Seamless Rate Adaptation for Wireless Networking

Hao Cui[*]
University of Science and
Technology of China
Hefei, 230027, P.R. China
hcuipro@mail.ustc.edu.cn

Chong Luo
Microsoft Research Asia
Beijing, 100080, P.R. China
chong.luo@microsoft.com

Kun Tan
Microsoft Research Asia
Beijing, 100080, P.R. China
kuntan@microsoft.com

Feng Wu
Microsoft Research Asia
Beijing, 100080, P.R. China
fengwu@microsoft.com

Chang Wen Chen
University at Buffalo, The
State University of New York
Buffalo, NY 14260-2000, USA
chencw@buffalo.edu

ABSTRACT

This paper aims at designing a *Seamless Rate Adaptation* for wireless networking which achieves smooth rate adjustment in a broad dynamic range of channel conditions. Conventional rate adaptation can only achieve a stair-case rate adjustment. Even when combining with hybrid ARQ, it suffers from an irreconcilable conflict between throughput and dynamic range. We tackle this problem from a new perspective by relying on modulation, instead of channel coding, for rate adaptation. We propose rate compatible modulation (RCM), in which modulation signals are incrementally generated from information bits through weighted mapping. Rate adaptation is achieved through varying the number of modulated signals. As more signals are transmitted, information bits gradually accumulate energy. The weights in bit-to-symbol mapping are delicately designed to ensure fine-grained energy accumulation so that smoothness and efficiency can both be achieved. We design and implement a rate adaptation system, called SRA and evaluate its performance through a software radio testbed. Results show that, under highly dynamic channel conditions, SRA achieves over 80% throughput gain over 802.11a adaptive modulation and coding, and achieves 28.8% and 43.8% gain over HARQ systems implemented with Turbo code and Raptor code. We believe that the concept of rate compatible modulation opens up a fresh research avenue toward the wireless rate adaptation problem.

Categories and Subject Descriptors

C.2.1 [**Computer-Communications Networks**]: Network Architecture and Design-*Wireless Communications*

General Terms

Design, Performance, Experimentation

Keywords

rate adaptation, modulation

1. INTRODUCTION

In wireless networking, rate adaptation is critical to the system performance by exploiting the dynamic bandwidth. Though it is not specified in current 802.11 standards, all 802.11 systems need to dynamically change the rate for transmission. In the standards, a set of rates is mandated in physical layer for selection. They are implemented by combining different modulation and channel coding schemes. According to where the rate selection is performed, rate adaptation can be categorized as sender rate adaptation and receiver rate adaptation.

Most existing rate adaptation schemes adopt sender rate adaptation, which is also known as adaptive modulation and coding (AMC). Specifically, the receiver observes channel condition and delivers a feedback to the sender for rate selection. These schemes mainly focus on which metric should be used for channel estimation. Usually, the metrics for evaluating channel condition are signal to noise ratio(SNR) [7, 8], frame loss rate [9, 23], interference-free bit error rate [22] and symbol-level dispersions [15]. However, there always exists a dilemma in sender rate adaptation no matter what metric is used. That is, in time varying wireless networks, instant channel estimation cannot guarantee accuracy, while accurate channel estimation needs a long term observation. Either inaccurate or delayed estimation may lead to degradation from achievable throughput. In addition, the rate set provided by physical layer is discrete. Even a perfect channel state is known at the sender, it can only achieve stair-case rate adjustment. The gap between adjacent rates cannot be fully utilized.

Hybrid ARQ (HARQ) is a complementary to sender rate adaptation in a form of channel coding, which fills this gap in some sense. It defines an acknowledgement mechanism to issue retransmission when error occurs. The retransmission is either a repetition (type I HARQ) or incremental redundancy (type II HARQ). The rate can be smoothly changed by increasing the retransmission times. Since the through-

[*]This work was carried out when H.Cui was a research intern at Microsoft Research Asia.

MSWiM'11, October 31–November 4, 2011, Miami, Florida, USA.
Copyright 2011 ACM 978-1-4503-0898-4/11/10 ...$10.00.

put of type II HARQ is much higher than type I HARQ, we only discuss type II HARQ in this paper and use HARQ as type II HARQ unless otherwise specified. In HARQ, the incremental redundancy is usually encoded by either a rate compatible code, e.g. punctured Turbo code [13] or fountain code, e.g. Raptor code [19]. However, both of them only have a limited rate dynamic range. To cover a typical range, from 0.5 bit/s/Hz to 6 bit/s/Hz, one way is to use AMC. This will fall back to the dilemma. The other way is to fix the modulation and use a low rate mother code for puncturing. The problem in punctured Turbo code is that a low rate mother code encounters throughput loss at high end [14]. In Raptor code, which is originally designed for binary erasure channel (BEC), a 0.9 outer LDPC code is needed before inner LT code for Gaussian channel [21]. In addition, the overhead of LT code in Gaussian channel is larger than that in BEC. Thus, the outer coding rate and larger overhead make it saturate at a lower transmission rate than punctured Turbo code. We will show this through simulation.

To tackle these issues, we need a receiver rate adaptation approach, which can be seamless. By seamless, we mean that the rate adjustment is rateless, smooth and spread on the typical range. The sender continuously generates and transmits coded symbols for a data block. The receiver collects symbols until the data block can be correctly decoded and delivers back an ACK for the transmission of next block. In addition, since we cannot guarantee an accurate channel estimation, the system performance should be insensitive to the channel estimation error.

In order to achieve the desired seamless adaptation, we propose a rate compatible modulation (RCM) which is motivated by the analysis of conventional modulation. RCM incrementally generates symbols with fine-grained bit energy allocation. Collecting symbols can progressively accumulate the transmission power until meeting the requirement, which is rateless. Unlike conventional modulation, the generated symbols are directly placed on each dimension of the constellation to modulate signals, so that the symbol locates in the same domain with channel noise. This makes it possible for smooth degradation. The constellation in RCM is fixed and very dense (23×23 QAM). By fixing the constellation, there is no need to make decision on modulation selection and avoid the dilemma as mentioned above. Dense constellation contains large amount information and it will not be easily saturated. Therefore, it covers a broad dynamic range.

