Wireless Rate Adaptation via Smart Pilot

Lu Wang*, Xiaoke Qi†, Jiang Xiao*, Kaishun Wu*‡, Mounir Hamdi* and Qian Zhang*
*Hong Kong University of Science and Technology
† University of Chinese Academy of Sciences
‡ Shenzhen University
{wanglu,hamdi,qianzh}@cse.ust.hk xiaoke.qi@nlpr.ia.ac.cn {jiangxiao, kwinson}@ust.hk

***Abstract*—Rate adaptation is an essential component in today's wireless standards, for its ability to adaptively approach the channel capacity, and maximize the system throughput. The difficulty in rate adaptation stems from estimating the optimal data rate in a fluctuated channel. Previous wisdoms leverage PHY layer information for rate estimation, such as SoftPHY hints or Channel State Information. These information solely comes from one same layer, which are insufficient to track the optimal data rate. We observe that by investigating the information in both PHY layer decoder and upper layer protocol headers, more pilots can be exploited to estimate the optimal data rate across both time and frequency domain. These smart pilots help remove the residual channel effect and calibrate the CSI with minimum overhead. Based on the calibrated CSI, we propose a novel greedy rate selection algorithm to harness frequency diversity, which obtains the optimal data rate over all the subcarriers. Our experiments on GNU radio testbed show that SmartPilot quickly tracks the link variance, and reduces the residual channel effect by 87%. Further, the trace driven simulation reveals that greedy rate selection algorithm predicts the data rate as good as the optimal rate adaptation algorithms for 802.11 standards.**

I. Introduction

Today's wireless communications are facing great challenges to meet the increasing user demands for higher data rate and more coverage. The latest 802.11 standards offer date rates ranging from 6Mbps to 600Mbps [1]. Accordingly, rate adaptation protocols are adopted to dynamically adjust the data rate on the basis of the channel quality. However, wireless links often experience a variety of impairments. These impairments add noise, introduce bit errors, or otherwise distort the transmitted signal, making the wireless links extremely unstable both in time domain and frequency domain [2]. This instability brings in a great challenge to rate adaptation design. How to predict such a fluctuated channel and estimate the optimal transmission data rate remains a critical research topic.

There are primarily two kinds of metrics for rate adaptation: packet loss rate (PLR) and signal to noise ratio (SNR)- bit error rate (BER). SampleRate [3] and RRAA [4] are two representatives of loss-triggered rate adaptation, which rely on PLR to infer the channel condition. The PLR is calculated by tens or hundreds of frame transmissions, making it not responsive to channel variance. Instead of operating on frame-level, SoftRate [5] utilizes SoftPHY hints (e.g., bit confidence level) to obtain per-packet BER, and picks the good bit rates on per-packet basis. Thus it is much more adaptive to the rapidly time-varying channel. Another short timescale metric is SNR retrieved on each packet reception [6], and is commonly measured by the received signal strength indication (RSSI). Yet RSSI is known to be coarse and insufficient, especially in frequency selective fading channel. To better reflect the channel variation in frequency domain, effective SNR (ESNR) is proposed in [7], which is computed by channel state information (CSI). ESNR serves as a promising metric for rate adaptation, since it takes frequency diversity into consideration. However, there still exists certain gap between CSI and the real channel status in the time varying and frequency selective fading channel.

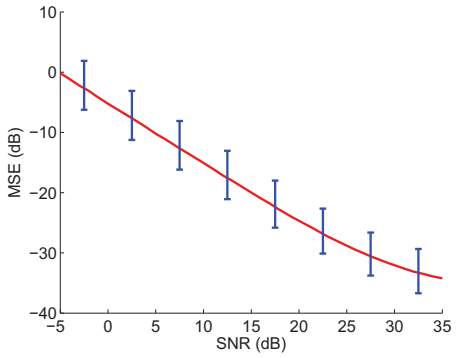


Fig. 1: Illustration of residual channel effect, which is defined as the Mean Square Error (MSE) between the real channel and the channel measured with 802.11 preamble. This residual channel effect keeps us away from tracking the ideal data rate.

