2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks

Continuous Wireless Link Rates for Internet of Things

Haoyang Lu
Department of Electrical and Computer Engineering
University of Pittsburgh
haoyanglu@pitt.edu

Wei Gao
Department of Electrical and Computer Engineering
University of Pittsburgh
weigao@pitt.edu

ABSTRACT

Internet of Things has stringent requirements on the wireless network throughput for timely transmission of the big data being produced. In order to maximize the throughput over dynamic fluctuations of wireless channel quality, current wireless systems adapt the link rate to the instantaneous channel condition, but fail to fully utilize the channel capacity due to the discrete choices of available link rates and the gap between these rates. Instead, in this paper we present vMod, a lightweight and practical solution towards maximum wireless network throughput by redesigning the wireless link rates from *discrete* to *continuous*. The key idea of vMod is to modulate a fractional number of data bits into each symbol by employing the Variable-Length Code (VLC), which is able to statistically yield any link rate. We implemented vMod on software-defined radio platforms. Experiment results demonstrate that under highly dynamic wireless network conditions, vMod greatly improves the WiFi throughput by 30% over a single narrowband link, but incurs only negligible overhead.

CCS CONCEPTS

• **Networks** → *Network protocol design*;

KEYWORDS

Wireless Networks, Continuous Link Rates, Modulation, Variable-Length Code

1 INTRODUCTION

Internet of Things (IoT) computerizes physical objects and integrates them into a networked ecosystem [21], producing an unprecedented amount of data about humans and their surrounding environment. It hence has stringent requirements on the wireless throughput to timely transmit such big data. To meet such demanding requirements, an increasing amount of wireless spectrum has been exploited. For example, higher frequency bands at 5 GHz are exploited to expand the bandwidth of a WiFi channel from 20 MHz to 160 MHz [2] and increase the highest link rate to >500 Mbps. They also allow multiple data links to co-exist through MIMO [30].

Unfortunately, current wireless networks have limited efficiency in utilizing these spectrum resources under dynamic wireless conditions in IoT. Particularly in future smart cities, IoTs being deployed at large public facilities, such as massive transportation systems, shopping malls and museums, may experience serious degradation of wireless signal quality due to the long communication distance or complicated indoor layout. Wireless networks in these scenarios, hence, cannot operate at the highest link rate and instead have to adapt to lower link rates according to the instantaneous channel condition [19, 28, 34, 36]. Such link rate adaptation, however, fails to fully utilize the wireless spectrum due to the *discrete* choices of link rates and the *gap* between these rates: a wireless link operates at the same data rate before the channel quality reaches the requirement of the next higher rate. The network throughput during the meantime, then, remains constant and may be much lower than the momentary channel capacity. For example, 802.11a provides only 8 link rates and suggests to stay at 24 Mbps (16-QAM 1/2) as long as the channel SNR is between 17.7 dB and 21.7 dB [32]. New generations of wireless standards such as 802.11ac, while enabling more link rates with higher orders of channel modulation schemes (e.g., 256-QAM) and code rate options (e.g., 5/6 and 7/8), further enlarge the gap between these link rates to >5 dB [25].

To address this challenge, existing research designs wireless networks to be rateless, i.e., the link rate continuously varies based on the channel condition without requiring channel state feedback. For example, rateless codes allow the sender to keep transmitting encoded data chunks at the highest rate, and the actual link rate is determined by the number of data chunks being transmitted until successful decoding of the entire data message [12, 26, 29]. However, rateless codes improve throughput at the cost of extra communication and computation overhead, which prevent them from being practically applied in resource-constrained IoT. First, a rateless sender consumes more energy to send data, because its wireless radio always stays at the high-power Tx mode [17]. Especially when the wireless link quality degrades, more data chunks are transmitted to deliver the same amount of data, even if many data chunks being transmitted are corrupt and wasted[1]. Second, rateless codes are computationally expensive, because the receiver has to continuously decode incoming data chunks. Even when prototyped over high-end processors with abundant computational capacity, they can only support a throughput up to 12.5 Mbps [16].

In this paper, we envision that a better solution towards maximum wireless throughput in IoT, instead of regardlessly sending and decoding rateless data chunks, is to transform the available choices for link rate adaptation from *discrete* to *continuous*, so that the link rate being chosen is always optimal for any channel condition. We present **vMod** (VLC-based Modulation), a novel modulation technique that realizes such continuous wireless link rates for IoT. Its design has the following important characteristics:

[1]Ordinary wireless senders switch their radios to low-power Rx mode after each transmission and wait for channel feedback. They hence consume less energy to send data.

DOI 10.1109/IPSN.2018.00012

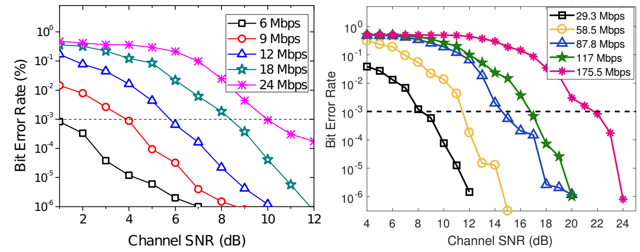
- vMod's modulation is *continuous*. It can modulate an arbitrarily fractional number of data bits into a symbol, while preserving every individual symbol to be independent from others. It can provide any link rate that is supported by the channel condition, and hence maximizes the network throughput under dynamic wireless channel conditions.
- vMod is *lightweight*. vMod does not produce any redundant data transmission, and hence retains the power efficiency of current wireless networks. Besides, vMod's demodulation over each symbol is independent and only involves demapping from a constellation point to data bits. It hence incurs 10x less overhead compared to rateless codes.
- vMod is *applicable* to any commodity wireless system with minimum hardware modification. It builds on the existing QAM modulation and works with link rate adaptation in current wireless networks, and hence incurs minimum modification on the PHY hardware. It does not require modification of any other PHY/MAC hardware or software, such as RF frontend, channel encoder/decoder and link ARQ.

How can a fractional number of data bits be modulated into a symbol that contains an integer number of constellation points? Our key insight is to design a Variable-Length Code (VLC) and split the data bitstream into variable-length codewords, which are then mapped to constellation points in symbols. Hence, each symbol randomly carries a variable amount of data bits, and any link rate can be statistically achieved by adjusting the range and constitution of codeword lengths. For example, by splitting the bitstream with a VLC of 4-bit and 5-bit codewords, we can achieve any link rate between 24 Mbps and 30 Mbps (with a 1/2 code rate), by adjusting the ratio between 4-bit and 5-bit codewords in the VLC. Such granularity of continuous link rates could be further improved by combining vMod with the existing variety of code rate options.

We ensure the generality and power efficiency of vMod by exploiting the modules in existing wireless systems whenever possible. First, we build our VLC by extending the fixed-length code being used in existing systems, and construct new constellation diagrams for modulation by mapping the new variable-length codewords to the unused signal positions in existing QAM constellations. In this way, we ensure that any link rate could be achieved by the same generic approach with the minimum wireless transmit power. Second, information about the lengths of VLC codewords carried by symbols is the key to correct demodulation. To convey such information without impairing throughput, we transmit both regularly modulated and VLC-modulated symbols in the same wireless link, so that such information can be indicated by the permutation of how these two types of symbols alternatively appear.

We implemented vMod over USRP with GNURadio toolkit[2], based on a 5 GHz WiFi transceiver. Minimum modification is conducted over WiFi PHY and MAC. Based on this implementation, we evaluated vMod under dynamic channel conditions, and also compared vMod with existing rateless codes (Strider [12] and Spinal codes [26]). The experiment results show that vMod scales well with the dynamic channel conditions and improves the WiFi throughput by 30% over a single narrowband link, but consumes up to 95% less computation and communication overhead compared to existing rateless codes. vMod, hence, could be easily integrated with other advanced wireless techniques, such as channel bonding and MIMO, to achieve much higher improvement of wireless throughput.

[2]http://gnuradio.org/redmine/projects/gnuradio.



(a) 20 MHz channel (b) 80 MHz channel

Figure 1: Inefficient utilization of WiFi channel capacity

2 MOTIVATION

We motivate the design of vMod by presenting experiment results that demonstrate the inefficiency of existing wireless networks in utilizing the channel capacity. We then discuss the limitations of existing solutions to this problem.

