

iCast: Fine-Grained Wireless Video Streaming Over Internet of Intelligent Vehicles

Lu Wang¹, Hailiang Yang, Xiaoke Qi², Jun Xu¹, and Kaishun Wu¹

Abstract—Recent years have witnessed a steep grow in the multimedia-oriented Internet of Things (IoT) over vehicular networks. Huge volume of multimedia traffic generated from the in-built IoT devices should be delivered among vehicles and immediate surroundings in real time. However, as network nodes with higher mobility, vehicles often experience more unpredictable wireless channels. Such time-frequency diversity poses substantial challenges to achieve pervasive and real-time multimedia connectivity. The hurdle lies in the inability of automatically approaching the subcarrier level channel variations in the existing video codecs. With coarse-grained traffic delivery rate, video decoding fails, and intermittent connection occurs. To break this stalemate, we propose a fine-grained wireless video streaming strategy, namely iCast, that intelligently achieves the most appropriate data rate and frame protection for multimedia traffic in highly mobile vehicular environments. The insight of iCast is a simple joint source-channel rateless code. It reaps the benefits of the frequency diversity to provide fine-grained data rate for the channel in conjunction with suitable protection for the source. Our experiments show that, by harnessing frequency diversity in mobile environments, iCast outperforms the existing competitive wireless video delivery schemes by up to 5 dB peak signal-to-noise ratio.

Index Terms—Frequency diversity, Internet of Intelligent Vehicles (IoIV), rateless code, wireless video streaming.

I. INTRODUCTION

RECENT advances in wireless communications, embedded systems, and sensor technologies have redefined the perception of a vehicle. As a smart object equipped with

Manuscript received April 16, 2018; revised August 13, 2018; accepted September 20, 2018. Date of publication September 28, 2018; date of current version February 25, 2019. This work was supported in part by China NSFC under Grant 61872246, Grant 61502313, Grant 61872248, Grant 61472259, Grant 61836005, Grant 61603390, and Grant 61601308, in part by the Joint Key Project of the National Natural Science Foundation of China under Grant U1736207, in part by the Guangdong Natural Science Foundation under Grant 2017A030312008, in part by the Shenzhen Science and Technology Foundation under Grant JCYJ20170817095418831, Grant JCYJ20170302140946299, and Grant JCYJ20170412110753954, in part by the Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China under Grant 161064, in part by the Guangdong Talent Project under Grant 2015TX01X111, in part by GDUPS (2015), and in part by the Tencent Rhinoceros Birds-Scientific Research Foundation for Young Teachers of Shenzhen University. (Corresponding author: Kaishun Wu.)

L. Wang, H. Yang, J. Xu, and K. Wu are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: wanglu@szu.edu.cn; yanghailiang2016@email.szu.edu.cn; xujun@szu.edu.cn; wu@szu.edu.cn).

X. Qi is with the School of Information Management for Law, China University of Political Science and Law, Beijing 100190, China (e-mail: qxiaoake@cupl.edu.cn).

Digital Object Identifier 10.1109/JIOT.2018.2872518

wireless interfaces, sophisticated sensors, and high illuminating infrared cameras, today's vehicle enables ubiquitous sensing and connectivity amid immediate surroundings [1]. In this backdrop, traditional vehicle ad-hoc networks have evolved toward Internet of Intelligent Vehicles (IoIV), handling driving and traffic issues via information obtained from versatile in-built Internet of Things (IoT) devices. The steep grow in IoIV boosts the proliferation of multimedia oriented and bandwidth-hungry applications. Industry research predicts that by the year 2019, the wireless video will become the dominant driver for IoT traffic [2]. Representative applications include cooperative navigation, on-board infotainment as well as visualization driving assistance, leading to an unlimited demand for pervasive and real-time wireless connectivity [3].

However, considering the high mobility of vehicles, fundamental differences lie between video streaming in IoIV and other existing video streaming systems in conventional IoT. In particular, wireless video streaming in IoIV is not a simple upgrade of its low-mobility predecessor. Unique channel characteristics manifest in vehicular channels, including inherent nonstationarity, multipath, and shadowing by immediate surroundings and high Doppler effects [4]. Accordingly, communications among vehicular nodes are susceptible to data distortion and intermittent connection, making it a great challenge for large volume and high-quality video streaming. We need new thoughts and insights in protocol design for ubiquitous wireless video delivery over IoIV.

Driven by the demand of more connected vehicles and better quality of wireless video streaming, considerable efforts have been made to fulfill the huge volume of multimedia traffic requirements, in line with the error-prone wireless channel characteristics. ParCast [5] separated the source and channel encoding, and applied an allocation strategy by jointly considering the source and the channel. DCast [6] shared the similar insights, where transmission power was allocated to the video frames according to their size and distribution. To combat the quick channel variation, rateless codes, such as strider [7] codes and spinal [8] codes, have received renewed interest as a new frontier for rate adaptation. Unlike the state-of-the-art encoding schemes that requires instant feedbacks on channel conditions at the transmitter side, rateless codes enable a self-adaptive code rate in terms of the instant channel statuses at the receiver side. Several works have incorporated rateless codes into vehicular data dissemination, as they offer the promise of better understanding of the unpredictable channel [9]–[11]. In particular, FlexCast [12] leveraged rateless code to protect the video bits in terms of their distortion. Albeit with

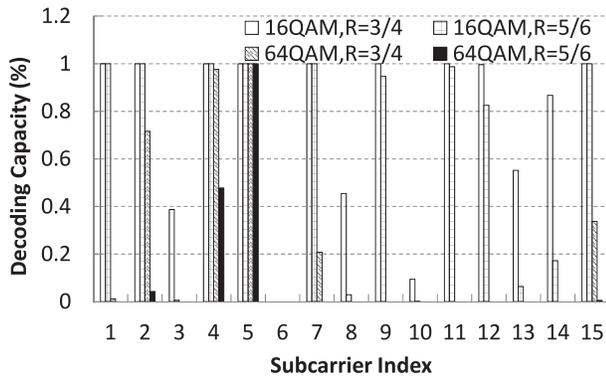


Fig. 1. Decoding capacity of the first 15 subcarriers over an 802.11 frequency selective fading channel. Different subcarriers exhibit various decodabilities in terms of frequency diversity over high mobility environments.

the assistance of rateless codes, the above approaches achieve more reliable communications, the encoding procedures are too coarse for wireless video streaming. With multipath effect, important frames may be stuck on a certain channel component, and thus the transmission efficiency is highly degraded. This problem is nontrivial, with considerable design issues entangled from the source coding to the channel coding.

In this paper, we observe that the hurdle lies in the inability of automatically approaching the subcarrier level channel variations in the existing video codecs. Due to frequency selective fading and frequency independent attenuation under highly dynamic environments, the signals carried by different subcarriers experience quite diverse transmission quality, introducing a significant frequency diversity across the entire channel. As a result, various decoding capacities are observed on each subcarrier. To be intuitive, we introduce the term “decoding capacity” or “decodability,” which represents the probability that a receiver can correctly recover the data on a particular subcarrier. Such decodability highly relies on the channel condition of that subcarrier. With the same combination of modulation and channel coding schemes, the decodability on different subcarriers exhibits quite diversely. Furthermore, we conduct several simulations to measure the decodability over a typical frequency selective fading 802.11 Wi-Fi channel. As depicted in Fig. 1, 1 refers to that the receiver can correctly decode all the received symbols, and 0 means the receiver cannot decode anything. Due to the frequency selective fading, each subcarrier exhibits quite different decodability. However, as rateless codes are inherently time-frequency correlated, they are unable to reap the benefits of the frequency diversity, either for fine-grained traffic delivery or frame protection.

Inspired by the above observations, we propose iCast, a fine-grained wireless video streaming scheme that intelligently achieves the most “appropriate” data rate and frame protection for video traffic over IoIV. The insight is a simple joint source-channel rateless code. It breaks down the frequency correlation in a rateless code while remains its self-adaptive feature. Sequentially, the frequency diversity in highly dynamic channel components can be harnessed to provide fine-grained data rate and suitable protection for each source component. In particular, iCast is based on OFDM modulation.

It abstracts MPEG-4 [13] as the source code, and rateless spinal code as the channel code. It constructs a subcarrier independent coding scheme for fine-grained rateless transmission. By assessing the subcarrier-wise decodability via soft information, along with video frame regrouping in terms of bit-wise importance, iCast offers fine-grained unequal protection through appropriate matching between the source and the channel.

We have implemented iCast on GNUradio/USRP2 platform by modifying the MPEG-4 codecs and 802.11g/n channel code. Our experiments show that compared with traditional fixed rate LDPC code with explicit rate adaptation and rateless FlexCast code, iCast can better approach the channel capacity in a fine-grained way under different channel conditions, and provide suitable protection for video streaming. In summary, our main contributions over existing works are as follows.

