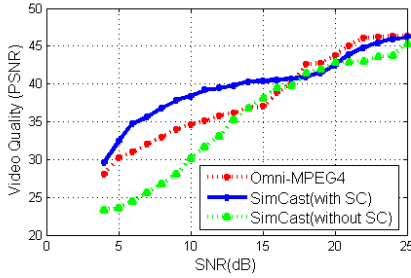
Fig. 7.   Impact of Spatial Compression (SC).
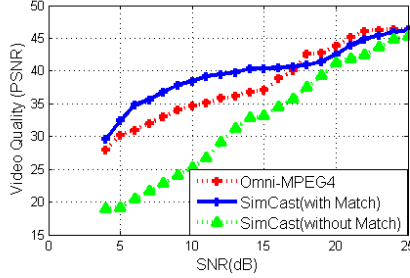

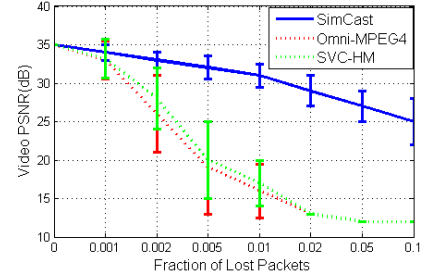Fig. 8.   Source-Channel matching effect.

Fig. 9.   Robustness with packet loss.

E. Supplement for the Downlink

Generally, the downlink process is just the identical inverse of uploading process we illustrate above. We explain the differential part between downlink and uplink in this section.

For the spatial similarity compression, on the downlink, AP will first evaluate the similarity of concurrent downloaded video streams and then decide whether to share the base frame. Another differential part is on MU-MIMO downlink, if the number of concurrent downloading users is no more than the surrounding APs' antennas, we can leverage beamforming [17] to achieve better performance. For both uplink and downlink, if the number of users is large, we use traditional CSMA-based channel contention mechanism for transmission.

IV. IMPLEMENTATION AND EVALUATION

The implementation of SimCast is based on MPEG4 encoding/decoding library [4]. This section first delivers the prototype implementation of SimCast on USRP platform. And then for large scale WLANs, we simulated SimCast on Matlab. We evaluate SimCast's performance with both experimental and simulation results.

A. Experiment Setup

Hardware and Testbed: We implement SimCast into a 2×2 MU-MIMO system with four USRP2 [3] boards and XCVR2450 daughterboards, which operate in the 2.4GHz range. We use IEEE 802.11 OFDM standard [11], which has 64 sub-carriers (48 for data). We connect USRP2 nodes via Gigabit Ethernet to our laboratory PCs, which are all equipped with a qual-core 3.2GHz processor, 3.3GB memory and running Ubuntu 10.04 with GNURadio software platform [2]. Since USRP2 boards cannot support multiple daughter boards, we combine two USRP2 nodes with an external clock [5] to build a two-antenna MIMO node. We use the other two USRP2 nodes serving as clients.

Test Videos: We choose from a group of standard test video samples [6], namely, *Foreman* (little motion), *Mother and Daughter* (medium motion), *Football* (significant motion). We use the SIF format (352×240 pixels with 30fps).

Performance Metric: We evaluate the video quality by Peak Signal-to-Noise Ratio (PSNR), which is a standard metric for video and image quality measurement. The calculation function is as equation 12. l refers to the number of bits to represent pixel luminance and it is typically 8. MSE is the

same as we have mentioned before in equation 7.

$$PSNR = 20 \log_{10} \frac{2^l - 1}{\sqrt{MSE}} \qquad (12)$$

To give a general idea of PSNR value, if the video's PSNR is less than 16dB, we can see apparent noise. The video with PSNR value more than 40dB is in good quality. An increasing of 3 dB in PSNR means the video quality is twice better.

Another metric we focus on is the network throughput which will be discussed in the following evaluation part.

B. Comparison Schemes

- **SimCast:** This is our scheme (details in Section III).

- **FlexCast:** This is a state-of-the-art video transmission scheme, which uses extra bits redundancy as parity bits to enhance the accuracy of recovering process in decoder.

- **Omniscient MPEG4 (Omni-MPEG4):** This is based on traditional MPEG4 [4]. We assume that this MPEG4 scheme knows the current channel condition and based on this information to choose the optimal bitrate. It concerns both data encoding and transmission.