2.1 Inefficient Utilization of Channel Capacity

Discrete choices of wireless link rates and the gap between these rates result in low efficiency of existing wireless networks in utilizing the channel capacity. To measure such gap, we conduct experiments for both narrowband and wideband WiFi channels over USRP N210 SDR boards. In these experiments, we emulate different levels of wireless channel SNR, and measure the receiver's channel status in realtime to compute the corresponding Bit Error Rate (BER). The results in Figure 1 show the relationship between channel SNR and BER when different link rates are used. When the narrowband 20 MHz channel is used, Figure 1(a) shows that 4 dB and 6 dB are the minimum SNRs to support 9 Mbps and 12 Mbps, respectively, in order to reach a BER less than 0.1%. The gap between these two link rates is hence 2 dB, corresponding to 3 Mbps of channel capacity being under-utilized. Similarly, when the link rate increases to 18 Mbps, a gap larger than 3 dB could be observed.

Recent technical advances in wireless networks are incapable of eliminating such gap. For example in Figure 1(b), the latest 802.11ac WiFi standard expands the channel bandwidth to 80 MHz through channel bonding, but also increases the SNR gap between consecutive link rates to be larger than 5 dB. These results highlight that the channel capacity could be better utilized by providing continuous choices of link rates to the gap between existing rates.

Furthermore, new generations of wireless networks, while improving the highest link rate under good channel conditions, make limited contribution to the achievable throughput in practical wireless networks with dynamic channel conditions. To verify this limitation, our experiment measures the WiFi signal strength and the corresponding link rates that are perceived by a Samsung Galaxy S4 smartphone at different locations. For fair comparison, the channel bandwidth is limited to 20MHz and the signal strength over the 5 GHz band is converted to the equivalent number in the 2.4GHz band. The results in Figure 2 show that the link rates provided by different WiFi standards only exhibit noticeable differences when the signal strength is higher than -60 dBm, which may not be achieved in many public wireless networks. Improving the wireless throughput

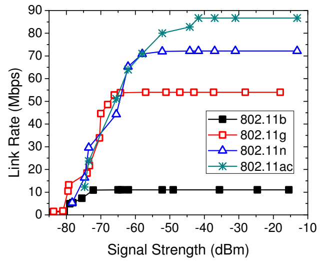
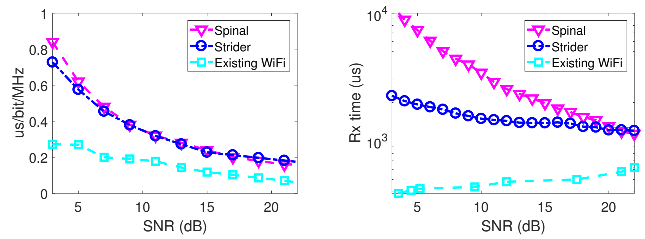


Figure 2: Throughput in practical wireless networks



(a) Tx communication overhead (b) Rx computation overhead

Figure 3: Overhead of rateless codes

in these scenarios, instead, depends on continuous link rates that completely eliminate the gap during link rate adaptation.

2.2 Why Existing Approaches Fail?

The key to continuous link rates is to carry a fractional number of data bits in each symbol. The most intuitive approach to this objective is to break the independency between symbols and convey N data bits with M symbols as a group [24], so that each symbol carries N/M data bits. However, the key limitation of this approach is that the N data bits are demodulated at the same time over the M symbols, and demodulation error of any symbol makes all the N bits to be wrong. Letting p denote the probability of symbol demodulation error determined by channel SNR, the probability for N bits to be correctly demodulated will be $(1-p)^M$, which diminishes when p or M increase. This approach, hence, is infeasible in practice due to its susceptibility to demodulation errors.

Another alternative is to apply multiple existing modulation schemes over the time or frequency domains in the same wireless frame. For example, multiple modulation schemes could be applied to different symbols of a frame. By applying QPSK to 25% of symbols in a frame and 16-QAM to other symbols in the same frame, we are able to convey $2 \times 0.25 + 4 \times 0.75 = 3.5$ bits per symbol on average. However, this approach builds on existing link rates and does not eliminate the gap between these link rates. Hence, it leads to much higher chances of demodulation error. When both QPSK and 16-QAM are used in the same frame, the channel SNR will not be high enough to support 16-QAM and demodulation of the 16-QAM symbols will be certainly prone to channel noise. On the other hand, FARA [27] adopts multiple constellation diagrams over different OFDM subcarriers in a frame by exploiting the frequency diversity of wideband channel. However, the performance of FARA could be easily impaired by inter-carrier interference or strong fading in frequency-selective channels. The large number of OFDM subcarriers also increases the overhead of operating FARA.

Rateless codes are capable of approximating link throughput to the channel capacity, but consume much more power in both communication and computation that is unaffordable by the existing wireless system. In this paper, we evaluate such energy consumption of rateless codes using the Tx communication overhead and Rx computation overhead as the metrics. First, the Tx communication overhead is measured as the average duration of RF waveforms on the air for sending one frame, which is proportional to the energy consumption of the Tx's RF frontend under constant transmit power. Second, the Rx computational overhead is measured by the average amount of time needed for decoding one frame on the receiver side and reflects the power consumption of the Rx's digital logic. To evaluate such overhead, we implemented two rateless code designs, namely Strider [12] and Spinal codes [26], over the GNU-Radio/USRP platform based on the open-source library of wireless system simulations[3]. For the Spinal code, we use 2^{20}-QAM as the constellation diagram and 256 bits and the data chunk size. For the Strider code, we use the recommended setup in the paper, i.e., $K = 33$, and the data chunk size is set to be 9500 bits. At the Tx side, we use the per frame duration on RF as the metric to evaluate the communication overhead, and Figure 3(a) shows that the transmission overhead at a rateless sender is at least 100% higher than that of existing WiFi. Especially when the channel SNR is lower than 10 dB and corrupts many rateless data chunks being transmitted, such difference of overhead could increase to 400%. At the Rx side, we use the per packet decoding time to evaluate the computation overhead, and Figure 3(b) shows that the decoding complexity of rateless code is up to 10x higher than existing WiFi, which requires more powerful chips for real-time decoding. Such high overhead, hence, seriously limits the applicability of rateless codes in practice.

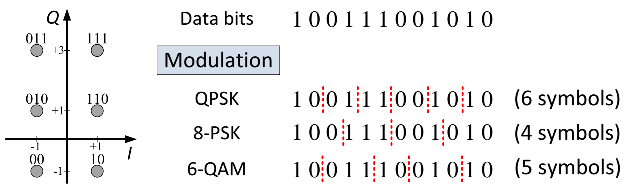
2.3 Variable-Length Code

VLC is widely used in source coding, which maps source symbols to a variable number of bits. By assigning shorter codewords to symbols that appear more frequently, a VLC code requires less bits to represent a certain number of symbols, making it a good solution for lossless data compression. In this paper, we use Huffman code [15] in our vMod design because of the following reasons. First, Huffman code is a classic prefix code with zero redundancy, i.e., no codeword in a VLC is the prefix of another. It hence provides a simple yet reliable solution to VLC codeword construction, by extending the existing fixed-length codewords with negligible computation overhead. Second, by using the Huffman code, vMod requires minimum modification to the existing wireless systems except for channel modulation. Other advanced coding techniques, such as Variable-Length Error-Correcting (VLEC) code [6, 7] and Reversible Variable-Length Code (RVLC) [31], have to be either used for joint source-channel coding or require PHY redesign.

3 OVERVIEW

In order to convey a fractional number of bits in each symbol while preserving independence between these symbols, vMod modulates a data codeword with a variable bit length into each symbol and statistically controls the link rate by adjusting the bit lengths of codewords. Figure 4 shows an example of how such VLC-based modulation is operated over a 6-QAM constellation diagram, to which a VLC of two 2-bit and four 3-bit codewords is mapped as

[3] http://www.yonch.com/wireless.



(a) 6-QAM (b) Modulation

Figure 4: VLC-based modulation

shown in Figure 4(a). Then, as depicted in Figure 4(b), the VLC-based modulation uses five 6-QAM symbols to transmit 12 data bits, and the bit lengths of codewords are either 2 or 3. It hence provides a link rate between the ones provided by QPSK and 8-PSK, which require 6 symbols and 4 symbols to transmit the same amount of data bits, respectively. To realize this high-level design, the following technical challenges need to be addressed.