- 1) We observe a distinct decodability diversity over IoIV communications, and propose iCast, an intelligent joint source-channel code to harness this diversity for rate adaptation and frame protection. To the best of our knowledge, this is the first work of its kind in the literature to utilize such decodability diversity to approach the channel capacity in conjunction with unequal protection for wireless video streaming over IoIV.
- 2) We introduce soft information to assess the subcarrier-wise decodability of the channel, and quantize bit difference to evaluate the bit-wise importance of the source. Compared to traditional unequal protection for video streaming, iCast unleashes the potential of rateless code and achieves a more intelligent video delivery approach.
- 3) We demonstrate the feasibility of iCast through GNU testbed, and conduct extensive trace-driven simulations to evaluate its performance for video streaming. Numerous results show that iCast outperforms competitive wireless video delivery schemes by up to 5 dB peak signal-to-noise ratio (PSNR).

II. RELATED WORK

iCast is related to source and channel coding and modulation design in existing wireless networks.

As for source and channel coding, ParCast [5] first separated the source and the channel, and applied an allocation strategy to jointly improve the video streaming performance. DCast [6] shared the similar insights, where transmission power was allocated to the video frames according to their size and distribution. Some recent works [14], [15] focused on video content in D2D communications for video delivery, which leveraged the properties of the on-demand video, i.e., asynchronous content reuse, to cache the content on the wireless device. In particular, SoCast [16] stimulated effective cooperation among mobile users to restore incomplete video frames. As multimedia IoT tasks are usually delay-sensitive, Long *et al.* [17] proposed an edge computing framework, which leverages cooperative processing on mobile devices for video streaming. The above approaches work fine under low-mobility environments. When it comes to IoIV, the high

dynamism induces data distortion and intermittent connection video delivery. On the contrary, iCast not only jointly considers the source importance and channel impact, but also reaps the benefit of fine-grained rateless codes to fight against the highly mobile IoIV and improve the video streaming performance.

iCast also relates to rateless codes design, which is born to combat the quick channel variation. As two representatives: 1) LT codes [18] and 2) Raptor codes [19] have displayed desirable paradigms for rate adaptation. Further, Strider emerged as a practical rateless code for wireless communication systems. It proposed a layered approach to generate linear combinations of symbols in a rateless manner. Although strider achieves the channel capacity at some fixed signal-to-noise ratio (SNR), the layered structure makes it complex to extend to other structures. On the other hand, Spinal codes [20] utilized hash function to produce rateless symbols. The sequential hashing function provides a rich structure, and simplifies the design compared with layered approach. However, the above rateless codes are not capable of combating the frequency selective fading introduced by dynamism, as the coding structure are correlated in frequency domain. iCast successfully breaks down such correlation while remains the rateless feature. Therefore, it achieves fine-grained data rate and bit protection for wireless video streaming.

It is noted that there are already several works that incorporated rateless codes into vehicular data dissemination, as they offer the promise of better understanding of the unpredictable channel [9]–[11]. FlexCast [12] leveraged rateless codes to protect the video bits in terms of their distortion. SoftCast [21] utilized rateless codes to compress the video source at the content server. Albeit with the assistance of rateless codes, the above approaches achieve more efficient communications compared with traditional fixed-rate schemes, the encoding procedures are too coarse for video streaming in IoIV. With multipath and Doppler effect, important frames may be stuck on a certain channel component, and thus the transmission efficiency is highly degraded. By breaking down the frequency correlation while remains the rateless feature, iCast achieves better video streaming over IoIV.

Finally, iCast is related to prior work that leverages frequency diversity to improve the wireless connectivity. FARA [22] and Adaptive OFDM modulation [23] try to improve the data rate by adjusting coding and modulation parameters across different frequencies to achieve a better data rate. A-HDAVT [24] harnessed the diversity in both video contents and realistic fading channels to improve the video streaming performance in mobile networks. But these fixed-rate-based strategies rely on the transmitter's perception to conduct adaptation for the receiver's sake. Thus, either costly feedback or inaccurate inference is required. Unlike the above works, iCast is potentially a receiver-based rate adaptation, where the receiver decides the transmission data rate for its own sake. With proper feedbacks on the subcarrier decodability, the transmitter offers better protection to the important video bits, by matching them with high decodable subcarriers in a rateless manner.

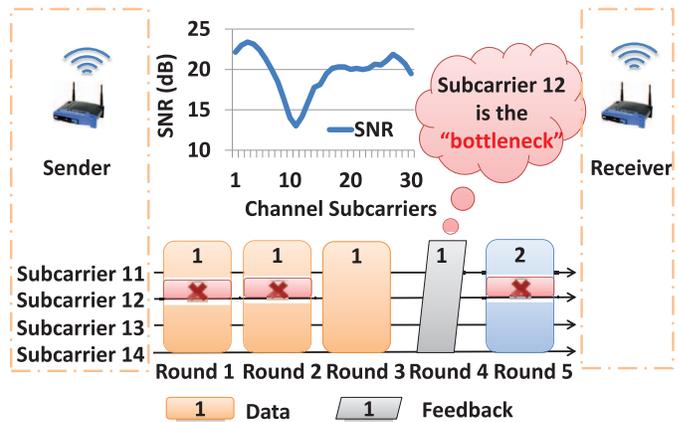


Fig. 2. Significant impact of frequency diversity in rateless codes.

III. OVERVIEW AND DESIGN CHALLENGE

We start with a brief explanation of why frequency diversity is essential for video streaming in IoIV. As a design guide, we first investigate the capacity gap between AWGN channel and frequency-selective fading channel. In a frequency-selective fading channel, the channel capacity of an OFDM-based system C_N is expressed as

$$C_N = \frac{1}{N} \sum_{l=0}^{N-1} \log_2 \left(1 + \frac{|h_l|^2 P}{N_0} \right) \quad (1)$$

where (P/N_0) is the SNR. N is the number of subcarriers, and h_l is the channel response of the l th subcarrier. Equation (1) simply describes the capacity of a wideband channel, and does not consider frequency diversity. Therefore, when subcarriers endure deep fading, the channel capacity is greatly restricted. To illustrate this problem, we determine the capacity gap G_{gap} between AWGN channel C_{AWGN} and frequency-selective fading channel C_N as follows:

$$\begin{aligned} G_{\text{gap}} &= C_{\text{AWGN}} - C_N \\ &= \log_2 \left(1 + \frac{P}{N_0} \right) - \frac{1}{N} \sum_{l=0}^{N-1} \log_2 \left(1 + \frac{|h_l|^2 P}{N_0} \right). \end{aligned} \quad (2)$$

When N is fixed, the value of G_{gap} is determined by h_l on each subcarriers. As some subcarriers have very small h_l s due to deep fading, G_{gap} turns out to be extremely large, indicating a huge capacity gap between the frequency-selective fading channel and the ideal channel. Equation (2) inspires the iCast design. To minimize this gap, we need a subcarrier independent coding scheme. The subcarriers with relatively high h_l s can be utilized for video bits with great importance. While the subcarriers with relatively low h_l s should be utilized for video bits that are not so important, and even sometimes can be excluded after certain transmissions. In this case, we can provide unequal protection without extra redundancy, in the mean while diminish the influence of the deep fading subcarriers with low decodability.

We use a toy example to emphasize the importance of frequency diversity in Fig. 2. The video frames are encoded by Spinal and modulated with OFDM. The five transmitted

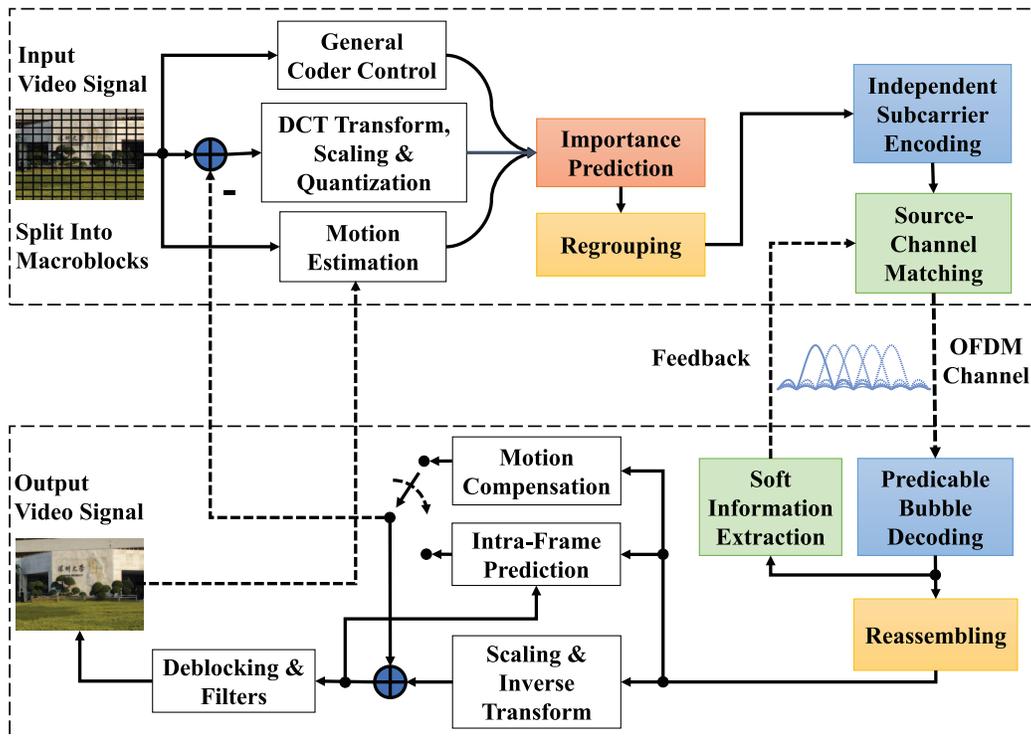


Fig. 3. iCast building blocks. The colored blocks are modifications to the existing video codecs.