- **SVC-HM:** This scheme uses scalable extension of H.264. In this scheme, the entropy coding bits are separated into different layer of importance level.

C. Performance with Unknown SNR

In this experiment, we randomly set the channel SNR value from 4dB to 25dB by moving USRP2 nodes. Only Omni-MPEG4 knows the accurate channel condition with precoding CSI feedback. It tries its best to adapt to the optimal coding bitrate. SimCast and FlexCast only use the coarse rate adaptation. More precisely, in order to exclude the effect of accurate rate adaptation, we only use three basic schemes for rough static rate adaptation. When the channel SNR is below 10dB, we use QPSK (Quadrature Phase Shift Keying) constellation scheme. When SNR is between 10-20dB, we use 16-QAM (Quadrature Amplitude Modulation) scheme. When overall SNR is above 20dB, we use 64-QAM. Note that this coarse rate adaptation is inaccurate, it apparently cannot be fit for continuous changed wireless condition. This experiment is repeated 10 times for each channel condition and we calculate the average values.
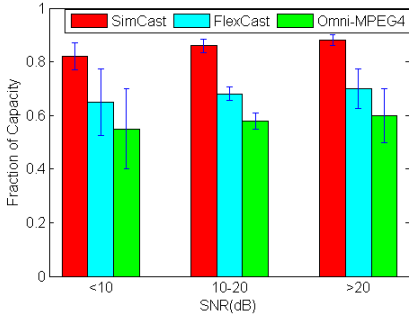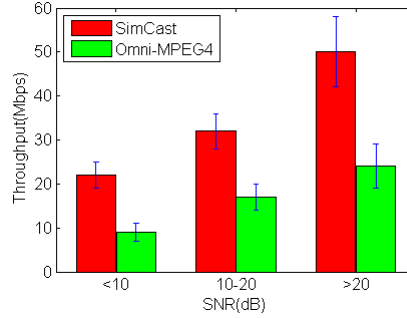
Fig. 10. Channel capacity of SimCast.
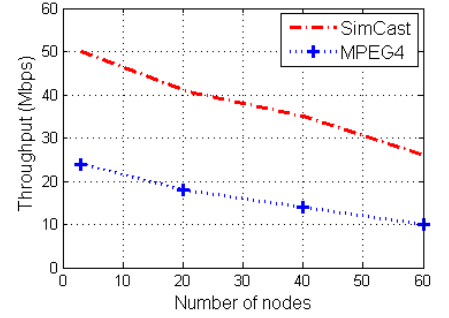

Fig. 11. Overall throughput of SimCast.

Fig. 12. SimCast performance in contention scenarios with different number of nodes.

The results in Fig.6 show that SimCast outperforms Omni-MPEG4 and FlexCast by nearly 5dB PSNR gains in low SNR scenarios. Furthermore, SimCast can achieve almost the same performance as Omni-MPEG4 in good channel conditions. Note that compared with FlexCast and Omni-MPEG4, SimCast achieves lower PSNR value at high SNR, it is because its high data compression ratio with less redundancy. Anyhow, this difference is merely little.

The reason for SimCast's good performance are as follows. First, *F-spinal* can harness frequency diversity to decrease transmission redundancy. Second, our spatial compression and source-channel matching process make great contributions to SimCast's high performance in low SNR scenarios.

D. Spatial Compression and Source-Channel Matching Effects

In this experiment, we uses with/without method to measure both Spatial Compression (SC) effect and source-channel matching effect on SimCast's performance.

Result in Fig.7 shows the impact of SC on video quality of SimCast. Note that in medium SNR situation (SNR around 18 dB), SimCast without SC even performs better than with it. The key behind is that without SC, the decoder can receive more data redundancy to help frame recovery. Thus it enhance the performance. However, when in other channel conditions, SimCast with SC achieves better video quality than without SC. The key reason is that SimCast with SC can be more robust to coarse rate adaptation.

Fig.8 plots the impact of UEP protection that source-channel matching process achieves. This UEP is a dominant factor for SimCast's performance. The performance drops drastically in without source-channel matching scenarios.

E. Robustness with Packet Loss

Here we mainly focus on the schemes' robustness with packets loss. However, since FlexCast separates one macroblock's bit stream into several parts, which is very vulnerable to packet loss. More precisely, if one part of the macroblock value is missing, the whole macroblock value will be impossible to recover. Given this, it is unfair to compare packet loss effect of FlexCast with others.