- First, for efficient modulation, the VLC codewords need to be appropriately constructed and mapped to the constellation diagram. Since the number of codewords in the VLC is arbitrary, the number of constellation points on the constellation diagram may not be an integer power of 2 and the constellation diagram hence needs to be redesigned. How to develop a generic approach that constructs and maps the VLC codewords towards any arbitrary link rate, while preserving the code completeness and retaining the wireless transmit power at the same time, is a challenging problem.
- Second, to minimize the amount of demodulation errors, information about the lengths of VLC codewords in each symbol has to be known by the receiver. Otherwise, if the received signal is distorted by channel noise and erroneously demodulated to another codeword with a different length, it will displace all the succeeding data bits and result in cascading failures. For example in Figure 4(b), if the first codeword "10" is demodulated as "110", the next codeword "011" will be placed to the 4th position of bitstream and result in additional errors. The major challenge, then, is how to convey such information without impairing throughput.

vMod addresses the first challenge by expanding the QAM-based constellation diagrams and fixed-length codewords being used in existing wireless systems for VLC-based modulation. For a given VLC with m codewords, we construct its constellation diagram by adding additional constellation points to an existing QAM constellation diagram with the size smaller than but closet to m. For example, the 6-QAM constellation diagram in Figure 4(a) is constructed from QPSK by adding two additional points. Then, we build our Huffman code by appending bits to some of the existing fixed-length codewords, and map the generated VLC codewords to the additional constellation points being added to the constellation diagram. For example in Figure 4(a), the four 3-bit codewords are expanded from QPSK codewords "01" and "11". In this way, vMod designs new constellation diagrams for different choices of continuous link rates in a generic manner by exploiting the symmetry of QAM, and continuously varies the link rate by adjusting the amount of fixed-length codewords being expanded. More details about such modulation are provided in Section 4.

To address the second challenge, vMod does not transmit the information of codeword lengths through frame preambles or data symbols, to avoid any non-negligible throughput loss. Instead, we embed such information into the pattern of how different modulation methods are used in a symbol. For each symbol modulated by OFDM, we only apply VLC-based modulation to half of its subcarriers and modulate the other half of subcarriers using the regular method in existing systems. Then, the information of VLC codeword lengths being used in this symbol is indicated as the permutation of how these two types of subcarriers alternatively appear in the symbol. During demodulation, this information can be retrieved based on the measurable difference between VLC-based and existing QAM-based constellation diagrams being used. More details about conveying such information are provided in Section 5.

Table 1: Continuous link rates with two VLC codeword lengths in the Huffman code

l_{avg}	Number of k-bit codewords (n_k)					Link rate (Mbps)
	n_2	n_3	n_4	n_5	n_6	
2	4	-	-	-	-	12.0
2.25	3	2	-	-	-	13.5
2.5	2	4	-	-	-	15.0
2.75	1	6	-	-	-	16.5
3	-	8	-	-	-	18.0
3.25	-	6	4	-	-	19.5
3.5	-	4	8	-	-	21.0
3.75	-	2	12	-	-	22.5
4	-	-	16	-	-	24.0
4.25	-	-	12	8	-	25.5
4.5	-	-	8	16	-	27.0
4.75	-	-	4	24	-	28.5
5	-	-	-	32	-	30.0
5.25	-	-	-	24	16	31.5
5.5	-	-	-	16	32	33.0
5.75	-	-	-	8	48	34.5
6	-	-	-	-	64	36.0

4 VLC-BASED MODULATION

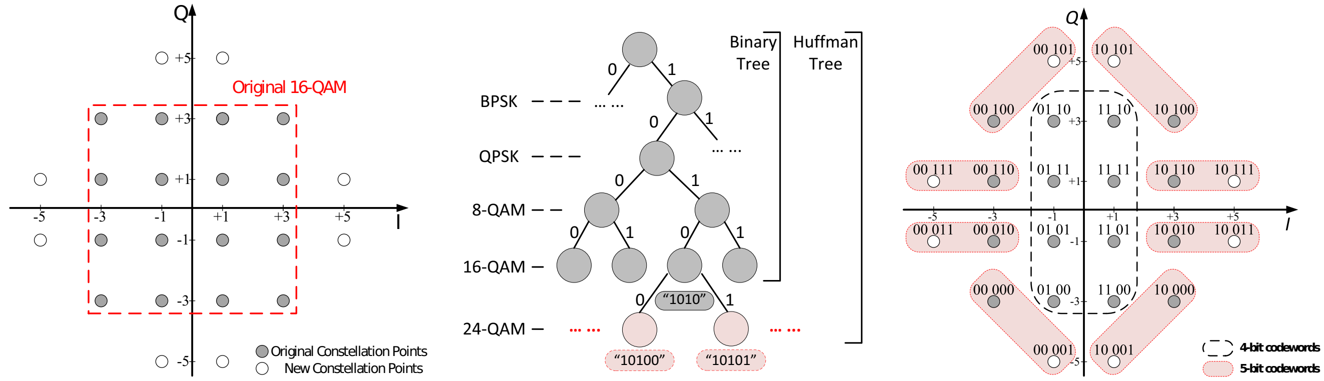
In this section, we present technical details of our proposed VLC-based modulation. We first present the constitution of continuous link rates enabled by the VLC, particularly, the Huffman code. Then, we describe how to realize such continuous link rates by constructing the Huffman code and the corresponding constellation diagram, with an end-to-end example shown in Figure 5.

4.1 Constitution of Continuous Link Rates

In vMod, we construct a Huffman code and map its codewords to the points in a constellation diagram. A link rate is then constituted by the codewords with different lengths that are actually used in modulation. Assuming that each bit in the input data has the equal probability to be "0" or "1", the probability for an l-bit Huffman codeword to be used is 2^{-l}. Then, the average number of data bits being modulated by a constellation symbol is

$$l_{avg} = \sum_{i=1}^{M} p_i l_i = \sum_{i=1}^{M} 2^{-l_i} l_i, \tag{1}$$

where M is the number of codewords in the Huffman code, and l_i and p_i are the length and probability of the i-th codeword, respectively. For any Huffman code, we always have $\sum_{i=1}^{M} p_i = 1$ to ensure the completeness of the code.



(a) Constellation diagram design of 24-QAM based on 16-QAM

(b) Construction of Huffman codewords

(c) Mapping Huffman codewords to 24-QAM

Figure 5: An example of VLC-based modulation: realizing the link rate of 27 Mbps with 24-QAM and $l_{avg} = 4.5$

As a result, we are able to continuously realize any arbitrary link rate by varying the number of codewords and the lengths of these codewords in the Huffman code. For example, when the codewords in the Huffman code have only two different lengths, i.e., k bits and $k+1$ bits, the code is always complete when

$$n_k 2^{-k} + n_{k+1} 2^{-(k+1)} = 1, \qquad 0 \leq n_k \leq 2^k \tag{2}$$

where n_k and n_{k+1} are the numbers of k-bit and $(k+1)$-bit codewords, respectively. Furthermore, we have

$$\begin{aligned} l_{avg} &= k n_k 2^{-k} + (k+1) n_{k+1} 2^{-(k+1)} \\ &= k + 1 - n_k 2^{-k}, \qquad 0 \leq n_k \leq 2^k \end{aligned} \tag{3}$$

By controlling the values of n_k and n_{k+1}, we can achieve different values of l_{avg} with an interval of 2^{-k}. Table 1 lists 17 intermediate link rates that can be realized between the 5 existing 802.11 a/g link rates[4], with a fractional value of l_{avg} (the 5 existing link rates correspond to integer values of l_{avg}). To further reduce the gap between link rates, more options of channel code rates and a larger variety of VLC codeword lengths could be incorporated.

4.2 Constellation Diagram Design

Based on the choice of link rate, the key problem of realizing this link rate is then how to construct the Huffman code and the corresponding constellation diagram, to which the codewords will be mapped. The number of constellation points in the constellation diagram has to be equal to the number of codewords in the Huffman code. When l_{avg} is fractional, the number of constellation points is not an integer number of the power of 2. Hence, a new constellation diagram needs to be designed for every different value of l_{avg}. For example when $l_{avg} = 4.5$, a constellation diagram with 24 constellation points is needed according to Table 1.

Our primary goal is to design new constellation diagrams for different values of l_{avg} in a generic manner, so that different choices of continuous link rates could be supported by the same wireless hardware. To reach this objective, our basic idea is to design the required constellation diagram by expanding an existing QAM constellation diagram with the size smaller than but closet to the size of Huffman code. For example, for the Huffman code with a size of 24, we construct 24-QAM from 16-QAM used in existing wireless systems. Generality of designing new constellation diagrams, in this way, is then ensured by the symmetry of QAM constellations[5].

Note that, for each value of l_{avg}, there may be multiple ways to design the new constellation diagram from an existing QAM constellation diagram. To minimize the average transmit power, as illustrated in Figure 5(a), we place the additional constellation points to the unused signal positions with the smallest amplitude in the complex plane. If there are multiple choices with the same level of average transmit power, the one that maximizes the symmetry of constellation diagram will be chosen. For example, among the two choices of 6-QAM shown in Figure 6, the one with better symmetry in Figure 6(b) will be chosen.