The reason why we fail to track the optimal data rate falls into two aspects. The first one is inaccurate channel estimation, which directly links to the inappropriate rate selection choice. Current rate adaptation protocols are not capable of obtaining the accurate channel quality. The main reason is the information they acquire for channel estimation is limited. In today's 802.11 standards, only preamble and a few inserted pilot symbols are utilized in each frame. Thus there always leaves a residual channel effect that cannot be truly removed. We illustrate the residual channel effect in Fig. 1, which is defined as the Mean Square Error (MSE) between the real channel and the channel measured with 802.11 preamble. This difference keeps us away from tracking the ideal rate. Inserting extra pilots may be a solution, yet it decreases the effective data rate. The second aspect lies in the inability to determine the appropriate data rate under frequency diversity. Multipath effect results in various subcarrier quality. Yet none of the existing metrics, including ESNR and BER, have differentiated the subcarrier impact. When some subcarriers experience deep

978-1-4799-6204-4/14 $31.00 © 2014 IEEE
DOI 10.1109/ICNP.2014.64

fading, they will conservatively select a lower data rate to ensure transmissions on all subcarriers. However, the subcarriers without fading will lose the opportunity for higher data rate. Adaptive modulation and coding with different data rates at each subcarrier could solve the problem, yet it will complicate the system. Thus how to choose an appreciate data rate to maximize the overall throughput still remains a concern.

The above observations motivate us to present SmartPilot, a wireless rate adaptation protocol that exploits potential pilots across multiple layers to track the channel variance, and selects the optimal data rate over a frequency selective fading channel. The SmartPilot design stems from two facts. First, at PHY layer, the decoded bits with relatively high confidence level are regarded reliable. They can be selected as pilots to help remove the residual channel effect and approach the real channel status. Second, in 802.11 protocol headers, there are a great number of fields that have fixed values within a certain link [8]. These fields can be harnessed as pilots to further improve the channel estimation. The former pilots are termed as Soft Pilots, and the latter ones are Hard Pilots. These smart pilots serve as a built-in strategy to calibrate the CSI, and better approach the real channel status with minimum overhead. We further present a novel greedy rate selection (GRS) algorithm with based on the calibrated CSI (CCSI). Unlike the previous methods that weight all the subcarriers as equal, we strategically leave the subcarriers with deep fading, and choose those with better channel quality to calculate the optimal data rate.

We have implemented SmartPilot protocol in a GNU Radio testbed [9]. Experiments with our software radio prototype verify that SmartPilot can reduce the channel residual effect by 87% with minimum overhead. We also conduct trace-driven simulations to evaluate the greedy rate selection over SmartPilot. The results demonstrate that it improves the throughput by $2.1\times$, $1.9\times$and $1.8\times$ compared with SampleRate, SoftRate and ESNR. The performance gain stems from SmartPilot's ability to quickly track the channel variance, and predict the optimal data rate against multi-path effect. In summary, the main contributions of this paper over the existing rate adaptation protocols are as follows:

- We observe the residual channel effect in existing rate adaptation protocols, and reduce it by leveraging SmartPilot with minimum overhead. To the best of our knowledge, this is the first work to exploit decoding data and header bits as pilots to improve rate adaptation.
- We present a novel greedy rate selection algorithm based on calibrated CSI, which differentiates the "good " and the "bad "subcarriers, and leverages frequency diversity to select the optimal data rate for transmission.
- We implement SmartPilot on a GNU Radio testbed. Experimental results verify that SmartPilot reduces the channel residual effect by 87%. We also conduct trace-driven simulations to demonstrate the effectiveness of greedy rate selection algorithm over smart pilots.