We compare SimCast with two most commonly used schemes, namely SVC and Omni-MPEG4. With varied percent of packets loss, Fig.9 shows that SimCast is more robust than these two schemes. Even one tenth of the packets are lost, SimCast can remain relatively good video quality. This attribute is due to our entropy level estimation process, which combine DCT coefficient and MSE value together. Comparatively, conventional schemes only measure distortion level by DCT coefficient, which is not appropriate. Additionally, *F-spinal* breaks up the transmission correlation between sub-channels, which also contributes to this robust performance.

F. Channel Capacity and Overall Throughput

In this experiment, we compare the Channel capacity and network throughput. With variety range of SNR, Fig.10 shows the channel capacity of SimCast is higher than competitive schemes, namely FlexCast and Omni-MPEG4. We normalize the results by making the channel capacity divided by Shannon capacity. In high SNR condition (i.e. SNR > 20dB), the channel capacity that SimCast achieves is 90% on average, which is very close to the upper bound of Shannon capacity.

Fig.11 shows that the overall throughput that SimCast achieves, which compares with the performance of Omni-MPEG4. Note that SimCast achieves more than 140% overall throughput over Omni-MPEG4 in low SNR scenarios (i.e. SNR < 20dB). And it outperforms Omni-MPEG4 over nearly 120% on average.

Performance in Contention Scenarios: For the evaluation on the scalability performance of SimCast, we conduct simulations on Matlab. We set one 3-antenna AP and 1-antenna client nodes ranging from 3-60. We set SNR ranging from 5 to 30 dB and calculate the average throughput. For the same error-prone reason of FlexCast as mentioned in Section IV-E, we compare SimCast with conventional MPEG4 scheme. Both schemes are using CSMA/CA MAC (Media Access Control) protocol. Fig.12 plots that, with efficient design of SimCast, it can outperform MEPG4 with nearly 120% gain on average, which also validates our experimental results. It shows that with burst of concurrent video transmission, SimCast can still perform better than existed schemes.

These good performances are because of our *F-spinal* codes, which removes transmission redundancy in high decoding capability sub-channels and breaks up transmission correlation among sub-channels. Further, source-channel matching and spatial compression enhance these good performances.

V. RELATED WORK

Wireless video delivery has been a hot topic for several years [24]. Recent joint source-channel approaches achieve

graceful video delivery performance by adding redundancy into the original data source. Most of these schemes are motivated by the fact that with more data redundancy can make video stream become less error-prone. SoftCast [16] and ParCast [19] use several linear and lossless transformations to encode the video source. And they directly transmit encoded analog source signal over channels. FlexCast [7] maintains most of the MPEG4 [4] codec standard and only replaces the entropy coding scheme with its own rateless codes. Then it transmits compressed digital bits over wireless links. However, all these schemes cannot achieve graceful performance in concurrent video delivery due to their high data redundancy.

One different approach, Apex [22] achieves UEP by its specially designed constellation mapping method. However, it needs to be implemented in high SNR and must know the channel SNR to adjust the symbol mapping procedure [7], which is unpractical in real-world mobile scenarios. SimCast can achieve the same UEP goal without pre-knowledge of channel SNR and can be used in low SNR scenarios.

SimCast is also related to the literatures that harness frequency diversity [15] [10]. Halperin et al. show the existence of frequency diversity on sub-channels' transmission [15]. Li et al. leverage it to improve retransmission efficiency [18]. All these works are harnessing sub-channel condition diversity and achieve better throughput. SimCast's source-channel matching process is inspired by these works. However, different from these schemes, SimCast explores frequency diversity especially for wireless video delivery. More precisely, different from general data bits, we need to concern about entropy level of stream bits for video quality and processing overhead.

To sum up, nearly none of previous works ever focus on concurrent video uploading/downloading problem. Different from all these previous literatures, our approach achieves much more efficient concurrent video uploading/downloading design with negligible video quality loss.