4.3 Codeword Construction and Mapping

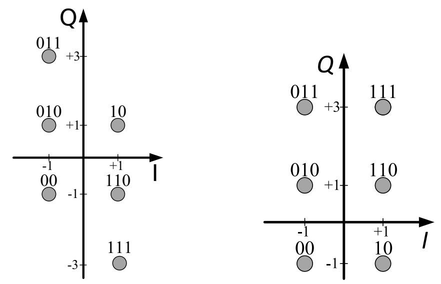
Being similar to our constellation diagram design, we construct the Huffman codewords based on the fixed-length codewords in existing wireless systems. Any 2^K-QAM constellation diagram corresponds to a set of 2^K fixed-length codewords, which can be represented by the leaf nodes of a full binary tree with a height of K. Our approach, then, constructs the Huffman tree by appending new leaf nodes to the corresponding binary tree of fixed-length codewords.

More specifically, two new leaves are appended to a leaf node in the binary tree, hence replacing one K-bit codeword with two $(K+1)$-bit codewords. The amount of leaf nodes in the binary tree being appended depends on the value of l_{avg} and the constitution of VLC codewords. For example, when $l_{avg} = 4.5$, according to Eq. (3) and Table 1, 16 5-bit codewords are needed in the VLC and hence 8 leaf nodes in the binary tree of 16-QAM need to be appended. As shown in Figure 5(b), by appending two new leaves to the leaf "1010" of the binary tree, two 5-bit codewords ("10100" and "10101") are generated. The remaining 4-bit codewords in the binary tree, on the other hand, are also used as Huffman codewords. It is straightforward to prove that the set of Huffman codewords generated by this approach is complete.

The remaining problem, then, is how to determine the leaves in the binary tree to be appended. Principally, we choose to append

[4] A fixed channel code rate of 1/2 is used.

[5] Other constellation shapes such as triangular [10] constellations, while providing better power efficiency, are generally asymmetric. Hence, new constellation diagram for each value of l_{avg} has to be designed in a distinct way implemented with specialized hardware.



(a) Asymmetric design (b) Symmetric design

Figure 6: Design choices for 6-QAM

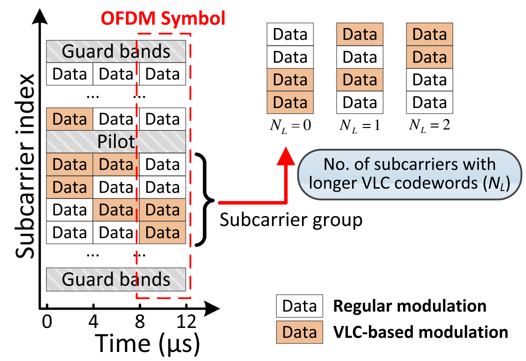


Figure 7: Conveying information of VLC codeword lengths. Every four subcarriers in a symbol forms a group. VLC codewords only have two different lengths.

the leaf nodes that correspond to the constellation points with the largest amplitude in the QAM constellation diagram. For example in Figure 5(c), the 4-bit codewords corresponding to the 8 constellation points at the two outer sides of the 16-QAM constellation diagram are appended to generate 5-bit codewords. These 5-bit codewords are then mapped to the same set of constellation points of 16-QAM, as well as the new constellation points added for 24-QAM.

Note that the constellation diagrams, Huffman codewords and their mappings between each other are completely constructed offline. During the online modulation and demodulation, the wireless system converts the data codewords and constellation symbols between each other based on the pre-defined mapping, and hence incurs negligible computational overhead on both Tx and Rx.

5 VLC-BASED DEMODULATION

In principle, vMod demodulates data by demapping the received symbols from the constellation diagrams developed in Section 4.2. However, due to the variable lengths of VLC codewords, the information about the lengths of VLC codewords being modulated in a symbol is crucial to the correctness of VLC-based demodulation. As described in Section 3, with such information, erroneous demodulation of any signal will not affect demodulation of others and the accuracy of demodulation is then linearly proportional to the level of channel noise. In this section, we describe how to convey such information without consuming any additional wireless spectrum or impairing any data throughput.

First, to minimize the amount of data sent, we do not send the numbers of VLC codeword lengths directly in data symbols. Instead, we partition the subcarriers within a symbol into equally sized groups and convey the required information with respect to each group, i.e., the numbers of VLC codewords with different lengths being used in each group. These numbers may be different over multiple groups according to the input bitstream.

Then, to send such information to the receiver without impairing data throughput, we only apply VLC-based modulation to some of the subcarriers in each group and modulate other subcarriers in the group using the regular modulation scheme in existing systems. For example in Figure 7, when the codewords in the Huffman code only have two different lengths, the number of longer VLC codewords (N_L) in each group is represented by how these two types of subcarriers in the group are permuted in the frequency domain. In particular, for a group of N subcarriers[6], the value of N_L is ranged between 0 and $N/2$, and each possible value of N_L corresponds to a specific permutation of the two types of subcarriers. It is easy to see that the number of possible permutations always exceeds the number of possible values of N_L. When more types of VLC codewords are involved in the Huffman code, larger groups of subcarriers and more complicated permutation of subcarriers will be needed to convey the extra information about codeword lengths.

To convey such information, a naive approach is to embed the value of N_L into the packet preamble or header, which can be represented by at least $\lceil \mathrm{Log}_2(N/2+1) \rceil$ data bits. However, this would significantly hurt the data throughput. For example, 4 data bits are required to represent N_L when $N = 24$, which translates to $1/6$ and $1/12$ throughput loss with BPSK and QPSK modulation, respectively[7]. Instead, our approach seeks to embed this information without throughput loss, and minimize the size of control information being embedded into the frame header.

Based on this design, we further solve two technical problems: 1) How could the demodulator decode the conveyed information by recognizing whether the received signal in a subcarrier is VLC-modulated? 2) How could the demodulator ensure the correctness of such information, which may be distorted by channel noise?

5.1 Recognition of the Received Signal

The demodulator determines whether a subcarrier is VLC-modulated based on the measurable difference between the constellation diagrams used for regular modulation and VLC-based modulation.

In practical WiFi systems, in order to maintain the transmit power to be constant, the power of every constellation diagram needs to be normalized before being used for modulation. Therefore, although the constellation diagram being used for VLC-based modulation is an expansion of an existing QAM-based constellation diagram (e.g., the 24-QAM constellation diagram is an expansion of 16-QAM as shown in Figure 5(a)), the positions of its constellation points are changed due to power normalization. For example in Figure 8 which shows parts of the 16-QAM and 24-QAM constellation diagrams in the first quadrant, although the constellation points with smaller amplitudes in 24-QAM are directly inherited from the ones in 16-QAM, their positions in 24-QAM are different from those in 16-QAM. Our measurement studies show that the power ratio of

[6]For simplicity, the value of N is always an even integer.

[7]Assuming that the same code rate is applied over preamble and data bits.

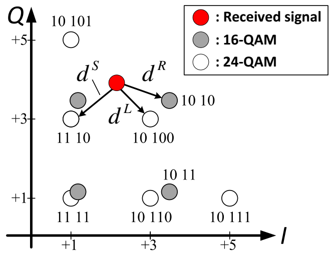


Figure 8: Recognition of the received signal

such difference could be as large as 0.165 under the unity power constraint, corresponding to more than 10 dB in transmit power that can be reliably measured by commodity transceivers.

Based on such difference, when we overlap the regular and the VLC-based constellation diagrams together, we map the received signal to the nearest constellation point on this overlapped diagram. Then, we can tell whether this constellation point belongs to the VLC-based constellation diagram. Further, according to the mapping between VLC codewords and constellation points in Section 4.3, we can decide whether this constellation point refers to a longer VLC codeword. As shown in Figure 8, we measure d^L and d^S as the distance from the received signal to the nearest VLC-based constellation point with shorter and longer codewords, respectively, and measure d^R as its distance to the nearest regular constellation point. If $\min\{d^L, d^S\} < d_R$, the demodulator considers the corresponding subcarrier as VLC-modulated, and in this case if $d^L < d^S$, the received signal is considered to carry a longer VLC codeword.

5.2 Resistance against Channel Noise

With the VLC-modulated subcarriers being recognized, the numbers of VLC codewords with different lengths can be retrieved according to the permutations of subcarriers in the symbol. However, in practical wireless networks where the channel is always noisy, if the type of any one subcarrier in the group is incorrectly recognized, the retrieved information about VLC codeword lengths will be wrong and prevent correct demodulation of all subcarriers.