II. Related Work

The existing wisdoms of rate adaption can be classified into two categories: reactive rate adaptation and rateless codes. The former one is also called sender-based rate adaptation, where the sender adjusts the data rates according to a certain metric. In SampleRate [3] and RRAA [4], packet loss rate (PLR) is used at the sender side to infer the channel condition. Since FRR is calculated by tens or hundreds of frame transmissions, these two approaches are not so responsive to channel variance. On the other hand, SoftRate [5] utilizes SoftPHY hints to obtain per-packet BER, and conducts rate adaptation on packet level. BER is much more adaptive to the rapidly varying channel, yet it has certain drawbacks as a rate adaptation metric, e.g., when BER = 0, it is difficult to determine to which level we should increase the data rate. Unlike SoftRate, SmartPilot utilizes calibrated CSI for rate selection to avoid the drawbacks. Also, it is noticed that SoftRate leverages the information from data bits to compute BER. The problem is, not all the soft information is reliable due to channel variance. SmartPilot discovers and only leverages the data bits that are proved to be reliable. Therefore, it can better track the channel quality. Another short timescale metric is SNR measured through RSSI on each packet reception [6]. Yet RSSI is known to be coarse and insufficient, especially in frequency selective fading channel. To better reflect the channel variation, ESNR [7] proposes effective SNR computed by CSI for rate adaptation, which takes frequency diversity into consideration. Even so, all the previous methods fail to track the optimal data rate, since the information they acquire for rate estimation is not sufficient enough, especially in a time varying channel with frequency selective fading. Also, the rate selection mechanisms do not take frequency diversity into consideration. On the contrary, SmartPilot investigates more pilots to estimate the data rate with minimum overhead. Thus it can better track the link variance. It also differentiates the subcarrier quality for rate selection, and thus maximizes the overall throughput.

The latter rate adaptation class is receiver-based [10]–[13], where the sender keeps transmitting packets at a fixed data rate, and the receiver decodes the packets adaptively according to its channel condition. Rateless codes, such as Luby Transform (LT) codes [10] and Raptor codes [11], have displayed a desirable paradigm for the receiver-based rate adaptation. The sender generates a potentially limitless number of output symbols from a fixed number of message symbols. The receiver continue collecting the symbols until the entire message is successfully decoded. Until now, several work has brought rateless codes into wireless channels. In [12], Strider codes adopt a layered approach to generate linear combinations of symbols in a rateless manner. Yet the design becomes complicated as the number of layers increases. On the other hand, Spinal codes [13] utilize hash function and random number generator to produce rateless symbols, which simplify the design compared with layered approach. The main drawback of these codes is how to conduct cost-effective

feedback. Also, the decoding complexity makes rateless codes not so practical right now. As a result, we only enable smart pilots in sender-based rate adaptation protocols.

III. Residual Channel Effect

Before we jump into the design of SmartPilot, we first introduce the idea of residual channel effect as the primary obstacle in rate adaptation protocols. Residual channel effect (RCE) is defined as the difference between the estimated channel status and the ground truth. Due to channel variance and limited information, the receiver cannot truly remove RCE, which keeps us away from tracking the ideal rate. We will present the detailed analysis of residual channel effect in the following paragraphs.

In a wireless time-varying channel, transmitted signal suffers from frequency selective fading [14]. To overcome this distortion, conventional method estimates channel response from preamble, and compensates its effect for each subcarrier in one OFDM symbol . Least square (LS) channel estimation is widely used for its simplicity, yet it will suffer from inaccuracy caused by noise, especially under low SNRs. To quantify this effect, we define residual channel effect as $E\{||\hat{\mathbf{h}} - \mathbf{h}||^2\}$, where $\hat{\mathbf{h}}$ and $\mathbf{h}$ denote the estimated channel response and ground truth channel response, respectively.

In 802.11 standards, channel estimation is conducted as follows: two identical pseudo-noise (PN) sequences are designated as preamble, each denoted as $\mathbf{x}$ with length N_d. Assuming $\mathbf{y_0^1}$ and $\mathbf{y_0^2}$ are the received sequences of the preamble, then they can be expressed as:

$$\begin{aligned} y_0^1(p) &= h(p)x(p) + n_0^1(p), \\ y_0^2(p) &= h(p)x(p) + n_0^2(p), p = 1, ..., N_d, \end{aligned} \tag{1}$$

where n_0^1 and n_0^2 are the Gaussian random variables with variance σ_n^2. To acquire $\hat{\mathbf{h}}$, LS channel estimation calculates the following equation:

$$\hat{h}_0(p) = \frac{y_0^1(p) + y_0^2(p)}{2x(p)} = h(p) + n^{'}(p), \tag{2}$$

where $n^{'}(p) = (n_0^1(p) + n_0^2(p))/2$. Therefore, the variance is calculated as

$$\sigma_{n'}^2 = E\left\{n^{'}(p)^2\right\} = \frac{1}{4}E\left\{(n_0^1(p) + n_0^2(p))^2\right\} = \frac{\sigma_n^2}{2}. \tag{3}$$