VI. CONCLUSION

SimCast is motivated by the fact that the performance of frequently happened concurrent video uploading/downloading is unacceptable. The main problems are frequently varied link condition and shared bandwidth. First it modified spinal codes (*F-spinal*) to ensure relatively fixed transmission bitrate and avoid transmission redundancy on high decoding capability sub-channels. In addition, SimCast incorporates the source-channel matching procedure to achieve UEP without introducing data redundancy. Further, due to our specific concurrent uploading/downloading scenarios, we leverage the spatial similarity of users to further compress data. SimCast provides an efficient and robust cross-layer coding scheme that exploits the unique attributes of video source, wireless channels and transceivers. SimCast can provide both high throughput and good video quality.

ACKNOWLEDGEMENTS

REFERENCES

[1] Cisco visual networking index: Forecast and methodology 2010-2015. *http://www.cisco.com*.

[2] E. blossom. gnu software defined radio. *http://www.gnu.org/software/gnuradio*.

[3] Ettus research llc, universal software radio peripheral. *http://www.ettus.com*.

[4] ffmpeg. *http://ffmpeg.org*.

[5] Oscilloquartz sa, osa 5200b gps clock. *http: //www.oscilloquartz.com*.

[6] Xiph.org test media. *http://media.xiph.org/video/derf*.

[7] S. T. Aditya and S. Katti. Flexcast: Graceful wireless video streaming. In *Proceedings of ACM MobiCom*, 2011.

[8] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi. Adaptive unicast video streaming with rateless codes and feedback. *IEEE Trans. on circuits and systems for video technology*, 20(2), 2010.

[9] L. Al-Jobouri, M. Fleury, and M. Ghanbari. Adaptive rateless coding for data-partitioned video streaming over a broadband wireless channel. In *6th Conference on Wireless Advanced*, 2010.

[10] A. Bhartia, Y.-C. Chen, S. Rallapalli, and L. Qiu. Harnessing frequency diversity in wi-fi networks. In *Proceedings of ACM MobiCom*, 2011.

[11] E. Blossom. *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*. IEEE Std 802.11, 2012.

[12] A. Chhetri, H. Nguyen, G. Scalosub, and R. Zheng. On quality of monitoring for multi-channel wireless infrastructure networks. In *Proceedings of ACM MobiHoc*, 2010.

[13] R. Crepaldi, J. Lee, R. Etkin, S.-J. Lee, and R. Kravets. Csi-sf: Estimating wireless channel state using csi sampling and fusion. In *Proceedings of IEEE INFOCOM*, 2012.

[14] R. Gallager. Low-density parity-check codes. *IRE Trans. Information Theory*, 8(1):21–28, 1962.

[15] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 packet delivery from wireless channel measurements. In *Proceedings of ACM SIGCOMM*, 2010.

[16] S. Jakubczak and D. Katabi. A cross-layer design for scalable mobile video. In *Proceedings of ACM MobiCom*, 2011.

[17] C. Jiang and L. J. Cimini. Energy-efficient multiuser mimo beamforming. In *Proceeding of IEEE Information Sciences and Systems (CISS)*, 2011.

[18] L. E. Li, K. Tan, H. Viswanathan, Y. Xu, and Y. R. Yang. Retransmission $\neq$ repeat:simple retransmission permutation can resolve overlapping channel collisions. In *Proceedings of ACM MobiCom*, 2010.

[19] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang. Parcast: Soft video delivery in mimo-ofdm wlans. In *Proceedings of ACM MobiCom*, 2012.

[20] I. Matsuda, Y. Nomoto, K. Wakabayashi, and S. Itoh. Throughput of rateless codes over broadcast erasure channels. In *16th European Signal Processing Conference*, 2008.

[21] J. Perry, P. A. Iannucci, K. Fleming, HariBalakrishnan, and D. Shah. Spinal codes. In *Proceedings of ACM SIGCOMM*, 2012.

[22] S. Sen, S. Gilani, S. Srinath, S. Schmitt, and S. Banerjee. Design and implementation of an approximate communication system for wireless media applications. In *Proceedings of ACM SIGCOMM*, 2010.

[23] A. Shokrollahi. Raptor codes. *IEEE Transactions on Information Theory*, 52(6), 2006.

[24] K. Sundaresan and S. Rangarajan. Scheduling algorithms for video multicasting with channel diversity in wireless ofdma networks. In *Proceedings of ACM MobiHoc*, 2011.

[25] Y. Yang and N. B.Shroff. Throughput of rateless codes over bradcast erasure channels. In *Proceedings of ACM MobiHoc*, 2012.