To minimize such risk and resist against channel noise, we encode permutations of subcarriers using the Hamming code, so that the error on recognizing the received signal in any single subcarrier could be corrected. For each group with a size N, we indicate the permutation of subcarriers in the group as a N-bit binary string: “0” indicates that the corresponding subcarrier is regularly modulated and “1” indicates a VLC-modulated subcarrier. For example, when the VLC codewords only have two different lengths, every possible value of N_L corresponds to a Hamming codeword with an equal number of 0s and 1s. All the codewords in the (8,4,4) Hamming code, except the two codewords of all 0s and all 1s, can be used for permutations of subcarriers. At the receiver, if the numbers of recognized regular and VLC-modulated subcarriers are not equal, the error could be easily corrected by a Hamming decoder.

As discussed in Section 4.1, using more ($N > 2$) codeword lengths could further increase the granularity of wireless link rates. For example, for the VLC of 24-QAM shown in Figure 5(c), we could further increase the average code length from 4.5 bit to 4.53125 bit by splitting one 5-bit codeword into two 6-bit codewords, which provides extra granularity to the codes that consist of only 4-bit and 5-bit codewords. In this case, the total number of bits within one group is not only decided by the number of the longest codeword that occurs, but also by that of the other $N - 2$ codeword lengths. For example, a group of 3 subcarriers carrying 15 bits can be represented as three subcarriers carrying 4-bit, 5-bit and 6-bit codewords, respectively, or three 5-bit subcarriers. In this case, our proposed technique could be easily extended to convey the additional information about more VLC codeword lengths, by adopting more complicated permutation patterns of subcarriers.

6 IMPLEMENTATION

We implemented vMod over USRP SDR boards with the GNURadio toolkit, which realizes a WiFi transceiver over the 5GHz frequency band. vMod requires the minimum amount of modifications over existing WiFi PHY by revising only the modulation and demodulation modules. Same as WiFi, the link rate of vMod frames is specified by the transmitter according to the instantaneous channel estimation, which is conducted by the rate adaptation algorithm. A specific OFDM symbol is appended to the frame header to indicate the vMod link rate, as well as other parameters required for decoding vMod frames.

6.1 PHY-Layer Implementation

The architecture of the PHY-layer system implementation is shown in Figure 9, which is based on the existing WiFi PHY with the newly added modules highlighted in color. Our design requires the minimum amount of modification on the WiFi modulation and demodulation, retaining other modules of WiFi PHY intact.

At the transmitter, we add a new VLC Symbol Mapper, which uses VLC-based constellation diagrams developed in Section 4.2, to be operated in parallel with the Regular (REG) Symbol Mapper in the existing WiFi systems. The input datastream, after channel encoding, is distributed to the two mappers, which both produce signal symbols to convey the information of VLC codeword lengths as described in Section 5. More specifically, the bitstream divider first passes the required number of data bits being used in the regular constellation diagram to the REG Symbol Mapper. Afterwards, it keeps passing the following data bits to the VLC Symbol Mapper until the required amount of signal symbols are generated. The outputs of these two mappers are merged to the Symbol Permutator, which embeds the information of VLC codeword lengths as described in Section 5.2.

At the receiver, the received RF signal will be first dissembled to OFDM symbols by FFT and recovered from channel distortion by the Frequency Domain Equalizer. Afterwards, the Permutation Detector detects the permutation pattern over symbols to retrieve the information of VLC codeword lengths in the VLC-modulated symbols. It also reorganizes the signal symbols and passes the symbols to the REG and VLC Symbol Demapper, respectively. Finally, the Viterbi decoder decodes the demodulated bitstream.

This implementation over USRP also has limitations. At the physical layer, we estimate the channel response and equalize the signals in the frequency domain. However, compared to commodity WiFi devices, the performance of the RF receiver at the USRP board is restricted due to the absence of automatic gain control (AGC) which cannot be emulated or implemented in the GNURadio software. This AGC, instead, shall be further implemented by customizing the FPGA core in the USRP hardware.

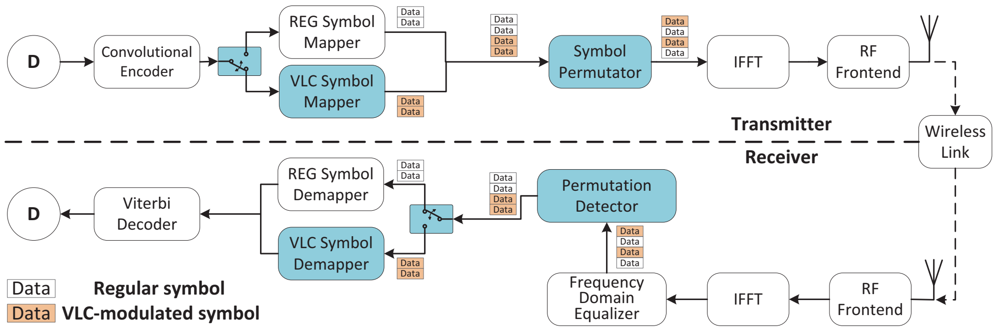


Figure 9: PHY-layer vMod implementation

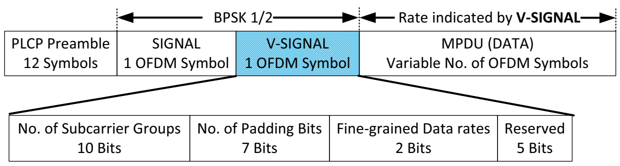


Figure 10: Modified frame format

6.2 MAC-Layer Implementation

We implemented a complete WiFi MAC layer with functionalities such as carrier sensing and acknowledgements [13]. However, since the RF signals harvested by USRP have to be sent to and processed at the host PC, a long delay is incurred at signal transmission and processing. It is hence difficult to meet the timing requirements of the Distributed Coordination Function (DCF) in WiFi. To address this challenge, we follow the same method in [13] and use a scaling factor $\beta(= 5000)$ to emulate the real WiFi MAC.

vMod can work with any existing rate adaptation algorithms, by providing them continuous choices of link rates to select from. We incorporated two mainstream rate adaptation algorithms, namely Minstrel [4] and Adaptive Auto Rate Fallback (AARF) [19], into our implementation to emulate the realistic behaviors of commodity wireless devices in practical scenarios. Minstrel is widely used in commodity systems based on past statistics of packet reception, and has been integrated as part of Linux OS kernel. AARF adaptively adjusts the link rate based on recent consecutive successes or failures of frame transmissions. This implementation will also be used in our real-world experiments in Section 8.

Our implementation also contains a modified format of WiFi frame. In frame assembly of WiFi, the number of OFDM symbols and padding bits used for a payload is only determined by the link rate and payload length, and is independent of payload content. However, this independency is invalid in vMod because of the usage of VLC. VLC-based modulation brings uncertainty to the codeword lengths and furthermore the number of OFDM symbols being used, and this uncertainty is related to payload content. To address this uncertainty, we insert one additional OFDM symbol, namely V-SIGNAL, between the SIGNAL symbol and MPDU symbols in a WiFi frame. As shown in Figure 10, this symbol carries information about the number of subcarrier groups and padding bits. It also uses two additional bits, along with the 4-bit RATE field in the existing SIGNAL symbol, to indicate the link rate being used. Being similar to SIGNAL symbol, the V-SIGNAL symbol is transmitted in the lowest link rate. Afterwards, MPDU is transmitted using the actual link rate declared in V-SIGNAL.

7 PERFORMANCE EVALUATION

Built on our implementation in Section 6, we evaluate the performance of vMod by comparing it with existing WiFi systems and rateless code designs. Our experiment results show that vMod is effective to fill the gap between existing discrete link rates. We also show that the communication and computation overhead of vMod is comparable to that of the existing WiFi, and is up to 10x lower than that of rateless codes. These results, then, ensure that vMod could be practically utilized to improve the network throughput in various severe wireless network scenarios.

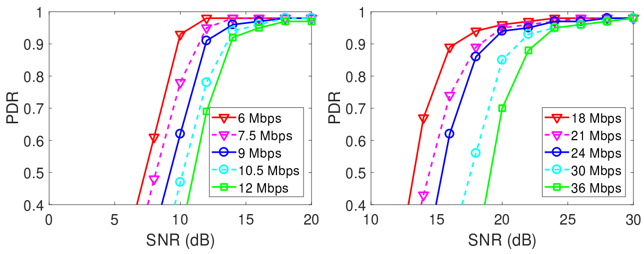
7.1 Experiment Setup

We conduct our experiments in a 10m × 10m open space office using two USRP N210 motherboards with UBX-40 RF daughterboards: one USRP continuously transmits 250 frames with random payloads of 300 bits, and another USRP decodes the frames and counts the ones that are received properly. Due to the MAC scaling factor β, the interval between two consecutive transmissions is around 0.4 seconds. We tune the channel SNR by adjusting the USRP Tx Gain between 2 dB and 30 dB, and the maximum Tx power is equivalent to 13 dBm. Each USRP is connected to a Dell desktop PC, which is equipped with an Intel i5-3475S CPU 2.9GHz and 8GB memory. The communication and computation overheads of transmission are then measured at the host PCs of both sender and receiver.