According to Equation (2), the residual channel effect is measured as:

$$\begin{aligned} \text{mse}_0 &= E\{||\hat{\mathbf{h}}_\mathbf{0} - \mathbf{h}||^2\} \\ &= \frac{1}{N_d}\sum_{p=1}^{N_d}|\hat{h}(p) - h(p)|^2 = \sigma_n^2/2. \end{aligned} \tag{4}$$

From Equation (4), we reveal that under poor channel conditions, e.g., when SNR is low, the residual channel effect becomes severe, and amplifies the gap between real channel status and the channel status we obtained from channel estimation. Since reliable channel estimation provides appropriate metric for rate selection, minimizing the residual channel effect will definitely improve the performance of rate adaptation protocols.

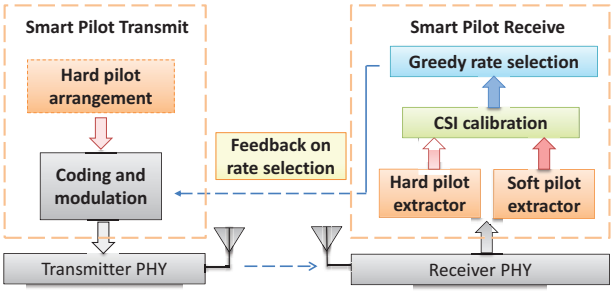


Fig. 2: The block diagrams of SmartPilot sender and receiver. Four important components are included: a) Soft Pilot Extractor, b) Hard Pilot Extractor, c) CSI Calibration, and d)Greedy Rate Selection.

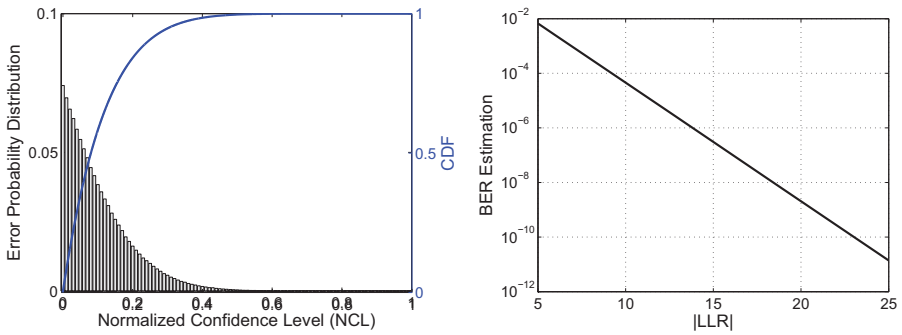
IV. SmartPilot Design

In this section, we describe the overall architecture of SmartPilot. SmartPilot provides the underlying functionality for accurate rate selection in 802.11 standard, and is compatible with the existing error correcting codes and error recovery schemes. By exploiting as many pilots as possible to approach the real channel status, it aims at selecting the optimal data rate against multipath effect. First, we present an overview of SmartPilot along with design challenges. Detailed designs, including pilot extraction, CSI calibration and greedy rate selection, are demonstrated later to see how we address these challenges.

A. Overview and Design Challenge

Before we start, some necessary assumptions are summarized as follows: 1) Currently we only consider point to point transmission, e.g., transmission between a sender-receiver pair. 2) SmartPilot is compatible with the existing error correcting codes and error recovery schemes. Here we apply it on prevailing 802.11 LDPC (Low Density Parity Check) codes with Hybrid ARQ (Automatic Repeat reQuest) for illustration.