Based on RCM, we design and implement a rate adaptation system, called SRA (seamless rate adaptation), on orthogonal frequency division multiplexing (OFDM) physical layer, and evaluate its performance on a software radio platform. Results show that, while SRA excels others in all four test scenarios, it shows greatest advantage in non-line-of-sight scenario, achieving over 80% gain over conventional 802.11 rate adaptation which is pure AMC, 28.8% and 43.8% improvement on Turbo code and Raptor code based HARQ with AMC.

The rest of the paper is organized as follows. Section 2 provides the recent works on receiver rate adaptation. Section 3 presents the motivation and design of RCM, then introduces the modulation and demodulation process and its complexity. Section 4 evaluates RCM through Matlab simulations. Section 5 describes the implementation and evaluation of SRA system on software radio testbed. Finally, we conclude this paper in Section 6.

2. RELATED WORK

One representative work of receiver adaptation is adaptive demodulation (ADM) [5]. It employs rateless coding and fixed 16QAM constellation. In particular, source bits are encoded by a Raptor code, then every four rateless bits form a symbol and is mapped to 16QAM constellation with Gray code. Upon receiving a symbol, the receiver collects reliable bits by adjusting the decision regions for demodulation. These rateless bits are then accumulated for decoding. This scheme has two problems. First, it obviously has a saturation rate at 4 bits/Hz. Second, the authors neglect the fact that rateless code can also be used in channels other than BEC. In a noisy channel, unreliable bits also contain some amount of information. Simply discarding them would dramatically decrease channel throughput.

A recent work automatic rate adaptation (ARA) [6] shares the same goal with our work to achieve feedback-free rate adaptation. In particular, ARA adds a minimum distance transformer (MDT) after existing coding and modulation to perform linear combinations for modulated symbols. Although ARA requires no change to existing modulation and coding, its performance is adversely affected by the limitations of existing modules. In addition, ARA performs linear combination through a dense random matrix. This dramatically increases complexity in the decoding. The authors have suggested Lattice decoding which has a complexity of $O(N^3)$ [10].

3. RATE COMPATIBLE MODULATION

Rate compatible indicates that the constellation is fixed and the transmitted symbols of high rate modulation are used by all low rate modulation. Let N be the bit block length for modulation, and K be the number of modulation signals for successful transmission. Then the modulation rate is $\frac{N}{K}$. First, K_0 signals are transmitted to keep a high rate modulation. If low rate modulation is needed for transmission, we just increase the number of signals step by K_c. Therefore, the achievable modulation rate are:

$$R = \frac{N}{K} = \frac{N}{K_0 + i \cdot K_c} \; i = 0, 1, \ldots$$

3.1 Motivation

In conventional modulation, the minimum Euclidean distance between adjacent constellation points is called free distance. Under the same noise amplitude, the bit error rate (BER) exponentially decreases with increasing free distance [18]. When the transmission power is fixed, high rate modulation conveys more data bits per transmission, but its free distance is small. Although low rate modulation conveys less data bits per transmission, its free distance is large. So switching between different modulations is to choose an appropriate free distance for resisting the immediate channel noise. In fact, the free distance is equivalent to another modulation parameter, i.e. average bit energy. Denote E_s as the average symbol energy of the constellation and K as the number of bits contained in each symbol, then the average bit energy is $E_b \triangleq E_s/K$. The free distance is proportional to E_b [18]. It is possible to change the free distance by tuning the number of transmitted symbols. However, this is

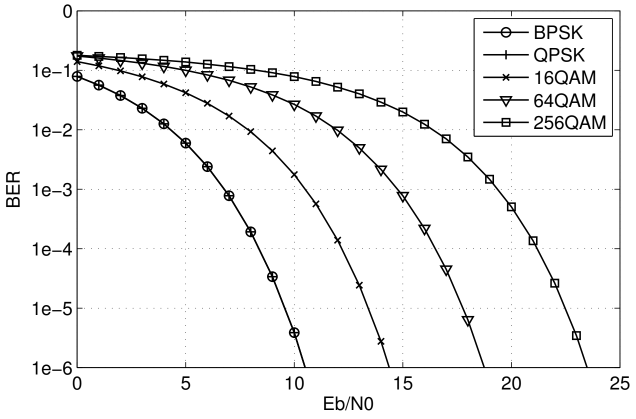


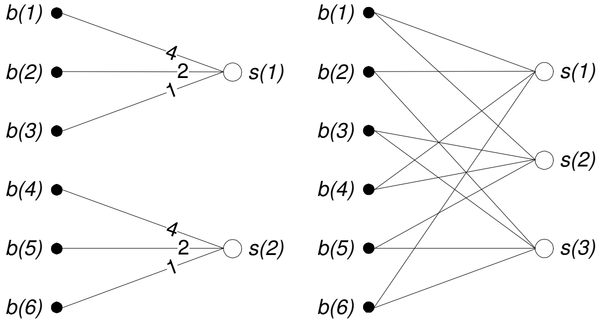
Figure 1: BER of different modulation schemes

not efficient for conventional modulations. To do this, the bit energy should be accumulated from a low point which means a high rate modulation. The problem in conventional modulation is that the bit energy is not equally allocated in the symbols of a higher rate modulation. Unequal bit energy allocation leads to unequal BER. Since the BER of a modulation is calculated by averaging over all bits, it will be dominated by the bits with higher BER. Fig. 1 shows the BER vs. bit energy to noise power ratio (E_b/N_0) for different modulations. We see that even under the same average bit energy, the BER of high rate modulation is higher than that of low rate modulation. That's why rate adaptation based on conventional modulation can only be achieved by switching modulation schemes.

Therefore, to realize the rate compatibility in modulation, we need an even bit energy allocation, which can increase the free distance by accumulating the received symbols and achieve the same effect of switching between different modulations. Fig. 2 presents our heuristic idea. Assume that there are 6 bits for modulation. Fig. 2(a) shows the mapping of conventional modulation 8 PAM (or one dimension of 64 QAM). It takes each 3 bits as one group and then generates a symbol by weighted sum. $\{1, 2, 4\}$ is taken as weight for identical bit to symbol mapping. However, this mapping cannot equally increase the bit energy with increasing retransmissions, since the bit energy ratio is always 1:4:16. A heuristic method is to replace the mapping by what is shown in Fig. 2(b), each bit is sampled by multiple symbol and the sampling weight vector for each bit is with equal norm. Only such a mapping can get evenly increasing bit energy.