We use the same implementations of rateless codes as in Section 2.2. Ideally, the rateless sender continues to send new data chunks without channel estimation or feedback, until the receiver successfully decodes the packet. However, when being implemented over GNURadio, if the receiver misses a data chunk because of failing to detect the packet rather than failing the packet CRC test, it will prolong the decoding process or even fail. Hence, we set the receiver to acknowledge reception of chunks in order. The receiver would pause further reception of chunks till a chunk with correct sequence number is received. Once the receiver successfully decodes a packet, it sends a *ACK_FOR_PACKET* frame to the sender.

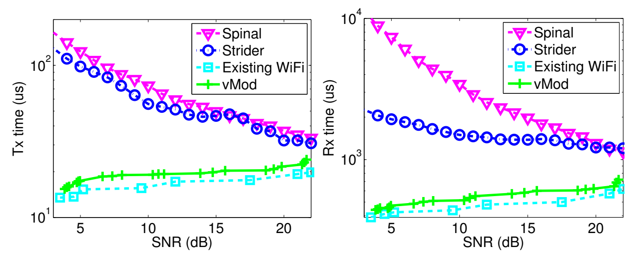
7.2 Packet Delivery Ratio

We first evaluate the Packet Delivery Ratio (PDR) of the continuous link rates under different channel SNR conditions. In order to achieve the same level of PDR, a better channel condition with higher SNR is required for a higher link rate. As shown in Figure 11



(a) Low SNR conditions (b) High SNR conditions

Figure 11: Packet delivery ratio (PDR) under different channel conditions. The new and existing link rates are displayed in dotted and solid lines, respectively.



(a) Tx overhead (b) Rx overhead

Figure 12: Computational overhead

which contains cases of both low and high channel SNR conditions, for any given SNR condition, the PDR of the intermediate continuous link rates always lies between that of their adjacent existing link rates. The PDRs of both categories of link rates eventually converge to 1 when the channel SNR condition is sufficiently good.

These results show that vMod is very effective to fill the gap between existing link rates and hence provides better choices for dynamic channel conditions. For example, when the channel SNR is 17 dB and the PDR is required to reach 90%, the link rate of 21 Mbps could be used to replace the existing link rate of 18 Mbps, yielding 16.7% improvement of throughput. Such improvement of data throughput in practical wireless scenarios with dynamic channel conditions will be further investigated in Section 8.

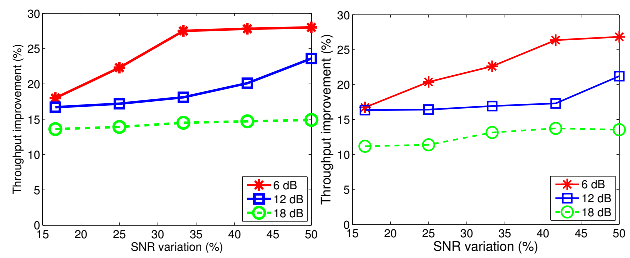
7.3 Overhead

We evaluate the Tx communication overhead and Rx computation overhead caused by vMod operations, which also measure the power consumption of vMod as explained in Section 2.2. The communication overhead at the wireless sender for sending a unit amount of data is shown in Table 2. The results indicate vMod's overhead is similar to that of WiFi, and is at least 50% lower than that of rateless codes. The major reason for such overhead reduction is that vMod constantly puts the RF radio at the wireless sender to idle after each transmission, instead of keeping the radio alive for sending rateless data chunks. Such reduction of wireless channel occupancy, in addition, also provide extra channel access to other concurrent wireless traffic in the same channel.

The computation overheads for processing one data frame are shown in Figure 12. For rateless codes, such time is measured until the entire data packet is successfully decoded. At both sides, the computational overhead of vMod is similar to that of existing WiFi

Table 2: Tx communication overhead ($us/bit/MHz$)

SNR	vMod	WiFi	Strider	Spinal
6	0.2043	0.2347	0.5152	0.5500
12	0.1463	0.1603	0.2955	0.3000
24	0.0499	0.0505	0.1591	0.1500



(a) AARF adaptation (b) Minstrel adaptation

Figure 13: vMod's throughput improvement over existing WiFi with a single narrowband link

and remains nearly constant. Comparatively, rateless codes incur up to 10x higher computational overhead, especially with low SNR, because a rateless receiver needs to continuously decode incoming data chunks even if many of them are actually corrupt. Due to such processing of extra data chunks, even when the channel SNR improves to 24 dB, Table 3 shows that the computational overhead of rateless codes is still more than 100% higher than that of vMod. Less computational complexity of vMod also reduces its requirement on computing power of digital processing chips.

Table 3: Rx computation overhead (us)

SNR	vMod	WiFi	Strider	Spinal
6	486	430.1	1842.9	6074.6
12	560.1	480.2	1436.7	2543.5
24	753.4	635.0	1282.5	1042.4

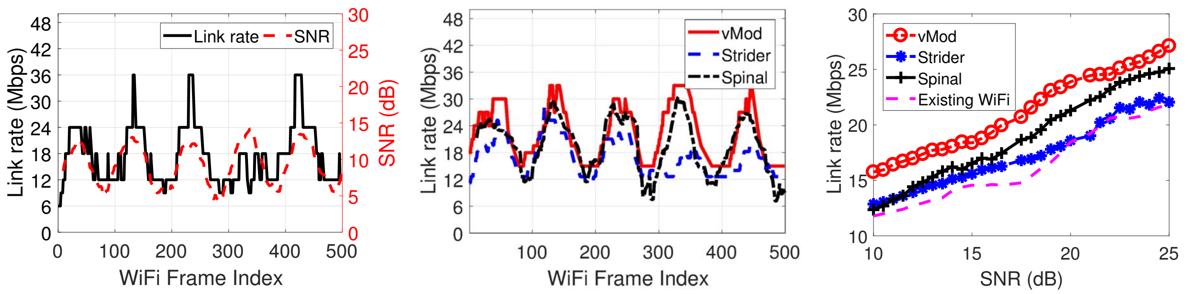
8 REAL-WORLD EXPERIMENTATION

In this section, we further apply our designs into real-world network scenarios with dynamic channel conditions and commodity rate adaptation approaches. In this way, we evaluate the improvement of data throughput by vMod. Our experiment results show that vMod outperforms rateless codes and improves the WiFi throughput by up to 30% over a narrowband link, but incurs 95% less overhead than rateless codes. It also retains such throughput in severe channel conditions with strong interference and channel fading.

8.1 Experiment Setup

Without loss of generality, in this section we only evaluate the wireless throughput over a single narrowband link. Further improvement of throughput in practical scenarios, can be achieved by either expanding the link bandwidth or employing multiple links at the same transceiver, and vMod is fully compatible with these techniques. We adopt both Minstrel and AARF algorithms for link rate adaptation, and conduct our experiments in a $5m \times 5m$ office with strong multipath fading. The sender and receiver are placed out of line of sight and 4 meters away from each other.

Similar to Section 7, we tune the Tx gain of the USRP sender to emulate different SNR conditions. To emulate both fast and slow



(a) Varying channel conditions and link rates of existing WiFi

(b) Achieved link rates over time

(c) Achieved link rates over different channel SNR

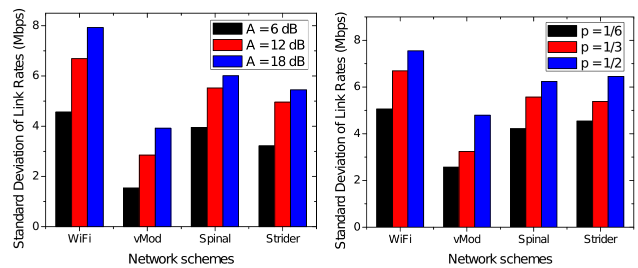
Figure 14: Comparison of achieved wireless throughput with existing WiFi and rateless codes over a single narrowband link. Tx gain is 6 ± 3dB. vMod outperforms Spinal and Strider by up to 25%, especially when the channel SNR is low. It also incurs much lower computational overhead compared to Spinal and Strider, as shown in Figure 12.

channel SNR variations, we configure the Tx gain for sending the N-th data frame as $A(1 + \rho \sin(2\pi\omega N))$, where $\omega = 0.1$ (i.e. the period of the Tx gain change is 10 frames) and A is the average level of the Tx gain being configured between 6dB and 18dB. The parameter $\rho \in \{\frac{1}{6}, \frac{1}{4}, \frac{1}{3}, \frac{5}{12}, \frac{1}{2}\}$ is an indicator of the channel variation, where a larger ρ indicates a fast fluctuating channel.