subcarriers are experiencing quite different channel conditions. Traditionally, all the subcarriers are used to transmit the same block of symbols sequentially till get decoded, even though the symbols on some subcarriers with high decodability (e.g., subcarrier 14) have been received many times. Unlike the previous methods, iCase aims to breakdown the frequency correlation in rateless code, and utilizes each subcarrier for video streaming and protection in terms of its own decodability. To provide the reader the brief idea of the end-to-end protocol, we outline the operational protocol in iCast as follows. When a pair of nodes has established a video streaming link, they will go through the following steps.

- 1) *Video Source Encoding*: We take the popular MPEG-4 as illustration. As depicted in Fig. 3, each image frame is partitioned into a set of 8×8 macroblocks and fed into the source encoder, including discrete cosine transform (DCT) that computes the frequency content, and quantization that equalizes each macroblock DCT component.
- 2) *Importance-Based Regrouping*: The transmitter leverages DCT along with quantization information to estimate the important level of each bit within a group of picture (GoP), and then regroup the bits into macroblocks in terms of their importance. The regrouped macroblocks are then fed into channel encoder.
- 3) *Subcarrier Independent Channel Encoding*: iCast replaces the last step in MPEG-4, entropy encoding, with its own channel encoder. The regrouped macroblock is encoded into groups of rateless symbols. In each round, multiple macro blocks are transmitted in parallel on all OFDM subcarriers.

- 4) *Predictable Bubble Decoding*: Upon receiving any rateless symbol on a certain subcarrier, the receiver tries to decode the macroblock independently from other subcarriers via an improved bubble decoder.
- 5) *Soft Information-Based Decodability*: After decoding a symbol at each subcarrier, the receiver computes the soft information and feeds back to the transmitter. The transmitter remains a soft information sliding window to keep track of the instant subcarrier-wise decodability.
- 6) *Source-Channel Matching*: The transmitter matches the remaining undecoded blocks with subcarriers at comparable importance for next round transmission.

The foundational notion of iCast is: to intelligently utilize frequency diversity to approach fine-grained data rate and proper protection for video streaming over IoIV. The basic idea is simple and efficient, yet there remains several challenges for implementation. First, in rateless codes, encoded symbols are correlated throughout the entire channel. If we transmit and decode symbols individually on each subcarrier, this correlation should be broken up. Thus, how to achieve subcarrier-independent transmission in a rateless manner brings about a great challenge. Second, how to accurately assess the video bit importance and subcarrier decodability remains a great concern. Third, with bit-wise importance and subcarrier-wise decodability, we need efficient matching algorithm to offer the unequal protection with minimum overhead.

As shown in Fig. 3, iCast offers four components to address the above challenges: 1) subcarrier independent coding that breaks down the correlation in frequency domain and achieves subcarrier independent rateless coding; 2) importance-based regrouping estimate the precise bit-wise importance within a

video frame; 3) soft information-based decodability that harnesses soft information from the PHY to assess the subcarrier-wise importance; and 4) source-channel matching that matches the video bits with subcarriers at comparable importance.

IV. VIDEO TRANSMIT IN ICAST

In standard video encoding (e.g., MPEG-4), images are organized in units of sequences. A sequence is an encoded data stream of a GOPs, and typically consists of eight frames. There are three different types of frames, including I-frame, P-frame, and B-frame. I-frame is an independent picture encoded via other pictures. P-frame contains barely the distinctions from an encoded reference video frame, and B-frame incorporates the previous frame. Each frame is a matrix of pixels. Typically 8-bits value is used to represent a pixel.

In MPEG-4, the original frame is partitioned into a set of 8×8 macroblocks and then passed into the source encoder. The encoding process mainly includes three steps.

- 1) *DCT*: Each macroblock computes their frequency content through a DCT transform. As natural images barely have steep variations, most information are gathering around low frequencies.
- 2) *Quantization*: Second, the resulting DCT components are quantized via a quantization matrix. The quantization matrix can be customized to provide higher resolution over less discernible components.
- 3) *Entropy Encoding*: Finally, the bits are encoded using Huffman encoding and run-length encoding for lossless compression.

iCast makes two modifications to the existing source code. It first adds an importance-based regrouping after quantization to obtain the bit-wise importance within a video frame. Afterward, it replaces the last step with its own subcarrier independent coding.

A. Importance-Based Regrouping

As MPEG-4 generally adopts variable length quantization. The low frequency components are important, and typically quantized into a higher degree of discrete levels. Therefore, they require more bits for representation. Meanwhile, the components that are not that important are quantized into a lower degree of discrete levels with fewer bits. This variable length quantization makes it nontrivial to precisely decide the degree of importance of each bit [25].

Here, we jointly consider the DCT coefficient and quantization information to assess the bit-wise importance. The DCT metric is used to estimate the degree of importance for frequency components within a macroblock. We incorporate mean absolute differences (MAD) [26] in image matching to evaluate the distortion effect of each bit within a GOP. To be specific, we first utilize quantized DCT coefficients to judge the effects of frequency components as follows:

$$\text{Coef} = C(k, n) = \alpha(k) \sum_{n=0}^{N-1} f(n) \cos \left[\frac{\pi}{N} \left(n + \frac{1}{2} \right) k \right] \quad (3)$$

for frequency index $k = 0, 1, 2, \dots, N-1$, and sample index $n = 0, 1, 2, \dots, N-1$, with $\alpha(0) = \sqrt{(1/N)}$ and $\alpha(k) =$

$\sqrt{(2/N)}$ for $k \neq 0$. When $k = 0$, the transform coefficient is the average value of the length- N $f(n)$ sequence which referred to as the dc coefficient. Other transform coefficients are called the ac coefficients.

After DCT transform, each macroblock DCT component is quantified. Obviously, bits from the quantized DCT coefficients contribute a large difference if it is decoded incorrectly. We calculate the MADs to measure the difference if a bit is erroneous

$$\text{Diff} = \frac{1}{M^2} \sum_{i=1}^M \sum_{j=1}^M |E(i, j) - O(i, j)| \quad (4)$$

where M represents $M \times M$ pixels macroblock that recovered from IDCT. i, j refer to the pixels' position in this matrix. $|E(i, j) - O(i, j)|$ depicts the pixel difference between i, j . E represents the macroblock recovered from quantized DCT coefficients with one bit error. O represents the macroblock recovered without errors.

iCast combines DCT coefficients and bit difference to jointly assess the bit-wise importance. Specifically, we normalize the differences and multiply it by corresponding DCT coefficients. The product is the bit-wise importance. The calculation formula is as follows:

$$I = \text{Coef} \times \text{Diff}_{\text{norm}} \text{ for } \text{Coef} \neq 0. \quad (5)$$

Afterward, iCast regroup bits in terms of their importance level I . All bits are sorted in descending order, and then divided into N macroblocks $G = \{G_1, G_2, \dots, G_N\}$, where N is the number of available data subcarriers. The most important bits resides in the headmost group, while the least significant bit in the backmost group. This regrouping information is appended in the header to inform the receiver for reassembling to the original encoding order.

B. Subcarrier Independent Coding

In order to leverage the frequency diversity for fine-grained data rate and bit protection, we need to break up the correlation among subcarriers to achieve a frequency independent rateless coding scheme. Instead of transmitting one block at a time, we transmit multiple blocks on all the subcarriers concurrently.

Once receives a GOP from the source encoder, the channel encoder partitions it into multiple macroblocks, each macroblock M has a length of $(n-16)$. A 16-bit CRC is computed and inserted at the end of each M to construct a link-layer frame. If there are N subcarriers, then N macroblocks are taken as a *batch*, and encoded in parallel on each subcarrier. The encoder on each subcarrier is called *subcarrier independent encoder*, where the basic encoding procedure is abstracted from spinal for illustration.