3.2 Design

Now the problem becomes how to map bits to symbols. Since the mapping is a weighted bipartite graph, it can be described by its incidence matrix from graph theory. Denote $\mathbf{G}$ with dimension $M \times N$ as the incidence matrix which is named mapping matrix in this paper. Let the row represents the bits and the column represents the symbols. If there exists one edge between a bit and a symbol, the corresponding element in $\mathbf{G}$ is the weight on the edge, or the corresponding element will be zero. In the following, we first describe the constraints on $\mathbf{G}$. Then we will show how to construct such a mapping matrix.



(a) Conventional

(b) Heuristic

Figure 2: Bit-to-symbol mapping

3.2.1 Constraints on Mapping Matrix

Constraint 1: $\mathbf{G}$ *should be regular in rows.*

We desire a fixed constellation, which implies a fixed symbol alphabet. As the RCM symbols are generated by weighted sum operation, the symbol alphabet is determined by the weights. Therefore, we require $\mathbf{G}$ to be regular in rows, i.e. each row contains the same number of non-zero entries drawn from a fixed weight set.

Let L be the number of non-zero entries in each row, and $\mathcal{W}_s = \{w_1, w_2, ...w_L\}$ be the fixed weight set, then each symbol $u(\cdot)$ can be computed by:

$$u(m) = \sum_{l=1}^{L} w_l b(m_l)$$

where $b(\cdot)$ is the bit block and m_l is the corresponding bit index of weight w_l in the m^{th} row.

Although the weight set is fixed, the weights should be randomly placed in each row. Such a random property can create many symbols, which enables rateless.

Constraint 2: $\mathbf{G}$ *should be as regular as possible in columns.*

All weights in a column are associated with its corresponding bit. The allocated energy to bit $b(n)$ is actually the norm of each column vector:

$$E_{b(n)} = C \cdot \sum_{r=1}^{R_n} w_{n_r}^2$$

where R_n is the number of non-zero weights in the n^{th} column, n_r indicates which weight is used and C is a normalization factor. Equal energy increase can be achieved if the columns are regular in norm.

It is trivial to get regular columns when $M = N$. We require the matrix to be regular when $M = N/2$ or even $M = N/4$. A positive-negative symmetric weight set can easily satisfy the regularity when $M = N/2$.

Constraint 3: *The weight set* $\mathcal{W}_s$ *should create diverse symbol values.*

From information theoretic perspective, each received symbol carries a certain amount of information upper bounded by the channel capacity and the entropy of the transmitted symbol. In order to avoid rate saturation at high channel SNR, the entropy of modulated symbols should be high enough.

Using diverse weights can increase the diversity of symbols, and consequently increase symbol entropy. We set a threshold for symbol entropy at 4 bit/dimension, which is equivalent to 256QAM rate.

3.2.2 *Construction of Mapping Matrix*

We find that the weight set $\{\pm 1, \pm 2, \pm 4, \pm 4\}$ satisfies all the constraints outlined above. Now, we present three steps to construct the generation matrix $\mathbf{G}$ such that it has good properties at varying values of M. First, we construct three elementary matrices A_1, A_2 and A_4. Each elementary matrix has dimension $N/8 \times N/4$. The structure of A_1 is shown as follows. Matrices A_2 and A_4 have the same structure, but the non-zero values are replaced with $+2/-2$ or $+4/-4$.

$$A_1 = \begin{bmatrix} +1 & -1 & & & & & \\ & & +1 & -1 & & & \\ & & & & \ddots & & \\ & & & & & +1 & -1 \end{bmatrix}$$

Second, we form an $N/2 \times N$ matrix $\mathbf{G}_0$ by stacking random permutations of the three elementary matrices as follows:

$$\mathbf{G}_0 = \begin{bmatrix} \pi(A_4) & \pi(A_4) & \pi(A_2) & \pi(A_1) \\ \pi(A_2) & \pi(A_1) & \pi(A_4) & \pi(A_4) \\ \pi(A_4) & \pi(A_4) & \pi(A_1) & \pi(A_2) \\ \pi(A_1) & \pi(A_2) & \pi(A_4) & \pi(A_4) \end{bmatrix}$$

where $\pi(\cdot)$ denotes randomly permutated columns of a matrix. By using different permutation choices, we may construct virtually unlimited number of matrix $\mathbf{G}_0$.

The third and final step in constructing $\mathbf{G}$ is to stack all the randomly generated $\mathbf{G}_0's$. In practice, we only stack two matrices to form an $N \times N$ matrix, and repeatedly use it when the channel condition is poor. This matrix can be communicated between the sender and the receiver before transmission. When a symbol is repeatedly transmitted, multiple versions can be combined through maximal ratio combining.

3.3 Modulation

A data packet from upper layer is first divided into bit blocks by length N. Let $\mathbf{b} = \{b(n)\}_{n=1}^{N}$ be one bit block. Based on the above mapping matrix $\mathbf{G}$, we can generate symbols by

$$\mathbf{u} = \mathbf{G} \cdot \mathbf{b}$$

where $\mathbf{u} = \{u(m)\}_{m=1}^{M}$ is the generated symbol block for transmission. Then the symbol is directly used to modulate the signal amplitude. Since the weight set $\{\pm 1, \pm 2, \pm 4, \pm 4\}$ is used, the signal amplitude ranges from $-11d$ to $11d$ and is Bernoulli distributed, where d is a scale factor and is also the minimum distance between constellation points. To fully utilize the constellation plane, i.e. I (in phase) and Q (quadrature phase), each two consecutive symbols compose one modulation signal by:

$$u(2k) + \sqrt{-1} \cdot u(2k+1), \ (k = 0, 1, \ldots, M/2 - 1).$$

Therefore, we are actually using a 23×23 QAM constellation. When the transmission power is normalized to 1 based on its distribution, $d \approx 0.1644$ which is similar to 256 QAM.