8.2 Throughput Improvement

As shown in Figure 13(a) and 13(b), vMod improves WiFi throughput with both AARF and Minstrel rate adaptation approaches. In AARF, the throughput is improved by up to 30%. As the channel condition becomes better, the throughput improvement decreases and remains nearly constant 14% when A reaches 18 dB. The figure shows that in lower channel condition, intermediate link rates can provide more throughput gain. For example, when in low channel condition, major link rates being used are 6 Mbps and 9 Mbps. When the channel condition cannot support 9 Mbps but is surplus to support 6 Mbps, an intermediate link rate of 7.5 Mbps can provide an extra 1.5 Mbps link rate, which is a 25% improvement compared to 6 Mbps. In higher SNR condition, however, 12 Mbps, 18 Mbps and 24 Mbps are mainly used and an extra 1.5 Mbps throughput only accounts for at most 12.5% throughput gain. Similarly, as shown in Figure 13(b), such throughput improvement could be further up to 30% with the Minstrel approach, although the overall throughput is slightly lower than that of AARF.

Furthermore, we have also compared the throughput achieved by vMod with that of rateless codes. As shown in Figure 14(a), when the Tx gain of the USRP sender is periodically fluctuating between 3 dB and 9 dB, the channel SNR varies accordingly and leads to different link rates. Nevertheless, under such dynamic channel conditions, Figure 14(b) shows that vMod can always provide a network throughput that exceeds the throughput achieved by both Strider and Spinal codes. Such advantage over different channel conditions is further demonstrated by Figure 14(c): although rateless codes are not based on link rate adaptation and also continuously vary the actual link rate with the channel SNR condition, vMod outperforms their performance by up to 25% under poor channel conditions. The major reason of this advantage is that under such severe scenarios, rateless codes require transmitting a large amount of data trunks in order to successfully deliver one data packet, and hence incur extra transmission latency.



(a) Different values of A with $\rho = 1/3$

(b) Different values of ρ with $A = 12$ dB

Figure 15: Stability of channel throughput over dynamic channel conditions

On the other hand, when the channel condition improves, although less data trunks are transmitted in rateless codes, the gap between discrete link rates also increases. Therefore, the throughput advantage of vMod over rateless codes remains at the same level. In particular, vMod improves the network throughput with a much lower amount of communication and computation overhead as shown in Section 7.3, and is hence applicable for practical deployment over the off-the-shelf wireless systems.

8.3 Stability of Channel Throughput

In practical wireless network scenarios where the channel SNR fluctuates, Figure 14(b) also shows that corresponding link rate changes over time, and demonstrates that the curve of link rate fluctuation using vMod is smoother than both the existing WiFi and rateless code designs. This experimental result, hence, verifies our expectation that continuous link rates can better accommodate to the momentary channel conditions. Furthermore, we also calculated the standard deviation of the link rates over the experiment period with different values of A and ρ that decides the fluctuation of channel condition. The results in Figure 15, again, verify the vMod's advantage in providing more stable throughput over dynamic channel conditions. In particular, when the channel condition fluctuates faster with a smaller value of ρ, vMod provides better stability to the WiFi network throughput.

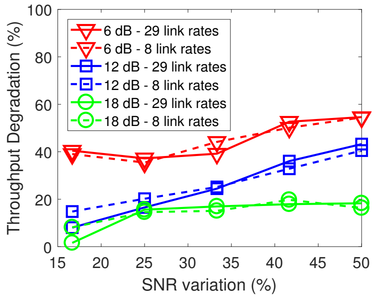


Figure 16: Throughput in an interfered channel

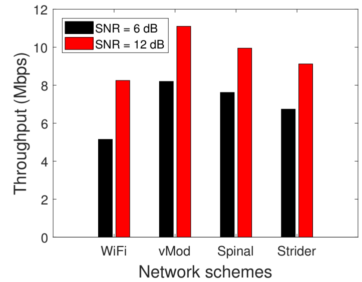


Figure 17: Throughput over fading channel

8.4 Impact of Channel Interference and Fading

We investigate the anti-interference capability of vMod by adding an interfering source node, which transmits a 1,500-byte WiFi frame every 0.1 seconds in the same frequency band as USRP. The throughput degradation caused by such interference is shown in Figure 16, where 8 link rates are employed by the existing WiFi standard and 29 continuous link rates are employed by vMod as specified in Table 1. When the channel SNR is lower or fluctuates more quickly, the throughput degradation becomes larger. However, such degradation with continuous link rates provided by vMod is very close to that with existing coarse-grained link rates in all channel SNR conditions. This result demonstrates that the adoption of continuous link rates is suitable in practice and does not impair the capability of WiFi networks in resisting channel interference.

We also tested the performance of vMod over a frequency selective fading channel, by passing the generated Tx waveform to a software-defined selective fading channel before emitting it to the air. The experiment results are shown in Figure 17, and demonstrate 40% and 30% throughput improvement compared to existing WiFi under poor (SNR=6 dB) and medium (SNR=12 dB) channel condition, respectively. vMod also outperforms rateless codes under different conditions in such a frequency selective fading channel.

9 RELATED WORK

Adaptive Modulation. Our proposed design of vMod is related to existing work on adaptive modulation, which improves throughput over fading channels by adjusting the modulation scheme to the instantaneous channel condition [11]. As a representative approach to adaptive modulation, Trellis Coded Modulation (TCM) [35] introduces intermediate data rates by combining the modulation procedure with the channel coding and applying different Trellis codes over the QAM constellation points. However, TCM requires large constellation diagrams and is rarely used in wireless networks due to the high computational complexity. Furthermore, TCM is known to be prone to phase ambiguity and hence not suitable to be used in dynamic wireless scenarios. Other approaches provide more choices of link rates through constellation diagrams that contain 3×2^p signal points [20, 24] and have been extended to the plastic optical fiber communication [33], but can only reduce the gap between consecutive link rates down to half bit per data symbol. Adjusting the RF transmit power over fading channel is proposed in [11, 18] so as to maximize the channel throughput. However, the performance gain is restricted due to the limited levels of RF transmit power in commodity WiFi devices [3].

Rateless Codes. Rateless codes aim to fully utilize the channel capacity without channel feedback [29]. Existing designs differ in the way they create continuous streams of encoded data: Strider [12] builds on a conventional channel code and transmits linear combinations of conventionally encoded symbols, and Spinal code [26] utilizes a pseudo-random hash function to produce the sequence of coded symbols. However, as described in Section 2.2, these designs incur a large amount of extra communication and computation overhead, and are hence inapplicable to be deployed over resource-constrained wireless devices, particularly embedded sensors in IoT applications. In addition, new MAC protocols are also needed for rateless codes [17], because the transmitter and receiver have to be appropriately coordinated so that the sender can know when the packets are successfully decoded and adjust the interval of packet transmissions accordingly. Such MAC protocol is, however, incompatible with any existing wireless network and requires fundamental network redesign. Instead, our approach is completely compatible with any existing MAC protocol, and does not require any modification of the network protocol stack.

Wireless Networking. Many techniques have been developed to tackle with wireless channel fluctuations, improve the throughput and reduce the delay. Such techniques include multiuser MIMO (MU-MIMO) [30], full-duplex communication [9] and side channels [23], and have been integrated to recent wireless standards such as LTE, 802.11ac and 802.11ax [5]. Since vMod provides continuous choices of link rates by modifying the channel modulation, it is independent from most PHY and MAC modules of existing wireless systems. It is fully compatible with and can be applied to the aforementioned advanced techniques, to further improve their performance in highly severe application scenarios.

10 DISCUSSION

Compatibility with link rate adaptation. Most rate adaptation algorithms reactively adapt the link rate according to frame delivery in the past [36]. Others adjust the link rate based on different channel state indicators, including the signal strength [8], channel state information (CSI) [22] or effective SNR [14]. Details of packet reception, such as channel BER [34] and symbol error vectors [28], have also been used. vMod, as a channel modulation scheme, provides continuous choices to link rate adaptation and is hence compatible with any link rate adaptation algorithm. It does not require any modification to existing rate adaptation algorithms, either.

Variable-length codewords based on Huffman code. Huffman code in this paper is exploited to construct the variable-length codewords from existing fixed-length codewords in QAM modulations. As described in Section 4, the completeness and optimality of Huffman code in vMod are ensured by the constitution of such variable-length codewords and the corresponding continuous link rates. In particular, multiple codeword lengths provide more choices of continuous link rates, but also increase the complexity of demodulation, because the number of each type of codeword being used must be explicitly notified to the receiver for correct demodulation.