With these assumptions in mind, we propose SmartPilot for wireless rate adaption. Unlike the previous rate adaptation protocols that treat PSDU (PLCP Service Data Unit) as transparent, SmartPilot investigates the reliable data bits in PSDU as pilots to calibrate the CSI and approach the real channel status. To be specific, SmartPilot exploits two kinds of data bits as pilots. First, from PHY layer decoder, some data bits have relatively high confidence levels. These data bits can be considered reliable and selected as known pilots after decoding. We call them soft pilots since they are extracted using softPHY hints. Second, 802.11 PDSU carries numerous upper layer protocol headers, e.g., MAC header, logical link header, network header, etc.. Some fields in these headers often have fixed bit values within a certain link, such as the source and destination MAC addresses, and the frame and service types. These bits could be extracted as hard pilots during



(a) (b)

Fig. 3: The relationship between bit errors and LLRs. (a) The error probability distribution (b) BER estimation based on SoftRate.

each transmission, and leverage for channel estimation and rate selection.

The fundamental idea of SmartPilot is: *to exploit as many pilots as possible from transparent data unit, and harness them to estimate the optimal data rate with minimum overhead.* The basic idea seems simple and efficient, yet there remains several challenges for implementation. First, how to utilize SoftPHY hints to choose the proper soft pilots remains a great challenge. As we try to extract as many pilots as possible, there always exists a tradeoff between quantity and quality. Second, unlike soft pilots that naturally spread within a frame across all the subcarriers, hard pilots are mainly located in protocol headers. They may have limited contribution to estimate the entire channel, and need to be carefully spread out within a transmission. Third, after we obtain smart pilots and use them to calibrate the CSI, we need to make full use of the CCSI, and choose the optimal data rate in a frequency selective fading channel.

As shown in Fig. 2, SmartPilot has four components to address the above challenges: **Soft Pilot Extractor** that selects soft pilots with guaranteed reliability, **Hard Pilot Extractor** that extracts hard pilots without extra control overhead, **CSI Calibration** that utilizes smart pilots to remove the residual channel effect and approach the real channel status, and **Greedy Rate Selection** that leverages CCSI to compute the highest available date rate in a frequency selective fading channel.

B. Smart Pilot Extraction

1) Soft Pilot Extraction: Soft pilots are the decoded bits extracted from PHY layer decoder using SoftPHY hints, e.g., posteriori Log-Likelihood Ratios (LLRs) [15]. We obtain the LLRs from the *maximum likelihood*(ML) or *maximum a posteriori probability*(MAP) decoder for each received bit [16], and calculate it using the following equation:

$$L_0(i) = \frac{2y_i}{\sigma^2}, i = 1, ..., N. \quad (5)$$

The critical problem in soft pilot extraction is how to ensure the reliability of soft pilots using LLRs. Due to channel impairments such as noise variation, channel multipath, and collision, LLRs may become unreliable and cannot represent the real confidence level of the bits. Therefore, we conduct simulations to investigate how LLR reflects the error probability of a decoded bit. Fig. 3(a) depicts the relationship between normalized confidence level (NCL) and the corresponding error probability distribution, where NCL is defined as:

$$NCL = \frac{|LLR|}{\max(|LLR|)} \quad (6)$$

From Fig. 3(a) we observe that, the erroneous bits have relatively low NCL, and the bits with higher NCL turn out to be always reliable. To be specific, when NCL is below 0.5, the decoded bits have high probability to be erroneous. After NCL exceeds a certain threshold, e.g., 0.8, the error probability quickly drop to 0, indicating that the decoded bits are correct with no errors. Therefore, we extract the decoded bits that have relative high NCL as soft pilots, and utilize them for channel estimation. According to our simulations, the NCL threshold, which is also termed relative threshold T_1, is better to be set to 0.8. That is, the decoded bits that have NCLs values more than 0.8 will be selected as soft pilots. In order to obtain more soft pilots, another threshold, absolute threshold T_2 is proposed. T_2 is based on absolute LLR value. According to SoftRate, the error probability has an approximately exponential decay relationship with $|LLR|$. We demonstrate this relationship in Fig. 3(b). When a decoded bit has $|LLR|$ value higher than 20, its BER is lower than 10^{-9}. The BER is small enough to be trusted. Thus we set T_2 to 20, and extract the decoded bits with $|LLR|$ values higher than 20 as soft pilots. These two thresholds help us to filter more reliable soft pilots for CSI calibration, and thus the results can be more accurate. It is noted that the threshold choice has certain impact on the performance of soft pilot extraction. Currently we use empirical values to determine the two thresholds. More sophisticated techniques can be incorporated to choose more appropriate thresholds according to the system requirements [17]. We leave it as future work.