3.4 Demodulation

We use the standard AWGN channel model to design our demodulation algorithm. Denote $\mathbf{u}'$ as the received symbols. We have

$$\mathbf{u}' = \mathbf{G} \cdot \mathbf{b} + \mathbf{e}$$

where $\mathbf{e} = \{e(m)\}_{m=1}^{M}$ is a Gaussian noise vector with $e(m) \sim \mathcal{N}(0, \sigma^2)$.

The demodulation process is to find the optimal solution to the following problem:

$$\hat{\mathbf{b}} = \underset{\mathbf{b} \in GF(2^N)}{\arg\max} P(\mathbf{b}|\mathbf{u}') \quad (1)$$

The transmitted symbol is generated by a non-identical mapping, we cannot perform symbol-wise demodulation that is to find the closest constellation point to the received symbol. From [4], the problem (1) can be solved by belief propagation. However, binary digits constraint $\mathbf{b} \in GF(2^N)$ is introduced in this problem. The computation of convolution by FFT [4] is expensive when performing on binary digits. We modify the FFT to convolution operations in a FPGA manner, where only non-zero elements are taken into computation. The details of our demodulation algorithm is shown in Appendix A.

During the belief propagation process, we assume that the channel SNR is correctly estimated. Later, in Section 4, we will show through simulation that our algorithm is not sensitive to SNR estimation error. In Section 5, the evaluation over software radio testbed will demonstrate that our demodulation algorithm performs well when the real channel is not AWGN.

3.5 Complexity

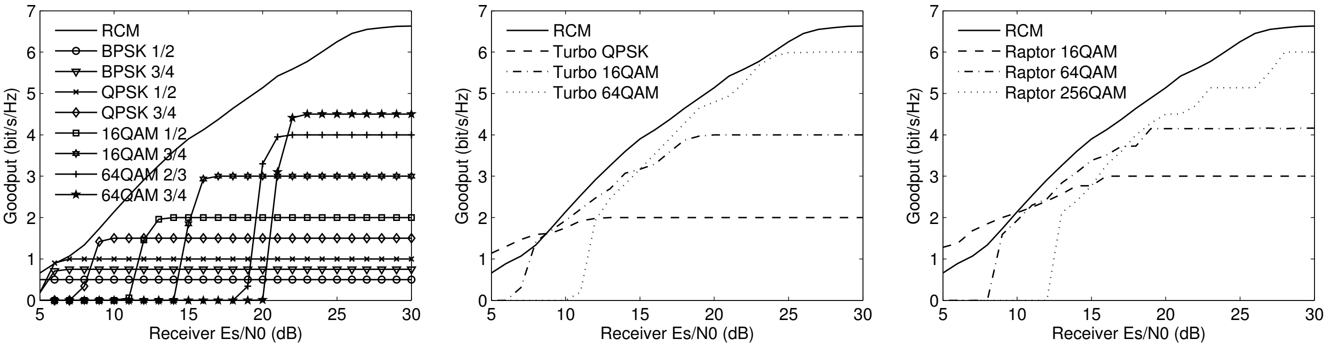
We shall analyze the complexity of RCM in two aspects: modulation complexity and demodulation complexity. Let R and L be the number of nonzero elements in each column and row of the generation matrix respectively. Constructing the generation matrix is a one-time task and the complexity is $O(R \cdot N)$. As each RCM symbol is the weighted sum of L bits, it can be generated through looking up a pre-defined table of size 2^L. Therefore, the modulation complexity for generating M symbols is $O(M)$. Our demodulation procedure is based on belief propagation algorithm. The only difference is that we use convolution and deconvolution to deduce the confidence of each bit from the received RCM symbol. In each iteration, the computation cost of belief deduction for each bit is bounded by $O(W \cdot R)$ which is a constant, where $W = 1 + \sum_{i=1}^{L} |w_i|$. Therefore, the demodulation complexity is $O(W \cdot R \cdot N)$, where W and R are constant for the RCM. Therefore, RCM has a linear modulation and demodulation complexity.

4. SIMULATION

The Matlab simulation serves two purposes. One is to evaluate the rate adaptation capability of RCM. The other is to choose appropriate parameters for the practical system design.

4.1 Reference Schemes

The rate adaptation capability is evaluated against ideal AMC and HARQ with AMC based on both Turbo codes and Raptor codes. For all schemes, we assume immediate feedback from the receiver. The sender stops the transmission when an ACK is received or the maximum symbol number is reached.



(a) Comparison with Ideal AMC (b) Comparison with HARQ-CTC (c) Comparison with HARQ-Raptor

Figure 3: Comparing RCM's rate adaptation capability with existing schemes

4.1.1 *Ideal AMC*

Conventional rate adaptation is exemplified by the AMC based on 802.11a standard [1]. The modulations are BPSK, QPSK, 16QAM and 64QAM and the channel code is convolutional code. In total, there are eight different combinations. By ideal, we mean that the sender knows exactly what the channel condition is and can make the optimal rate selection. Therefore, ideal AMC is the upper bound of all conventional rate adaptation schemes.

4.1.2 *HARQ with AMC based on Turbo code*

The first implementation of HARQ is based on convolutional Turbo code (HARQ-CTC). We use 1/3 Turbo code as mother code. As in WCDMA and LTE [3], the code length is set to 1200 bits, and the component encoder is based on recursive convolutional code with polynomial $(13, 15)$ in octal. In order to achieve smooth rate adaptation, the puncturing period is set to 8. We use the same puncture pattern as presented by Rowitch and Milstein [14]. Rates corresponding to $\frac{8}{8+l}$ $(l = 0, 2, ...16)$ are used.

HARQ-CTC cooperates with three modulation schemes: QPSK, 16QAM, and 64QAM. Different modulation and coding schemes create 27 combinations and 21 identical rates. The decoder we used at the receiver is soft input Viterbi decoder with 8 iterations.

4.1.3 *HARQ with AMC based on Raptor code*

The second implementation of HARQ is based on Raptor code (HARQ-Raptor). We use an LDPC code with rate 0.9 as the outer code. Rateless bits are generated using the optimal degree distribution at BEC channel [17]. According to WiMax PHY [2], the code length is set to 1800 bits before LDPC coding and therefore 2000 bits before LT coding. Using a longer block may improve the performance but the complexity and transmission delay will also be increased.