Accuracy in VLC-based demodulation. The accuracy of VLC-based demodulation is decided by two major factors: *i)* the accuracy of symbol recognition, and *ii)* the capability of error correction over the Hamming code being used. Symbol recognition requires detectable distance between constellation points from the regular QAM diagram and the new vMod diagram. As channel SNR improves, denser constellation diagrams are used and hence reduce such distance. However, the higher SNR also suppresses channel noise and achieves higher accuracy of symbol detection. Hence, the dependency of Hamming code on successful decoding varies across different levels of channel SNR. We found that a revised (8,4,4) Hamming code meets the requirements of any constellation diagram smaller than 64-QAM. We refer exploration of such accuracy in higher-order constellation diagrams as future work.

I/Q imbalance in vMod. The power efficiency and demodulation of vMod could also be affected by the heterogeneous I/Q imbalance caused by the asymmetry of the proposed constellation diagrams. However, in practice, such I/Q imbalance is usually calibrated by manufacturers according to the IEEE standard [1], before the wireless system has been released to market. The impact of such I/Q imbalance at run-time, hence, can be efficiently eliminated by the RF frontend of wireless, before the received wireless symbols are being demodulated in the digital logic.

11 CONCLUSION

In this paper, we present a lightweight solution towards maximum wireless throughput for IoT, by transforming the choices for link rate adaptation from discrete to continuous. Our design builds on the variable-length code (VLC) that conveys a fractional number of data bits in each symbol. We implemented and evaluated our design over practical SDR platforms, and show that vMod greatly improves the wireless throughput with negligible overhead.

ACKNOWLEDGMENTS

We sincerely thank our shepherd Bhaskar Krishnamachari and anonymous reviewers for their valuable comments and feedback. This work was supported in part by the National Science Foundation (NSF) under grant number CNS-1812399, CNS-1812407 and CNS-1821874. This work was also supported in part by the Army Research Office (ARO) under grant number W911NF-15-1-0221.

REFERENCES

[1] 2012. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Standard 802.11* (2012).
[2] 2013. IEEE Approved Draft Standard for IT - Telecommunications and Information Exchange Between Systems. *IEEE P802.11ac/D7.0* (2013), 1–456.
[3] 2017. Channels and Maximum Power Settings for Cisco Aironet Lightweight Access Points. (2017). https://www.cisco.com/c/en/us/td/docs/wireless/access_point/channels/lwapp/reference/guide/lw_chp2/1100_chp.html#wp1042809
[4] 2017. Minstrel rate control algorithm for 802.11 networks. (2017). https://wireless.wiki.kernel.org/en/developers/documentation/mac80211/ratecontrol/minstrel
[5] Boris Bellalta, Luciano Bononi, Raffaele Bruno, and Andreas Kassler. 2015. Next generation IEEE 802.11 Wireless Local Area Networks: Current status, future directions and open challenges. *Computer Communications* (2015).
[6] Margaret Bernard and Bhu Dev Sharma. 1988. Some combinatorial results on variable length error correcting codes. *Ars Combinatoria* 25 (1988), 181–194.
[7] Margaret Ann Bernard and Bhu Dev Sharma. 1990. A lower bound on average codeword length of variable length error-correcting codes. *IEEE transactions on information theory* 36, 6 (1990), 1474–1475.
[8] Joseph Camp and Edward Knightly. 2008. Modulation Rate Adaptation in Urban and Vehicular Environments: Cross-layer Implementation and Experimental Evaluation. In *ACM MobiCom*.
[9] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. 2010. Achieving single channel, full duplex wireless communication. In *ACM MobiCom*.
[10] Xiaodai Dong, Norman C Beaulieu, and Paul H Wittke. 1999. Signaling constellations for fading channels. *IEEE Transactions on Communications* 47, 5 (1999).
[11] Andrea J Goldsmith and Soon-Ghee Chua. 1998. Adaptive coded modulation for fading channels. *IEEE Trans. on Communications* (1998).
[12] Aditya Gudipati and Sachin Katti. 2011. Strider: Automatic rate adaptation and collision handling. In *ACM SIGCOMM*.
[13] J. R. Gutierrez-Agullo, B. Coll-Perales, and J. Gozalvez. 2010. An IEEE 802.11 MAC Software Defined Radio implementation for experimental wireless communications and networking research. In *Wireless Days (WD), 2010 IFIP*.
[14] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. 2010. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. In *ACM SIGCOMM*.
[15] D. A. Huffman. 1952. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE* 40, 9 (Sept 1952), 1098–1101. https://doi.org/10.1109/JRPROC.1952.273898
[16] Peter Iannucci, Kermin Elliott Fleming, Jonathan Perry, Hari Balakrishnan, and Devavrat Shah. 2012. A Hardware Spinal Decoder. In *ACM/IEEE ANCS*.
[17] Peter Anthony Iannucci, Jonathan Perry, Hari Balakrishnan, and Devavrat Shah. 2012. No symbol left behind: a link-layer protocol for rateless codes. In *ACM MobiCom*.
[18] Jiho Jang and Kwang Bok Lee. 2003. Transmit power adaptation for multiuser OFDM systems. *IEEE Journal on selected areas in communications* 21, 2 (2003).
[19] Mathieu Lacage, Mohammad Hossein Manshaei, and Thierry Turletti. 2004. IEEE 802.11 Rate Adaptation: A Practical Approach. In *ACM MSWiM*.
[20] Anh Tuan Le and Kiyomichi Araki. 2008. A group of modulation schemes for adaptive modulation. In *Proceedings of the 11th IEEE Singapore International Conference on Communication Systems*. 864–869.
[21] Yong Li and Wei Gao. 2017. Interconnecting heterogeneous devices in the personal mobile cloud. In *Proceedings of IEEE INFOCOM*.
[22] Zhenjiang Li, Yaxiong Xie, Mo Li, and Kyle Jamieson. 2015. Recitation: Rehearsing wireless packet reception in software. In *ACM MobiCom*.
[23] Haoyang Lu and Wei Gao. 2016. Supporting Real-time Wireless Traffic through a High-Throughput Side Channel. In *In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.
[24] Seiichi Noda, Yoichi Saito, and Teruaki Yoshida. 2007. Configuration and error ratio performance of M-QAM whose number of signal points is not a power of 2. *Electronics and Communications in Japan* 90, 2 (2007), 46–57.
[25] Eldad Perahia and Michelle X Gong. 2011. Gigabit wireless LANs: an overview of IEEE 802.11ac and 802.11ad. *ACM SIGMOBILE Mobile Computing and Communications Review* (2011).
[26] Jonathan Perry, Peter A Iannucci, Kermin E Fleming, Hari Balakrishnan, and Devavrat Shah. 2012. Spinal codes. In *ACM SIGCOMM*.
[27] Hariharan Rahul, Farinaz Edalat, Dina Katabi, and Charles G. Sodini. 2009. Frequency-aware Rate Adaptation and MAC Protocols. In *ACM MobiCom*.
[28] Souvik Sen, Naveen Santhapuri, Romit Roy Choudhury, and Srihari Nelakuditi. 2010. AccuRate: Constellation Based Rate Estimation in Wireless Networks.. In *Proceedings of USENIX NSDI*.
[29] Amin Shokrollahi. 2006. Raptor codes. *IEEE transactions on information theory* 52, 6 (2006), 2551–2567.
[30] Sanjib Sur, Ioannis Pefkianakis, Xinyu Zhang, and Kyu-Han Kim. 2016. Practical MU-MIMO User Selection on 802.11ac Commodity Networks. In *ACM MobiCom*.
[31] Yasuhiro Takishima, Masahiro Wada, and Hitomi Murakami. 1995. Reversible variable length codes. *IEEE Transactions on Communications* 43, 2/3/4 (1995).
[32] John Thomson et al. 2002. An integrated 802.11a baseband and MAC processor. In *IEEE International Solid-State Circuits Conference*. 126–451.
[33] Davide Visani et al. 2011. QAM Constellation Formats for DMT Over 1-mm Core Diameter Plastic Optical Fiber. *IEEE Photonics Technology Letters* 23, 12 (2011), 768–770.
[34] Mythili Vutukuru, Hari Balakrishnan, and Kyle Jamieson. 2009. Cross-layer wireless bit rate adaptation. *ACM SIGCOMM Computer Communication Review* 39, 4 (2009), 3–14.
[35] Lee-Fang Wei. 1987. Trellis-coded modulation with multidimensional constellations. *IEEE Trans. on Information Theory* (1987).
[36] Starsky H. Y. Wong, Hao Yang, Songwu Lu, and Vaduvur Bharghavan. 2006. Robust Rate Adaptation for 802.11 Wireless Networks. In *ACM MobiCom*.