As shown in Fig. 4, subcarrier independent encoder consists of one hash function h and one random number generator (RNG). Every k bits \vec{m}_i ($\{0, 1\}^k$) are taken from block M , and be encoded into one rateless symbol [20]

$$h : \{0, 1\}^k \times \{0, 1\}^v \rightarrow \{0, 1\}^v \quad (6)$$

$$\text{RNG} : \{0, 1\}^v \times \mathbf{N} \rightarrow \{0, 1\}^c \quad (7)$$

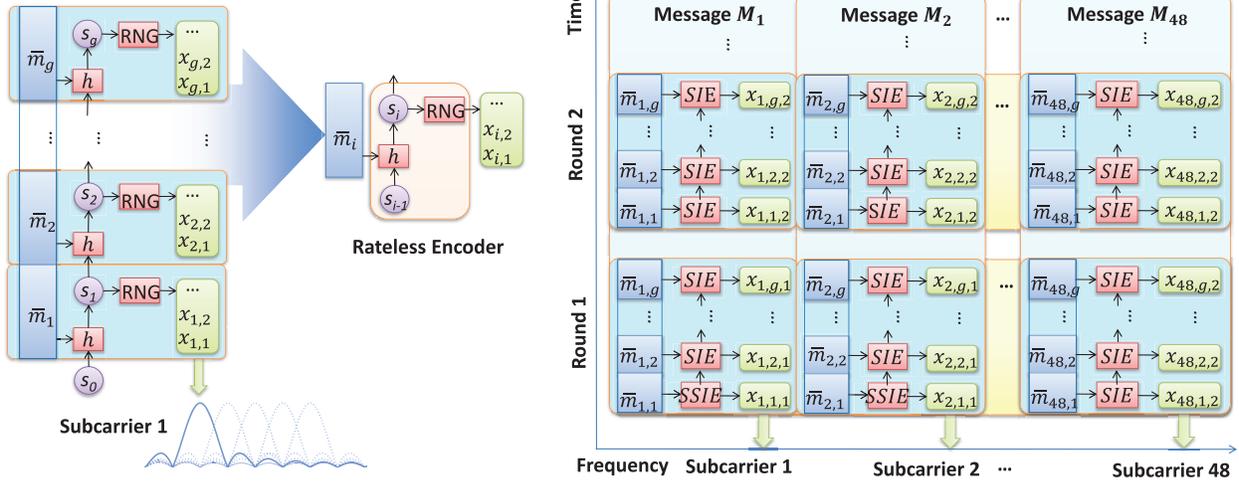


Fig. 4. Subcarrier independent encoder, designed on one subcarrier (left) and processes macroblocks in parallel across the entire channel (right).

where $\{0, 1\}^v$ is a spine value s_i with v -bit, and $\{0, 1\}^c$ is the encoded rateless symbol x_i . Equation (6) uses sequential hashing to build the spine s_i for message bits \bar{m}_i . Afterward, spine s_i is seeded to (7) to generate sequences of rateless symbols $x_{i,j}$ for message bits \bar{m}_i .

Fig. 4 illustrates the encoding and transmission schedule of iCast. A batch of macroblocks is processed concurrently on N subcarriers. After one transmission round, a feedback vector containing the decodability on each subcarrier is received at the transmitter side. The transmitter matches the subcarrier and video bits with comparable importance. This process continues until all the blocks in this batch are successfully decoded. Afterward, the transmitter moves on to the next batch.

V. VIDEO RECEIVE IN ICAST

At the receiver side, it first decodes the received bits using an improved bubble decoder. Bubble decoder [20] is known to be suboptimal, which only maintains V best survivors at each decoding stage instead of searching the entire decoding tree. Therefore, it has a linear decoding complexity when the parameters used in the algorithm are constant. We notice that bubble decoder wastes unnecessary decoding time, which greatly compromises its performance.

A. Predictable Bubble Decoder

According to the Shannon–Hartley theorem, the reliable information rate of a channel has certain limit, which depends on its bandwidth and SNR. Therefore, during the first few transmission rounds, it is impossible for the receiver to successfully decode any of the macroblocks at that time. So it should skip the unnecessary decoding and start decoding from the round at which the achieved data rate is below the Shannon rate. Here, we propose a *predictable bubble decoder* that helps the receiver predicate from which round it should start decoding.

From Shannon–Hartley theorem we deduce that, $k/L < C(\text{SNR})$, where k is the number of message bits encoded

into one rateless symbol, and L is the number of transmission rounds. $C(\text{SNR})$ denotes the Shannon rate under a certain SNR. $C_{\text{AWGN}}(\text{SNR})$ in AWGN channel is

$$C_{\text{AWGN}}(\text{SNR}) = \log_2(1 + \text{SNR}). \quad (8)$$

Then for a certain subcarrier i with a fixed SNR, the number of rounds needed for successfully decoding $L_{\min,i}$ is at least

$$L_{\min,i} = \left\lceil \frac{k}{C(\text{SNR}_i)} \right\rceil, i = 1, 2, \dots \quad (9)$$

We use channel state information (CSI) to estimate the per-subcarrier SNR. CSI is a subcarrier-level channel measurement obtained from two long repeated training sequences in 802.11n preamble. Thus, it is easy to obtain. The SNR on each subcarrier SNR_i is estimated as

$$\text{SNR}_i = \frac{|\hat{\mathbf{H}}(i)|^2}{\sigma_n^2}, i = 1, 2, \dots \quad (10)$$

where $\hat{\mathbf{H}}(i)$ denotes the CSI on the i th subcarrier, and σ_n^2 is the corresponding noise power. For each batch of macroblocks, the receiver first computes the $L_{\min,i}$ on each subcarrier i . It starts decoding only from the $L_{\min,i}$ th round on the i th subcarrier. Note that SNR estimation could be inaccurate due to noise fluctuation. So we increase the estimated SNR by 2–3 dB to acquire a $L_{\min,i}$. $L_{\min,i}$ indicates that we can start decoding a bit earlier to avoid performance loss.

To further improve the data rate, we adopt a puncturing scheme [20] to get higher and fine-grained value. As shown in Fig. 5, in each subround, we only transmit a “skeleton” of the whole macroblock. With the help of the encoding structure, it is possible that receiver can decode the macroblock within a few subrounds. If we divide a transmission round into r subrounds, the minimum number of subrounds needed for successfully decoding $\tilde{L}_{\min,i}$ on each subcarrier is

$$\tilde{L}_{\min,i} = \left\lceil \frac{rk}{C(\text{SNR}_i)} \right\rceil, i = 1, 2, \dots \quad (11)$$

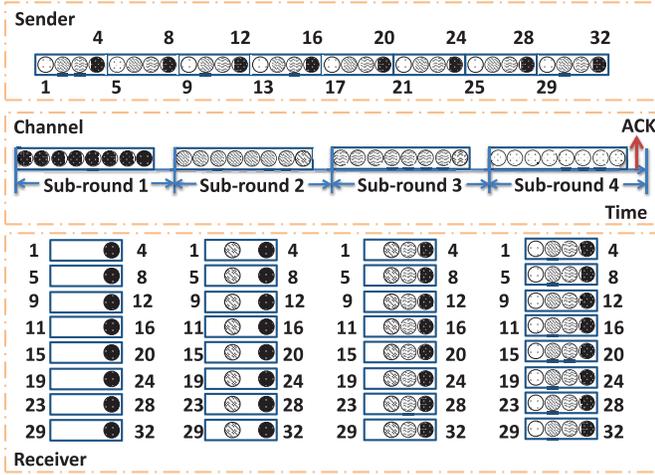


Fig. 5. Illustrated example of puncturing transmission schedule and lightweight feedback.

B. Soft Information-Based Decodability

To assess the subcarrier-wise decodability, iCast incorporates the soft information from the PHY. The soft information, or SoftPHY hints, are posteriori log-likelihood ratios (LLRs) [27] reported from the bubble decoder. It indicates the reliability of a decoded bit. Heuristically, a bit with higher LLR implies high probability to be correctly decoded. Traditionally, LLR is used to help the decoder make decoding decisions. It is calculated from the *maximum a posteriori probability* decoder or *maximum likelihood* [28]

$$\text{LLR}(i) = \frac{2y_i}{\sigma^2}, i = 1, \dots, \hat{N} \quad (12)$$

where $y_i, i = 1, \dots, \hat{N}$ is the bits output from the decoder within a macroblock, and σ^2 is the Gaussian noise variance.

As LLR is fine-grained decoding information from the PHY, we harness it to estimate the subcarrier-wise decodability. The receiver accumulates the LLR within a transmission round. For a certain subcarrier j , the instant decodability $\text{Dec}(j)$ is calculated as follows:

$$\text{Dec}(j) = \frac{1}{\hat{N}} \sum_{i=1}^{\hat{N}} \frac{2y_i}{\sigma^2}, i = 1, \dots, \hat{N}. \quad (13)$$

After each transmission round, the receiver feedbacks the Dec on all the subcarrier to the trainmaster. Upon receiving the feedback, the transmitter sorts in descending order, and matches the video bits with comparable importance. This matching information is also appended in the header to inform the receiver for decoding.

C. Correlated Lightweight Feedback

For wireless communication systems with commercial half-duplex radios, feedback has not yet been well designed. In iCast, the diversity gain mainly results from the fine-grained feedback that allows the transmitter to acquire the instant subcarrier decodability. Therefore, the tradeoff amid the overhead and diversity gain should carefully addressed.