HARQ-Raptor combines with three modulations schemes, 16QAM, 64QAM and 256QAM. As explained earlier, the highest achievable rate of Raptor code is well below 1. Therefore, we do not use QPSK modulation as the rate saturates too early. The transmission step size is set to 50 modulation signals, which translates to 200 rateless bits in 16QAM, 300 rateless bits in 64QAM and 400 rateless bits in 256QAM. The maximum number of bits generated for each data block is 16000, creating a minimum code rate of 1/8. At the receiver, belief propagation decoding is deployed.

4.2 Achievable Rates

In this simulation, RCM uses block size $N = 400$ and the retransmission step is 10 modulation signals. The rate adaptation granularity is actually the same as HARQ-Raptor, as the block length is shorter. We run all schemes on every integer SNR from 5dB to 30dB. For each SNR, a total of 10^7 bits are transmitted. We use goodput as evaluation metric [11], which is the rate of correctly received bits. We separately compare RCM with the other three schemes and show the performance of RCM in each comparison.

Fig. 3(a) shows the achievable rates of each combination of modulation and coding in 802.11a. The envelope of these combinations is the performance of ideal AMC. RCM is well above ideal AMC. Due to the coarse-granularity of AMC rate adaptation, RCM shows great advantage around the SNRs where switching occurs. For example, at 20dB, AMC still suggests to use (16QAM, 3/4) scheme whose rate is only 3 bit/s/Hz. In contrast, RCM can achieve 4.65 bit/s/Hz which is over 53% gain.

Fig. 3(b) shows the performance of HARQ-CTC with AMC. From the curves of each modulation, we see that the dynamic range is limited when only HARQ-CTC is adopted, i.e. the single curve of each modulation. To cover the typical rate range, it needs to cooperate with AMC, i.e. the envelope of three modulations. RCM has a significantly larger dynamic range than HARQ-CTC. In addition, the achievable rate of RCM is above the envelope of using 16QAM and 64QAM. When the channel SNR is below 7dB, HARQ-CTC with QPSK has a slightly better performance. However, it contributes a very small portion to the overall performance.

Fig. 3(c) shows the rate of HARQ-Raptor with AMC. Surprisingly, the dynamic range of HARQ-Raptor is even smaller than that of HARQ-CTC. We find that, although the Raptor code rate can be arbitrarily small, the additional parity bits transmitted at low SNR can not improve the decoding performance. From the curves, we see that the highest achievable rate of Raptor code is around 0.75. Therefore, the rate saturates at 4.2 bit/s/Hz when combined with 64QAM and at 6 bit/s/Hz when combined with 256QAM.

This simulation confirms that HARQ need to be combined with AMC to cover a large dynamic range. In contrast, the proposed RCM scheme has been using the same set of parameters from 5dB to 30dB, and the overall goodput is higher than the "Oracle" combined HARQ and AMC scheme which knows exactly when to switch the modulations.

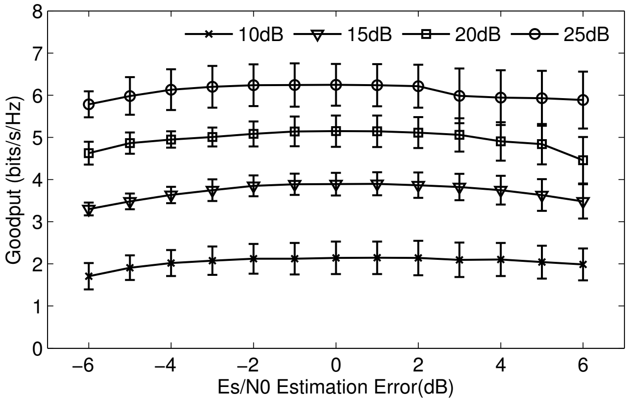


Figure 4: Sensitivity to SNR estimation errors

4.3 Sensitivity to SNR Estimation Error

As proposed earlier, precise channel SNR is hard to obtain in practical system and an ideal rate adaptation should be insensitive to channel estimation error. Therefore, we evaluate whether RCM demodulation is sensitive to inaccurate SNR input. We implemented the SNR estimation algorithm proposed by Rahul et al. [12], and found that 98% of SNR estimation error is within -6dB to 6dB.

We run four sets of simulations at four typical SNR values: 10dB, 15dB, 20dB and 25dB. In each set of simulation, we perform RCM demodulation at imprecise SNR estimation. The estimation error varies from −6dB to 6dB step by 1dB. Fig. 4 shows the mean and standard deviation of achieved goodput. Data presented are averaged over 1000 data blocks of length $N = 400$. It can be found that RCM demodulation is not sensitive to imprecise SNR input. The performance loss is not significant even when the input SNR is 6dB lower than the actual value.

4.4 Demodulation Threshold

In a practical system, a receiver should start RCM demodulation after a minimum number M_0 of modulation signals have been received. A small M_0 will increase the system complexity, but a large one may degrade the achievable rate. We create a lookup table of demodulation threshold with respect to channel SNR.

The table is created by the following process. We run a large number of simulations on each integer SNR value from 5dB to 30dB, and plot the cumulative distribution function (CDF) of successful decoding at different M's. Fig. 5 shows the CDF from 5 to 25dB. The curves for 26dB and above are very close to each other and are not shown here. Then we take number of symbols corresponding to the 10% percentile as the demodulation threshold as shown in Table 1. These settings are adopted in our system implementation.

5. IMPLEMENTATION AND EVALUATION

We design and implement a rate adaptation system, called SRA, based on the proposed rate compatible modulation and OFDM physical layer.

5.1 Implementation

The protocol of SRA can be depicted by Fig. 6. At sender end, information bits are block-wise mapped to RCM symbols. Then, the RCM symbols are pair-wisely transmitted as in-phase and quadrature-phase components of QAM modulation. For each block of information bits, the sender progressively transmits modulation signals with a certain step until an acknowledgement is received or the maximum number is achieved. At receiver end, the RCM symbols are accumulated for demodulation after the symbol number exceeds a pre-defined threshold. This threshold is shown in Table 1. If the demodulated bits pass the CRC check, an acknowledgement is delivered to the sender. Otherwise, more RCM symbols need to be collected for another demodulation.