We divided the decoding process into several loops. In each loop, decoded bits with $NCL \geq 0.8$ and $|LLR| \geq 20$ are

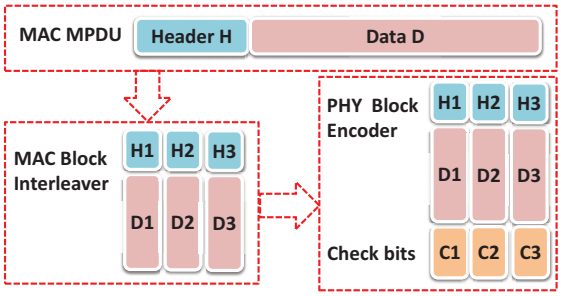


Fig. 4: Illustration of MAC layer block interleaver, from MAC MPSU to PHY layer coded message.

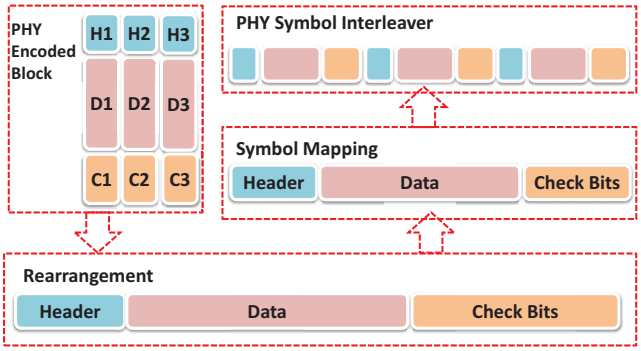


Fig. 5: Illustration of PHY layer symbol interleaver, which operates on the PHY layer coded message.

extracted as soft pilots. These soft pilots are also regarded as the known bits in the next loops to improve the decoding performance. During pilot extraction, the pilot values are the hard decisions of the outputs $|LLR|$. However, this method is likely to extract erroneous bits. To avoid this situation, we make the decision based on the decoding status. If the decoding is successful, we choose the hard decisions from current loop as the decoded bits. Otherwise, we choose the values from the previous loops as the decoded bits. This scheme is motivated by the observation that, the extraction in the first loops is always considerate realizable, and decoded bits in different loops are reliable in different positions. Therefore, if we average the decoded bits, the errors can be suppressed. The detailed soft pilot extractor is summarized in Algorithm 1. Our experiments in Sec. V-B also prove the efficiency of this scheme.

Algorithm 1 Soft Pilot Extractor

$\boldsymbol{p} \leftarrow [], \boldsymbol{v} \leftarrow [], n_p \leftarrow 0$ ▷ Initialization, $\boldsymbol{p}$: pilot position; $\boldsymbol{v}$: pilot value; n_p: number of pilots.
for $loop$ = 1 **to** L **do**
  [LLR, d] = channel_decoder(data, $\boldsymbol{p}, \boldsymbol{v}, n_p$);
  calculate NCL;
  if successful recovery **then**
    exit $loop$;
  end if
  $\boldsymbol{p}$ = [$\boldsymbol{p} \bigcup$ find $(NCL > T_1) \bigcup$ find$(|LLR| > T_2)$];
  $\boldsymbol{v} = d(\boldsymbol{p})$;
  n_p = length($\boldsymbol{p}$);
end for

2) Hard Pilot Extraction: Different upper layer protocols have various packet semantics. Since hard pilots exhibit quite distinct features and highly depend on data traffic types, it is not practical for the sender to inform the exact pilot locations to the receiver. To simplify the design and minimize the overhead, we leverage the idea of bit bias to conduct pilot extraction at the receiver side. Bit bias is the high probability that a bit takes value of 0 or 1 in a window of packets [8].