To obtain the instant decodability, a transmitter pauses after each round for the feedback. However, conventional 802.11 ACK is costly due to its exhaustive control overhead. To convey more control information without incurring much overhead, we utilize correlated symbol sequence (CSS) as a *correlated lightweight feedback*. CSS is proved to be efficient and robust as a new paradigm to convey control information [29]. Instead of decoded, CSS is detected merely by correlating it with the incoming samples. Therefore, it avoids the control overhead and reduces the frame duration.

iCast uses CSS to replace the control overhead in 802.11 ACK, including the PHY layer preamble, MAC layer header, and frame check sequence. Each receiver has its distinct CSS, which is predefined and known by the corresponding transmitter. The CSS detection via cross-correlation exempts the usage of preamble, and enables the symbol level synchronization. After CSS, an OFDM symbol is appended as the payload, which contains the subcarrier-wise decodability.

We conduct simple simulations to verify the reliability of the correlated lightweight feedback under a Rayleigh fading channel. The detection rate exceeds 97% when SNR is greater than 3 dB. Accordingly, the false alarm rate is less than 1% and only 3% miss detection rate is caused, which demonstrates that correlated lightweight feedback has a high tolerance of interference and channel fading. However, if miss detection or false alarm occurs due to fading or collision, the transmitter and the receiver will lose the pace of decoding and the performance will be compromised. Fortunately this dissonance is limited to a single batch, which can rely on error detection or correction methods from upper layer. We leave it as future work to address this problem.

The feedback overhead incurred is computed as follows. The length of the CSS is set to 63 symbols, which ensures a low false alarm rate and no miss detection [29]. With a bandwidth of 20 MHz, the transmission time is $t_{\text{CSS}} = (63/20) = 3.15 \mu\text{s}$. The payload is one OFDM symbol. With 64-point FFT, the transmission time is $t_{\text{OFDM}} = (64/20) \times (1 + (1/4)) = 4 \mu\text{s}$. The total time of conveying a Correlated Light Weight Feedback leads to $t_{\text{CSS}} + t_{\text{OFDM}} = 7.15 \mu\text{s}$. Recall that predictable bubble decoder asks the receiver to skip the first few decoding subrounds. Accordingly, we define there is no feedback for the first f subrounds across all the subcarriers, where $f = k/C(35 \text{ dB})$. In this way, the feedback overhead is further reduced. Thus, the latency introduced in iCast is $7.15 \mu\text{s}$ in each subround. Comparing with the 802.11 ACK duration of over $60 \mu\text{s}$ [29], the feedback latency has been greatly reduced.

VI. SOURCE-CHANNEL MATCHING

We first introduce a simple *cyclic allocation algorithm*, which requires no knowledge on the transmitter side. This is suitable for latency sensitive applications. The transmitter and receiver number all the subcarriers in ascending order starting from index 1 to N for initialization. A batch of macroblocks is allocated to the subcarriers from 1 to N in sequence. This schedule is written into the *decoding table* at both the sender and receiver side to keep records. Upon receiving high decodability on a certain subcarrier, the transmitter knows that the block on that subcarrier has been successfully decoded and

Algorithm 1 Source-Channel Matching

-
- 1: Set $S = \{S_1, S_2, \dots, S_N\}$ as N data subcarriers
Set $G = \{G_1, G_2, \dots, G_N\}$ as the N bit groups
 - 2: **for** $i = 1$ to N **do**
 - 3: Calculate the reliability level L_i of S_i subcarrier based on Dec_i
 - 4: **end for**
 - 5: Build unallocated subcarriers table $U = \{U_1, U_2, \dots, U_N\}$ in descending order of L level
 - 6: **for** $i = 1$ to N **do**
 - 7: Allocate Group G_i to subcarrier U_i .
 - 8: Remove assigned subcarrier U_i and assigned G_i bit groups from the table
 - 9: **end for**
-

thus deletes it from the *decoding table*. Afterward, the transmitter vacates this subcarrier, picking up the first macroblock from the *decoding table* and allocates the subcarrier to it. This allocation is conducted cyclically until all the vacated subcarriers are occupied. Algorithm 1 describes the detailed procedure. Since the transmitter and receiver have the same information (decoding status), it is easy for them to keep pace in the transmission schedule.

The *cyclic allocation algorithm* is simple and efficient. To further utilize the frequency diversity for video streaming, especial for quality-sensitive applications, a *source-channel matching* algorithm is proposed, which requires the transmitter keeps record of the bit-wise importance and subcarrier-wise decodability. We formulate the block allocation problem as a *one-to-one matching* algorithm.

Problem Definition: Given a set of macroblocks $\{M_1, M_2, \dots, M_N\}$ and a set of subcarriers $\{S_1, S_2, \dots, S_N\}$, each S_j has a decoding capacity I_j , assign each macroblock to the subcarrier with “comparable” decodability.

We seek to find the most suitable matching for each macroblock. To solve this problem, we get $Dec = \{Dec_1, \dots, Dec_N\}$ on all the subcarriers based on the feedback received. Then, we sort the subcarriers in descending order according to its Dec . Recall that we have already regrouping the bits to $G = \{G_1, G_2, \dots, G_N\}$ based on I value, where the important bits are in the headmost of the sequence, and the bits of low importance are in the backmost. We build two tables for both unallocated subcarriers and unallocated bits. In each round, we allocate the unassigned most important bit group to the most reliable unallocated subcarrier. The matching continues till all the bit groups are assigned.

VII. PERFORMANCE EVALUATION

We evaluate the performance of iCast through experiments and simulations under various environments. Our source encoder is based on MPEG-4/H.264 codec, and the channel encoder is spinal-based PHY layer building on top of 802.11 OFDM modules over GNURadio/USRP2 platform. The USRP2 nodes adopt the RFX2450 daughterboard as RF frontend, which operates in the 2.4–2.5 GHz range. The simulations are interconnected written in C++ and MATLAB.

TABLE I
SIMULATION PARAMETERS

Code	Parameter	Value
LDPC	Message size n	648 bits
	Message size n	128 bits
	Group size k	4 bits
FlexCast	Beam size B	256
	Message size n	128 bits
iCast	Group size k	4 bits
	Survivor size V	256
	Message size n	128 bits

Rateless FlexCast and fixed-rate LDPC codes are selected as competitive under different channel conditions.

A. System Implementation

1) *LDPC:* The source encoder adopts in our LDPC implementation is standard MPEG-4 [13], and the combinations of channel coding and modulation schemes are the same as in 802.11n [30]. The receiver utilizes soft demapper and belief-propagation decoder with 50 full iterations to ensure efficient and accurate decoding. A state-of-the-art rate adaptation technique SoftRate [31] is adopted to ensure that LDPC achieves its best rate adaptation performance. We test all the possible combinations of channel coding and modulation schemes. For each SNR value, we plot the highest rate to demonstrate the best envelope of LDPC code. The detailed parameter is shown in Table I.

2) *FlexCast:* Our FlexCast implementation follows the configurations indicated in [12]. The transmitter replaces the compression in standard MPEG-4 with its own rateless encoder. One important notice is, with OFDM modulation, more than one pass of rateless symbols could be transmitted at a time (e.g., 48 rateless symbols at a time for 64-point FFT), so we use the same parallel transmission structure for FlexCast as in iCast, where 12 macroblocks are multiplexed on one OFDM symbol. Then for each macroblock, four rateless symbols are transmitted per OFDM symbol.

3) *iCast:* We implement iCast as described before. The hash function h and RNG are also based on *one-at-a-time*. We puncture every 32 rateless symbols per round in a macroblock into four subrounds, and each with eight rateless symbols. The allocation algorithm we use is the *one-to-one matching*.

4) *Tested Video:* We choose a standard reference video provided by Xiph.org foundation [32], and store it as a YUV format file as the source input. The video has 300 frames, each with a size of 176×144 pixels. PSNR is selected as the video quality metric. Specifically, we calculate the cross-frame average PSNR. The detailed computation procedure can be found in [33].

5) *Methodology:* In the following scenarios, we use a sender–receiver pair topology, where a sender keeps transmitting packets to a receiver using LDPC, FlexCast, or iCast. Each run of an experiment transfers 2500 packets, which are randomly generated with length of 1460 bytes. We choose achievable data rate as the transmission metric and PSNR as the video quality metric. The channel bandwidth is 20 MHz with 64-point FFT, where 48 subcarriers are used for data transmission, and four serves as pilots.

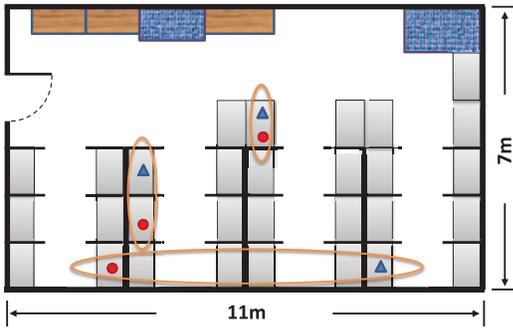


Fig. 6. Experimental environment. Three sets of nodes are distributed in different locations.