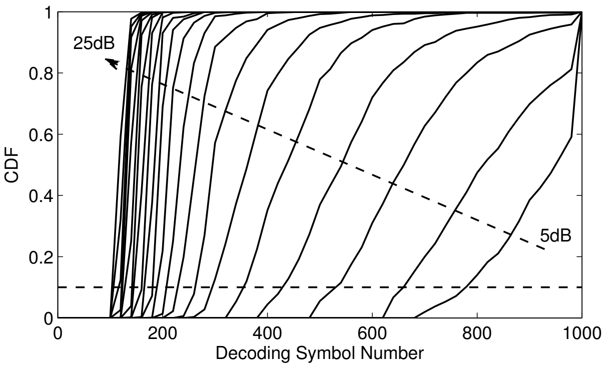


Figure 5: RCM demodulation threshold

SNR (dB)	5	6	7	8	9	10	11
Threshold	760	640	520	420	340	280	260
SNR (dB)	12	13	14	15	16	17	18
Threshold	220	200	180	180	160	160	140
SNR (dB)	19	20	21	22	23	24	25
Threshold	140	120	120	120	100	100	100

Table 1: Demodulation threshold at different channel SNRs

Based on the above protocol, we implement SRA on OFDM PHY defined in IEEE 802.11a, which is illustrated in Fig. 7. Specifically, the channel is divided into 64 subcarriers and 48 of them are used to transmit modulation signals. To reduce the overhead of PLCP header, we use 150 OFDM symbols in each PLCP frame for data transmission. Therefore, the total number of modulation signals in each transmission is 7200. Since RCM is a block based modulation, the data from MAC layer will be divided into bit blocks before transmission. In current SRA system, 400 bits block size is adopted. The progressive transmission step is 10 modulation signals and the maximum number of modulation signals is set to 800 which is equal to the lowest rate 0.5 bits/s/Hz in 802.11a for PLCP header transmission. Therefore, we transmit 720 bit blocks in parallel pipeline style and need to maintain a 36KB buffer at both sender and receiver end.

5.2 Evaluation Methodology

We perform trace-driven evaluation on the software radio platform SORA [20]. SORA is a programmable software radio platform built on commodity general purpose PCs. We perform trace-driven experiments instead of real-time experiments mainly for two reasons. First, as wireless channel is changing over time, it is not possible to fairly compare

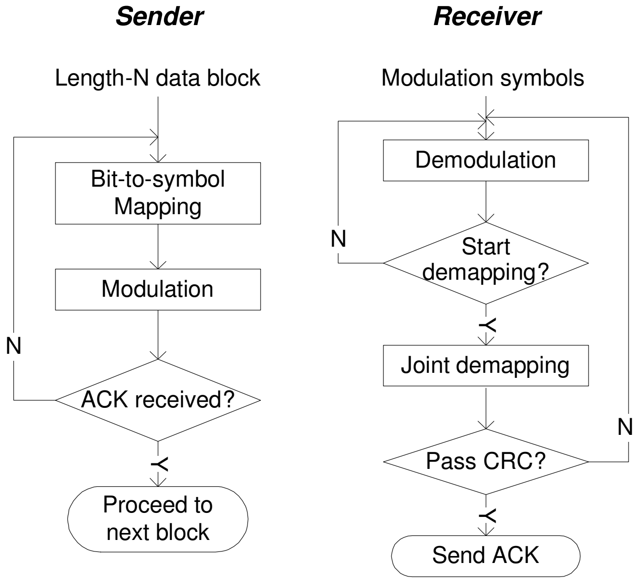


Figure 6: Data transmission flow charts at the sender and the receiver in SRA system

several schemes under exactly the same channel condition. Second, SORA does not allow for parallel RCM demodulation at present, so our implementation is not real-time. Real-time demodulation implementation will be one of our future works.

The channel condition is traced by transmitting uncoded BPSK symbols. This is valid because Sen et al. [16] have demonstrated that the dispersions on transmitted symbols are not influenced by the modulation scheme and is only the function of channel condition. We create four distinct channel conditions by combining static/mobile and Line-of-Sight (LOS)/Non-Line-of-Sight (NLOS). In all scenarios, the sender location is fixed. The receiver keeps motionless in static settings and moves at walking speed in mobile settings. For each scenario, one minute data is traced including the fading gain and the noise on 4 pilots and 48 carriers.

In each scenario, our SRA system is compared against with the three reference systems. System1 is an AMC scheme based on 802.11a physical layer. The eight modulation and coding combinations are the same as in the Simulation section. System2 is HARQ-CTC combined with AMC. According to the simulation results shown in Fig. 3, HARQ-CTC should switch to QPSK modulation when the SNR is below 9dB, and should switch to 64QAM when the SNR is above 15dB. In the SNR region between 9dB and 15dB, 16QAM modulation is used. System3 is HARQ-Raptor combined with AMC. The switching of modulation in all these three systems is determined by the channel estimation based on previous transmission.

In all systems, we use the SNR estimation algorithm proposed by Rahul et al. [12]. Similar to the setting in simulation, we assume that both channel SNR feedback and ACK are immediately available.

5.3 Results

In all four scenarios, SRA significantly outperforms the other schemes. Fig. 8 shows the SNR statistics of the four scenarios. Fig. 9 presents the CDF (cumulative distribution function) of goodput for all schemes. Table 2 summarizes the average goodput of SRA and the three reference systems.

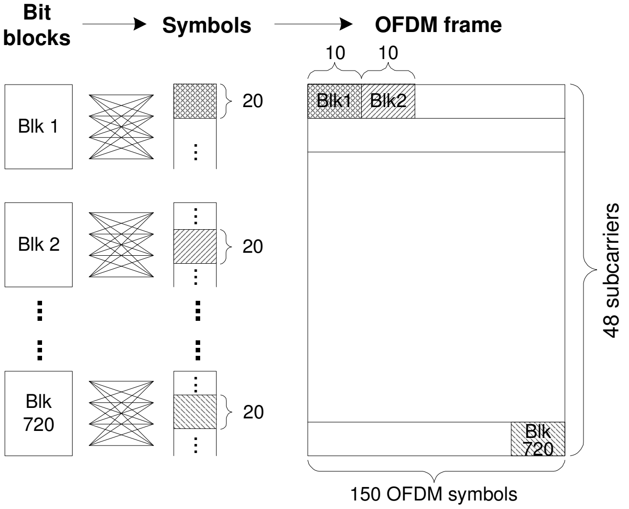


Figure 7: SRA implementation on OFDM

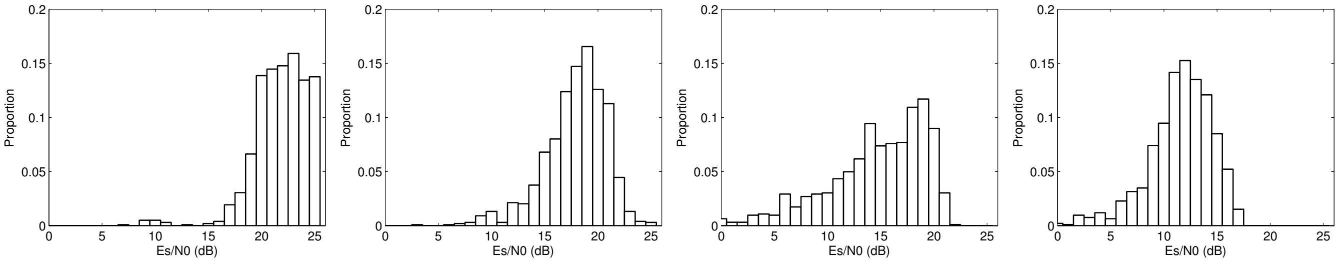
bits/s/Hz	LOS	LOS_M	NLOS	NLOS_M
SRA	5.50	4.71	3.78	2.89
System1	4.04	3.03	2.27	1.56
System2	5.06	4.02	2.93	2.43
System3	4.71	3.77	2.63	2.29

Table 2: Average Rate of each scenario

In LOS static scenario, the channel is very stable and the SNR is in the range of 17dB to 25dB. All schemes perform fairly well. The poor performance of System1 in this scenario is mainly due to two reasons. First, 802.11a only provides a discrete rate set. The achievable rates are stair-shaped. By looking into the trace results, we find that System1 selects rate 3.0 (16QAM, 3/4) and 4.5 (64QAM, 3/4) in most time slots, so the average goodput is around 4 bit/s/Hz. Second, convolutional code cannot achieve the optimal coding performance so the achievable rates will be lower than using LDPC code or Turbo code under the same channel condition. In the other three scenarios, the performance gain of SRA over System1 becomes more significant. This is due to the fact that the channel varies dramatically and the imprecise estimation (prediction) misleads the rate selection algorithm. In NLOS mobile scenario, the gain of SRA over System1 is over 80%.

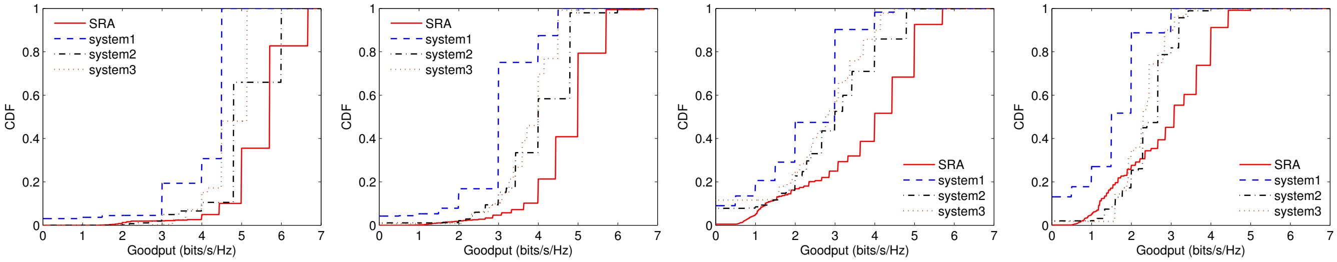
Comparing Fig. 8(a)(b)(c), we find that the dynamic range is gradually extended. The gain of SRA over System2 and System3 also gradually increases. This is because the advantage of using a fixed modulation in SRA becomes prominent when the other two systems have to switch between different settings. In NLOS scenario, SRA achieves 28.8% gain over System2 and 43.8% gain over System3.

In the last NLOS mobile scenario, the channel SNR varies between 2dB and 17dB. As shown in simulations, when channel SNR is below 8dB, SRA has a little performance loss with respect to HARQ schemes using low rate modulations. Despite of this, SRA manages to achieve around 18%



(a) LOS (b) LOS Mobile (c) NLOS (d) NLOS Mobile

Figure 8: PDF of Es/N0



(a) LOS (b) LOS Mobile (c) NLOS (d) NLOS Mobile

Figure 9: CDF of Goodput

gain over System2 and around 26% gain over System3 in average throughput.

The CDF of goodput shown in Fig. 9 presents the details of rate selection. The results show that SRA successfully achieves our design goal of smooth and efficient rate adaptation over a broad dynamic range.

6. CONCLUSION

We have described a novel rate compatible modulation scheme that successfully achieves smooth and efficient rate adaptation over a broad dynamic range of channel conditions. By imposing the rate compatible restriction on modulation, instead of channel coding, we successfully extend the dynamic range of rate adaptation without compromising the spectral efficiency. The seamless adaptation is achieved through an incremental symbol generation process with fine-grained bit energy allocation.

We carry out extensive simulations and obtain the achievable rates of proposed RCM for the typical operation range of channel SNR. Results show that RCM covers a much larger dynamic range than HARQ systems based on rate compatible channel code. We have also designed and implemented a rate adaptation system SRA over OFDM. The evaluation on SORA testbed confirms the advantage of SRA when channel condition varies dramatically. In NLOS scenario, SRA outperforms the combined HARQ with AMC scheme by 35% on average. These promising results validate the practical use of the proposed RCM in real wireless environments.

7. REFERENCES

[1] IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11-2007*, 2007.

[2] IEEE Standard for Local and metropolitan area networks - Part 16: Air Interface for Broadband Wireless Access Systems. *IEEE Std 802.16-2009*, May 2009.