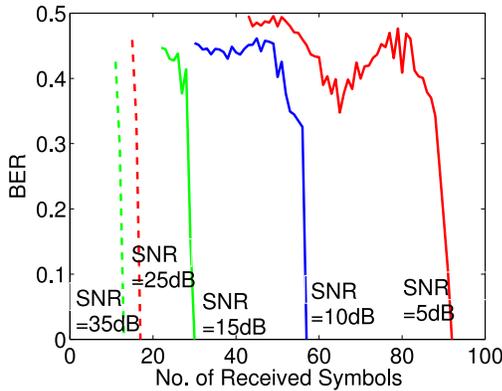


Fig. 7. BER versus number of the received symbol of iCast.

B. iCast Over-the-Air

In this part, we conduct experiment to evaluate the performance of iCast on our GNU radio platform. Fig. 6 illustrates the experimental environment. The aim is to find out that whether our subcarrier independent encoder and predictable bubble decoder is feasible and reliable in practice. Here, we use the two USRP2 nodes topology mentioned before, where one node keeps transmitting packets using subcarrier independent encoder, and the other uses predictable bubble decoder to decode the received rateless symbols without any feedback. The SNR measured at the receiver is about 20 dB.

We first measure the bit error rate (BER) as a function of the number of received symbols under different SNRs. This is to verify that iCast can function well as a rateless code. Fig. 7 shows that, with higher SNR, the decoding speed grows faster. With increasing number of the received symbol, BER drops slightly. This indicates that the receiver are piling up symbols for reliable decoding. Upon accumulating sufficient symbols, BER are quickly close to 0 since the macroblock is successfully decoded, which verifies that iCast uses every single symbol for decoding, and thus can achieve a fine-grained and appropriate data rate at any channel condition.

Fig. 8 depicts the achievable data rates on each subcarrier and the corresponding CSI. The CSI is calculated using 802.11 preamble, which is set as the ground truth to measure the subcarrier quality. We can see that the achievable data rate has distinct value on different subcarrier, which directly relates to the CSI. Specifically, the subcarriers under deep fading (e.g.,

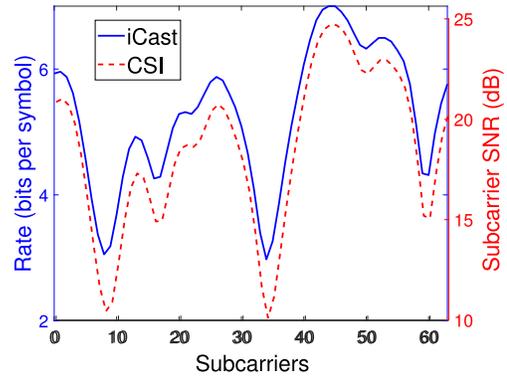


Fig. 8. Performance of iCast over-the-air. The achievable data rate by iCast has directly relates to the baseline CSI.

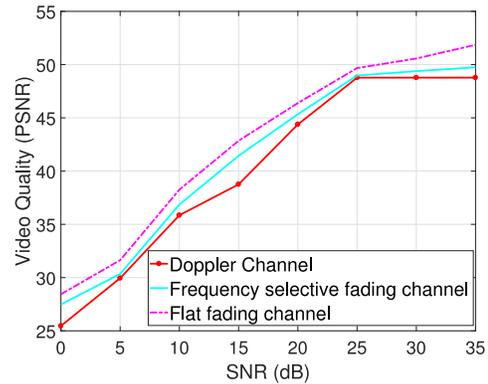


Fig. 9. Video quality achieved by iCast with standard reference video under different channel conditions.

subcarrier 9 and 35) have the lowest rate of 3 bits/symbol, while the subcarrier with the highest CSI (e.g., subcarrier 45) has a data rate larger than 7 bits/symbol. If we use traditional rateless codes, the overall data rate will be restricted by subcarrier 9 and 35, since the rateless symbols transmitted on them will always fail, and may ruin the entire block decoding. On the contrary, independent subcarrier encoder ensures that the transmitted block will not be stuck on certain subcarriers.

C. iCast Under Different Channel Conditions

As USRP2 has latency constraint, we are not allowed to conduct the realtime evaluation under mobile environments. Therefore, we conduct trace-driven simulations to evaluate the performance of iCast compared with LDPC codes and FlexCast. We measure the SNR ranging from 0 to 35 dB with 1 dB interval. The simulations are conducted under four channel models, including AWGN channel, flat fading channel, frequency selective fading channel with/without Doppler effects. We first illustrate the video quality achieved by iCast in Fig. 9 with standard reference video as source inputs. Normally, a PSNR ranging amid [16, 40 dB] is decent, while exceeding 40 dB is excellent quality [12]. The simulation results show that iCast achieves desirable performance under all channel conditions, which stems from its ability to harness frequency diversity. We will present the evaluation details in the following sections. Without loss of generality, we use

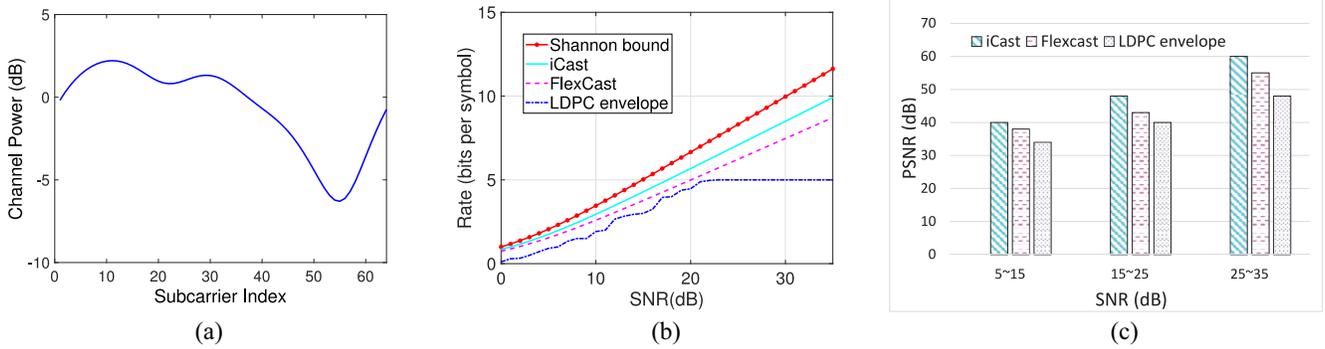


Fig. 10. Flat fading channel. (a) Channel response. (b) Achieved data rate by iCast, LDPC, and FlexCast. (c) PSNR by iCast, LDPC, and FlexCast.

Shannon Bound in AWGN channel as the baseline in both flat fading and frequency selective channel, as it is assumed to be the upper bound in all channel models.

1) *iCast Over AWGN Channel*: In this section, we evaluate the performance of iCast over an AWGN channel, where white noise follows Gaussian distribution, and the energy spreads across the entire channel. Ideally, there is no fading or frequency selectivity. Thus, iCast is expected to achieve a comparable performance with FlexCast.

According to the simulation results, when exceeding a certain threshold, the data rate of LDPC is bound (e.g., 5 bits/symbol for 64QAM with 5/6 coderate). Afterward, it cannot utilize a higher SNR to achieve a higher data rate, since there is no such combination of channel coding and modulation schemes. Conversely, both iCast and FlexCast achieve data rates that are very close to Shannon rate across the entire SNR range. Since when SNR is high, rateless codes can use less time for decoding than fixed rate codes. When SNR is low, rateless codes still take this advantage as temporal noise varies over time. Thus, iCast achieves up to 60% rate gain over LDPC codes. iCast further outperforms FlexCast by between 3% (0 to 10 dB) and 7% (above 20 dB). These gains also benefit from the noise fluctuation on different subcarriers, making important macroblocks decoded earlier than others. iCast takes this advantage to achieve a higher rate.

2) *iCast Over Flat Fading Channel*: In this section, we evaluate the performance of iCast over a flat fading channel. We choose a Rayleigh fading environment in 802.11 channel model from [34]. As shown in Fig. 10, the root-mean-square (RMS) delay δ_t is 25 ns, which is relatively small than sampling rate T_s of 50 ns. Also, the variation of power in the frequency domain does not exceed 15 dB. Therefore, this channel experiences little frequency selectivity.

The performance comparisons among iCast, LDPC, and FlexCast are shown in Fig. 10. Not surprisingly, LDPC achieves lower data rate than in AWGN channel due to fading. Its data rate is still bound at 5 bits/symbol when SNR exceeds a threshold (e.g., 19 dB). Meanwhile, FlexCast performs worse than in AWGN channel, since slight frequency selectivity leads decoding capacities to differ among various subcarriers. Symbols on subcarriers with lower decodability need longer decoding time. So FlexCast has to keep transmitting more rateless symbols on all the subcarriers, even

though not all of them are helpful. Meanwhile, iCast efficiently utilizes frequency diversity. It utilizes the subcarrier with high decodability for important bits in video streaming. Thus, it approaches higher and more appropriate data rate and bit protection on every subcarrier. The results show that iCast outperforms FlexCast by up to 40%, and achieves an average gain of 10% under a typical SNR working range (e.g., 20–30 dB) [35], and a PSNR gain of 4 dB on average.

3) *iCast Over Frequency Selective Fading Channel*: In this section, we demonstrate the performance of iCast over a frequency selective fading channel. This is expected to be the most suitable channel for iCast due to great frequency diversity. We generate the channel according to the model in [34], with a maximum RMS delay δ_t of 80 ns. This is assumed to be deep fading since δ_t is greater than our OFDM symbol duration T_s . The maximum number of paths is determined by δ_t and T_s according to: $p_{\max} = \lceil 10 \cdot \delta_t / T_s \rceil$. Then we have at most 16 paths. For each path, the amplitude and phase are random variables following Rayleigh and uniform distribution, respectively. We generate totally ten channels according to the above parameters, each is repeated ten times. Fig. 11 illustrate the frequency response of one possible channel.

We depict the achieved data rate and PSNR for LDPC, FlexCast, and iCast. As shown in Fig. 11, both LDPC and FlexCast perform quite poorly due to deep fading and frequency selectivity, even SNR has a large value. The reason why FlexCast cannot achieve good performance as in flat fading channel is that, in frequency selective fading channel, strong decodability diversity caused by different channel quality exist. Symbols on the subcarrier with low decodability become the “bottleneck,” which significantly restricts the overall data rate. On the other hand, due to the capacity to harness frequency diversity, iCast achieves a desirable data rate and approaches the Shannon bound across the whole SNR range. As shown in Fig. 11, iCast outperforms FlexCast by up to 90%, and achieves an PSNR gain up to 5 dB under a typical SNR working range. We notice that there is a little bit performance degradation due to some deep fading subcarriers, since iCast may use them for transmission till the end. Therefore, we define a counter i to record the transmission rounds for each macroblock on a particular subcarrier. If i exceeds a certain number (e.g., $\lceil (rk/[C(\text{SNR}_{\text{current}} - 10)]) \rceil$ dB), we abandon this subcarrier and use another one for transmission.

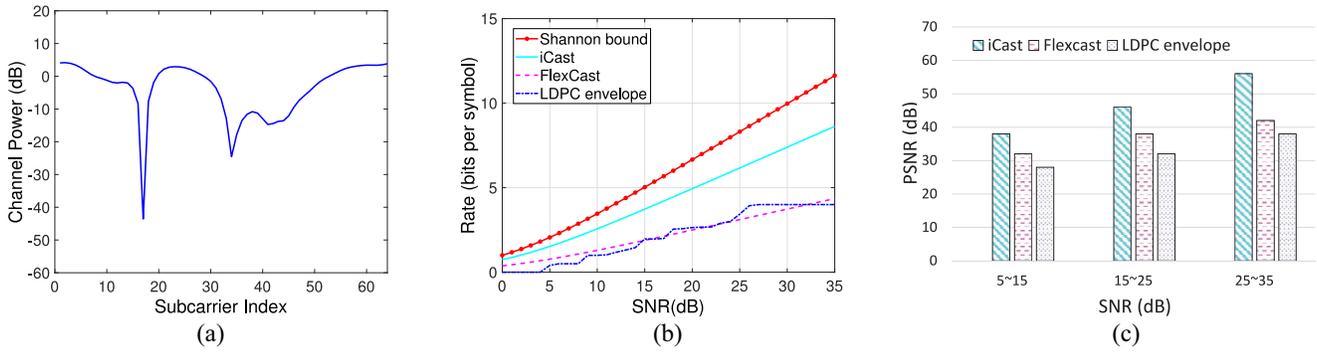


Fig. 11. Frequency selective fading channel. (a) Channel response. (b) Achieved data rate by iCast, LDPC, and FlexCast. (c) PSNR by iCast, LDPC, and FlexCast.

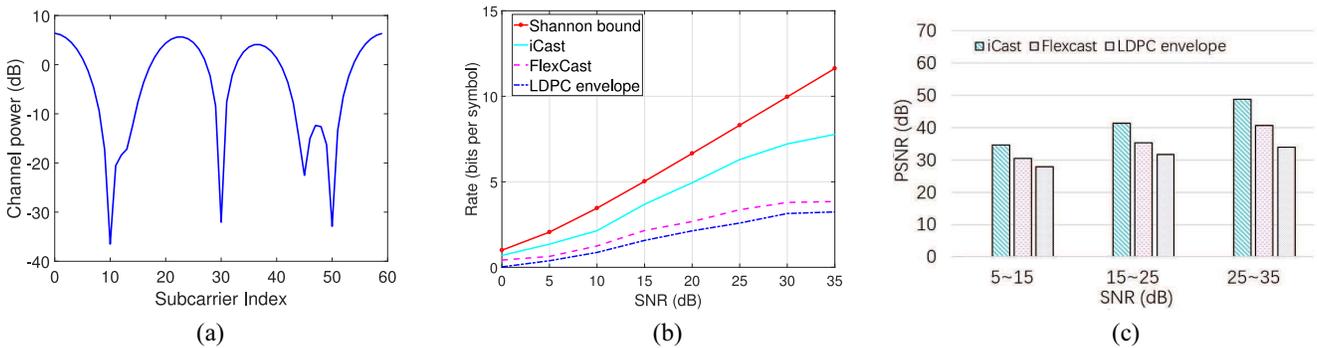


Fig. 12. Frequency selective fading channel with Doppler effects. (a) Channel response. (b) Achieved data rate by iCast, LDPC, and FlexCast. (c) PSNR by iCast, LDPC, and FlexCast.

Also, under some channel conditions, LDPC achieves a little bit higher data rate than FlexCast. This mainly benefits from the forward error correcting (FEC) used in LDPC. Even not all the symbols can be received successfully, LDPC can correct the missing part using FEC. As for FlexCast, since some rateless symbols are essential for decoding (e.g., tail symbol), missing such symbols will definitely degrade its performance.

4) *iCast With Doppler Effects*: In this section, we evaluate the performance of iCast over a multipath fading channel with Doppler shifts, multipath scattering effects, and time dispersion. This is a typical mobile environment, where the above effects rise from the relative motion between the transmitting and the receiving vehicles. We set up the channel according to the model in [34]. The carrier frequency is 2.4 GHz. The relative velocity between the transmitter and the receiver is 20 km/h. The corresponding Doppler frequency shift is 50 Hz. The RMS delay is 25 ns and the sampling time is 50 ns. We generate totally ten channels according to the above parameters, each is repeated ten times. Fig. 12 illustrates one possible channel response.

We depict the achieved data rate and PSNR for iCast, LDPC, and FlexCast, as shown in Fig. 12. All transmissions are hampered by Doppler effects, as there are frequency shifts between the transmitter and the receiver, making the decoding more difficult than in static environment. However, iCast still achieves a desirable performance, especially for the PSNR, which verifies that the fine-grained rate adaptation and frame protection are suitable for IoIV.

5) *iCast With Different Parameters*: Last, we conduct simulations to see how different parameters influence the performance of iCast. Among all the parameters, the number of survivors V in best-survivor decoder and the macroblock size n are critical to the performance of iCast. A larger V leads to more accurate decoding, yet results in higher computational overhead and delay. While a smaller n can better leverage frequency diversity, yet makes the feedback costly. Different communications systems may have various requirements, and thus these parameters may be chosen differently. Here, we investigate them as a design guideline.

It is noted that the number of survivors V during predictable bubble decoding has a tradeoff. Large V can ensure decoding accuracy, but cost a lot a time for decoding due to computational complexity. While smaller V spent much less time for decoding but the accuracy has also been degraded somehow. As shown in Fig. 13, for a system has high computational capacity, we suggest it to choose $V = 256$ to ensure the data rate. Otherwise, $V = 4$ can still achieve a acceptable data rate with less computations. Another key parameter is the macroblock size n . As n becomes smaller, the transmission time for each macroblock becomes shorter, and we can better utilize frequency diversity and adjust to fast varying channel conditions. However, since the feedback costs constant time, the overhead becomes relatively expensive and degrades the overall data rate. Through extensive simulations, we find that n cannot be too large (e.g., 1024) or too small (e.g., 64), among which $n = 128$ ensures a desirable data rate.

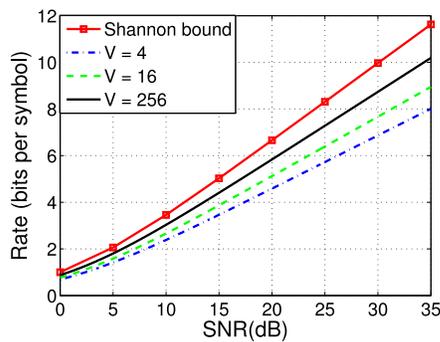


Fig. 13. Performance of iCast with different values of V over AWGN channel.