[3] 3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Evolved Universal Terrestrial Radio Access (E-UTRA);Multiplexing and channel coding. *3GPP TS 36.212*, Sep 2010.

[4] D. Baron, S. Sarvotham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *Signal Processing, IEEE Transactions on*, 58(1):269 –280, Jan. 2010.

[5] J. D. Brown, S. Pasupathy, and K. N. Plataniotis. Adaptive demodulation using rateless erasure codes. *Communications, IEEE Transactions on*, 54:1574–1585, 2006.

[6] A. Gudipati and S. Katti. Automatic rate adaptation. In *ACM Hotnets'10*, 2010.

[7] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multi-hop wireless networks. In *Proc. of ACM MobiCom*, pages 236–251, 2001.

[8] G. Judd, X. Wang, and P. Steenkiste. Efficient channel-aware rate adaptation in dynamic environments. In *Proc. of ACM MobiSys*, pages 118–131, 2008.

[9] R. T. Morris, J. C. Bicket, and J. C. Bicket. Bit-rate selection in wireless networks. Technical report, Masterąŕs thesis, MIT, 2005.

[10] W. H. Mow. Universal lattice decoding: principle and

recent advances. *Wireless Communications and Mobile Computing*, 3:553 – 569, Aug. 2003.

[11] D. Qiao, S. Member, S. Choi, and K. G. Shin. Goodput analysis and link adaption for ieee 802.11a wireless lans. *IEEE Trans. on Mobile Computing*, 1:278–292, 2002.

[12] H. Rahul, F. Edalat, D. Katabi, and C. Sodini. Frequency-aware rate adaptation and mac protocols. In *Proc. of ACM MobiCom*, pages 193–204, 2009.

[13] D. Rowitch and L. Milstein. On the performance of hybrid fec/arq systems using rate compatible punctured turbo (rcpt) codes. *Communications, IEEE Transactions on*, 48(6):948 –959, June 2000.

[14] D. N. Rowitch and L. B. Milstein. On the performance of hybrid fec/arq systems using rate compatible punctured turbo (rcpt) codes. *Communications, IEEE Transactions on*, 48(6):948–959, Jun 2000.

[15] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Accurate: constellation based rate estimation in wireless networks. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 12–12, Berkeley, CA, USA, 2010. USENIX Association.

[16] S. Sen, N. Santhapuri, R. R. Choudhury, and S. Nelakuditi. Accurate: constellation based rate estimation in wireless networks. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 12–12, Berkeley, CA, USA, 2010. USENIX Association.

[17] A. Shokrollahi. Raptor codes. *Networking, IEEE/ACM Transactions on*, 14:2551–2567, 2006.

[18] B. Sklar. *Digital communications: fundamentals and applications.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1988.

[19] E. Soljanin, N. Varnica, and P. Whiting. Punctured vs rateless codes for hybrid arq. In *Information Theory Workshop, 2006. ITW '06 Punta del Este. IEEE*, pages 155 –159, 2006.

[20] K. Tan, J. Zhang, J. Fang, H. Liu, Y. Ye, S. Wang, Y. Zhang, H. Wu, W. Wang, and G. M. Voelker. Sora: high performance software radio using general purpose multi-core processors. In *NSDI'09*, pages 75–90. USENIX Association, 2009.

[21] A. Venkiah, C. Poulliat, and D. Declercq. Jointly decoded raptor codes: analysis and design for the biawgn channel. *EURASIP J. Wirel. Commun. Netw.*, 2009:16:1–16:11, January 2009.

[22] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. In *Proc. of ACM SIGCOMM*, pages 3–14, 2009.

[23] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *Proc. of ACM MobiCom*, pages 146–157, 2006.

APPENDIX

A. RCM DEMODULATION ALGORITHM

Algorithm 1 Demodulation

```
Initialization:
$p_j = p(b_j = 1) = 0.5$; $q_{ji}^{(0)}(0) = 1 - p_j$; $q_{ji}^{(0)}(1) = p_j$
Iterations: {$T$ is the maximum number of iterations}
while $t < T$ do
    Horizontal Processing:
    for $i = 1$ to $M$ do
        $r_{ij}^{(t)}(0) = p(b_j = 0|u'_i) = p(U_j = u'_i)$
        $r_{ij}^{(t)}(1) = p(b_j = 1|u'_i) = p(U_j = u'_i - w_l)$
        {where $u'_i = \sum_{l=1}^{L} w_l \cdot b_{i_l} + e_i$ is the received symbol}
        {$U_j = u'_i - w_l \cdot b_j$ and $j = i_l$}
        Distribution $P(U_j) = \{\bigotimes_{i_k, 0<k\leq L, k\neq l} P(w_k \cdot b_{i_k})\} \otimes P(e_i)$
        {where $p(w_k \cdot b_{i_k} = 0) = q_{ji}^{(t-1)}(0)$, $p(w_k \cdot b_{i_k} = w_k) = q_{ji}^{(t-1)}(1)$}
        {$p(e_i) \sim N(0, \sigma^2)$, $\otimes$ is convolutional product}
    end for
    Vertical Processing:
    for $j = 1$ to $N$ do
        $q_{ji}^{(t)}(0) = C_{ji} \cdot (1 - p_j) \cdot \prod_{k \in T_j \setminus i} r_{kj}^{(t)}(0)$
        $q_{ji}^{(t)}(1) = C_{ji} \cdot p_j \cdot \prod_{k \in T_j \setminus i} r_{kj}^{(t)}(1)$
        {$C_{ji}$ is the normalization factor which ensures $q_{ji}^{(t)}(0) + q_{ji}^{(t)}(1) = 1$}
        {$T_j$ is the index set of the check nodes connected with variable node $j$}
    end for
end while
Output: {Hard Decision}
$b_j = 0$, if $q_j^{(T)}(0) \geq q_j^{(T)}(1)$; Otherwise, $b_j = 1$
{$q_j^{(T)}(0) = C_j \cdot (1 - p_j) \cdot \prod_{k \in T_j} r_{kj}^{(T)}(0)$}
{$q_j^{(T)}(1) = C_j \cdot p_j \cdot \prod_{k \in T_j} r_{kj}^{(T)}(1)$}
```