VIII. CONCLUSION

In this paper, we propose iCast, a fine-grained wireless video streaming scheme that intelligently achieves the most appropriate data rate and frame protection for video traffic over IoIV. We first identify a strong decodability diversity introduced by frequency diversity in high mobility communication systems. With the assist of such diversity, we assess the subcarrier-wise decodability and bit-wise importance, and propose a source-channel matching algorithm to “intelligently” decode the video bits on each subcarrier according to the channel condition, in the meanwhile provide more suitable protection for the video source. Extensive simulation results show that outperforms the state-of-the-art wireless video delivery systems by up to 5 dB PSNR over a frequency-selective channel.

REFERENCES

- [1] F. Yang, J. Li, T. Lei, and S. Wang, “Architecture and key technologies for Internet of Vehicles: A survey,” *J. Commun. Inf. Netw.*, vol. 2, no. 2, pp. 1–17, 2017.
- [2] Cisco Visual Networking Index (VNI). (2014). *The Zettabyte Era: Trends and Analysis*. Accessed: May 29, 2013. [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/serviceprovider/visual-networking-index-vni/VNI_Hyperconnectivity_WP.html
- [3] J. Chen *et al.*, “Narrowband Internet of Things: Implementations and applications,” *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2309–2314, Dec. 2017.
- [4] X. Cheng, C. Chen, W. Zhang, and Y. Yang, “5G-enabled cooperative intelligent vehicular (5genCIV) framework: When Benz meets Marconi,” *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 53–59, May/June 2017.
- [5] X. L. Liu, W. Hu, Q. Pu, F. Wu, and Y. Zhang, “Parcast: Soft video delivery in MIMO-OFDM WLANs,” in *Proc. 18th Annu. Int. Conf. Mobile Comput. Netw.*, 2012, pp. 233–244.
- [6] X. Fan, F. Wu, D. Zhao, and O. C. Au, “Distributed wireless visual communication with power distortion optimization,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 6, pp. 1040–1053, Jun. 2013.
- [7] A. Gudipati and S. Katti, “Strider: Automatic rate adaptation and collision handling,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 4, pp. 158–169, Aug. 2011.
- [8] J. Perry, P. A. Iannucci, K. E. Fleming, H. Balakrishnan, and D. Shah, “Spinal codes,” in *Proc. ACM SIGCOMM*, 2012, pp. 49–60.
- [9] A. Ahmad, N. U. Hassan, M. Assaad, and H. Tembine, “Joint power control and rate adaptation for video streaming in wireless networks with time-varying interference,” *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6315–6329, Aug. 2016.
- [10] C. Stefanovic *et al.*, “Urban infrastructure-to-vehicle traffic data dissemination using UEP rateless codes,” *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 94–102, Jan. 2011.
- [11] P. Cataldi, A. Tomatis, G. Grilli, and M. Gerla, “CORP: Cooperative rateless code protocol for vehicular content dissemination,” in *Proc. 8th IFIP Annu. Mediterr. Ad Hoc Netw. Workshop Med-Hoc-Net*, 2009, pp. 1–7.
- [12] S. Aditya and S. Katti, “FlexCast: Graceful wireless video streaming,” in *Proc. 17th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 277–288.
- [13] A. Puri and A. Eleftheriadis, “MPEG-4: An object-based multimedia coding standard supporting mobile applications,” *Mobile Netw. Appl.*, vol. 3, no. 1, pp. 5–32, 1998.
- [14] M. Ji, G. Caire, and A. F. Molisch, “Wireless device-to-device caching networks: Basic principles and system performance,” *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.
- [15] M. Ji, G. Caire, and A. F. Molisch, “Fundamental limits of caching in wireless D2D networks,” *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.
- [16] Y. Cao, T. Jiang, X. Chen, and J. Zhang, “Social-aware video multicast based on device-to-device communications,” *IEEE Trans. Mobile Comput.*, vol. 15, no. 6, pp. 1528–1539, Jun. 2016.
- [17] C. Long, Y. Cao, T. Jiang, and Q. Zhang, “Edge computing framework for cooperative video processing in multimedia IoT systems,” *IEEE Trans. Multimedia*, vol. 20, no. 5, pp. 1126–1139, May 2018.
- [18] M. Luby, “LT codes,” in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, 2002, pp. 271–282.
- [19] A. Shokrollahi, “Raptor codes,” *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2551–2567, Jun. 2006.
- [20] J. Perry, H. Balakrishnan, and D. Shah, “Rateless spinal codes,” in *Proc. ACM HotNets*, 2011, Art. no. 6.
- [21] B. Tan *et al.*, “Analog coded SoftCast: A network slice design for multimedia broadcast/multicast,” *IEEE Trans. Multimedia*, vol. 19, no. 10, pp. 2293–2306, Oct. 2017.
- [22] H. Rahul, F. Edalat, D. Katabi, and C. G. Sodini, “Frequency-aware rate adaptation and MAC protocols,” in *Proc. ACM MobiCom*, 2009, pp. 193–204.
- [23] C. Y. Wong, R. S. Cheng, K. B. Lataief, and R. D. Murch, “Multiuser OFDM with adaptive subcarrier, bit, and power allocation,” *IEEE J. Sel. Areas Commun.*, vol. 17, no. 10, pp. 1747–1758, Oct. 1999.
- [24] X. Zhao, H. Lu, C. W. Chen, and J. Wu, “Adaptive hybrid digital–analog video transmission in wireless fading channel,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 6, pp. 1117–1130, Jun. 2016.
- [25] I. E. Richardson, *H. 264/MPEG4 Part 10*. Chichester, U.K.: Wiley, 2003.
- [26] S. K. Sahani, G. Adhikari, and B. K. Das, “A fast template matching algorithm for aerial object tracking,” in *Proc. Int. Conf. Image Inf. Process. (ICIIP)*, 2011, pp. 1–6.
- [27] L. Wang *et al.*, “Exploring smart pilot for wireless rate adaptation,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 4571–4582, Jul. 2016.
- [28] J. Hagenauer and P. A. Hoeher, “A Viterbi algorithm with soft-decision outputs and its applications,” in *Proc. Glob. Telecommun. Conf. Exhibit. Commun. Technol. 1990s Beyond GLOBECOM*, 1989, pp. 1680–1686.
- [29] E. Magistretti, O. Gurewitz, and E. W. Knightly, “802.11ec: Collision avoidance without control messages,” in *Proc. ACM MobiCom*, 2012, pp. 65–76.
- [30] IW Group *et al.*, *IEEE Standard for Information Technology—Local and Metropolitan Area Networks—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput*, IEEE 802.11n-2009, 2009.
- [31] M. Vutukuru, H. Balakrishnan, and K. Jamieson, “Cross-layer wireless bit rate adaptation,” in *Proc. ACM SIGCOMM*, 2009, pp. 3–14.
- [32] *Xiph.Org Test Media*. Accessed: Apr. 25, 2006. [Online]. Available: <http://media.xiph.org/video/derf/>
- [33] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, “Optimized transmission of JPEG2000 streams over wireless channels,” *IEEE Trans. Image Process.*, vol. 15, no. 1, pp. 54–67, Jan. 2006.
- [34] Y. Cho, J. Kim, W. Yang, and C. Kang, *MIMO-OFDM Wireless Communications With MATLAB*. Chichester, U.K.: Wiley, 2010.
- [35] M. Souryal, L. Klein-Berndt, L. Miller, and N. Moayeri, “Link assessment in an indoor 802.11 network,” in *Proc. IEEE WCNC*, Las Vegas, NV, USA, 2006, pp. 1402–1407.

Lu Wang received the B.S. degree in communication engineering from Nankai University, Tianjin, China, in 2009 and the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2013.

She is currently an Assistant Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. Her current research interests include wireless communications and mobile computing.

Hailiang Yang is currently pursuing the M.S. degree with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

His current research interests include WLAN and future generation cellular networks.

Xiaoke Qi received the B.S. degree in communication engineering from Nankai University, Tianjin, China, in 2009 and the Ph.D. degree in signal and information processing from the Marine Information Technology Laboratory, Institute of Acoustics, Chinese Academy of Sciences, Beijing, China.

She is with the School of Information Management for Law, China University of Political Science and Law, Beijing, China. Her current research interests include wireless communication, spatial audio, and natural language processing.

Jun Xu received the B.S. degree from Yanshan University, Qinhuangdao, China, in 2005 and the M.E. and Ph.D. degrees in computer architecture from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2005 and 2011, respectively.

He is currently a Research Associate with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His current research interests include artificial intelligence, big data analytics, wireless communication, network management, computer architecture, and mobile computing.

Kaishun Wu received the Ph.D. degree in computer science and engineering from the Hong Kong University of Science and Technology, Hong Kong, in 2011.

He is currently a Distinguished Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He has co-authored two books and published 80 refereed papers in international leading journals and premier conferences. His current research interests include wireless networking, mobile computing, and Internet of Things.