With a window of packet of $T_{m,n}$ in a certain link:

$$T_{m,n} = \begin{bmatrix} b_{1,1} & \cdots & b_{1,n} \\ \vdots & \ddots & \vdots \\ b_{m,1} & \cdots & b_{m,n} \end{bmatrix}, \tag{7}$$

the bit bias $\beta_i(T_{m,n})$ of bit i can be computed as:

$$\beta_i(T_{m,n}) = 2 \times \left| \frac{\sum_{k=1}^{m} b_{k,i}}{m} - \frac{1}{2} \right| \tag{8}$$

where m is history size of the received packets, and n is the header length. $b_{i,j}$ defines the bit value of the i^{th} bit in the j^{th} packet. If bit i is fixed to the value 0 or 1 within m packets, $\beta_i(T_{m,n}) = 1$, and then it will be selected as hard pilot and stored in a *hard-pilot* buffer for this link. Otherwise, $\beta_i(T_{m,n}) = 0$. The receiver will keep sniffing the received packets at the running time, and compute the bit bias for each bit located in the first 80 bytes of each packet. This ensures that we can include all protocol headers from MAC layer to the transport layer.

During each transmission, the receiver will first update the *hard-pilot* buffer before using them to calibrate CSI. It is possible that the pilots used for channel estimation does not match the actual received bits. This is called pilot misprediction and will lead to inaccurate channel estimation. Here we incorporate the block checksum to guarantee the accuracy of the extracted pilots. To be specific, the first 80 bytes of the header bits are divided into two blocks. Each block is then calculated and appended a checksum. At the receiver side, if the extracted hard pilots within a block do not match their checksum fails, the receiver will abandon the pilots in this block. The process continues until the pilots are updated using a correctly received block.

As we mentioned before, the hard pilots are extracted from the protocol headers, e.g., within the first 80 bytes of the packets. Therefore, they are not capable of tracking the channel variation of the whole packet. To make them useful, we propose two interleavers at the sender side: *MAC block interleaver* and *PHY symbol interleaver*. MAC interleaver aims to distribute the header bits evenly within the entire

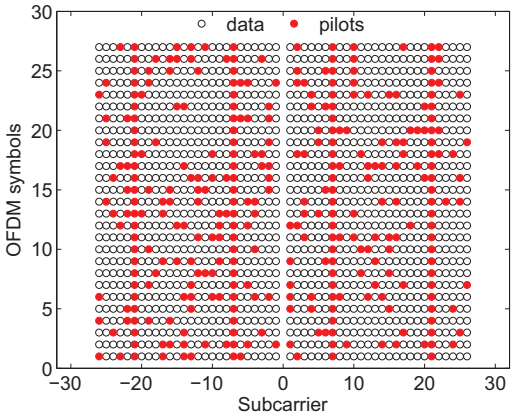


Fig. 6: The hard pilot positions after OFDM modulation, using SIGCOMM'08 trace [18].

Message Protocol Data Unit (MPDU). It divides the header of MPDU into several blocks, and inserts each block into a subsequent data block that divided by the encoder (e.g., LDPC encoder). In this way, we can ensure that each coded block has at least one header block after PHY encoding, and thus the header block can be spread over the entire packet. As illustrated in Fig. 4, the header and data units in MPDU have been divided into three blocks respectively. The row-column interleaver assigns each data block with a header block. After the assignment, they are feeded into PHY layer block encoder for further operation.

Meanwhile, PHY symbol interleaver is a random interleaver that conducted after symbol mapping. As shown in Fig. 5, it spreads the modulated symbols from header bits to all OFDM symbols uniformly. Since channel estimation is based on symbol-level instead of bit-level, our PHY symbol interleaver distributes the pilot symbols within the entire packet. We simulate the hard pilot extraction using SIGCOMM'08 trace [18]. As illustrated in Fig. 6, the red points denote the hard pilots. After interleaving, they are perfectly spread across the packet, verifying that they can provide great support for CSI calibration.

C. CSI Calibration

CSI calibration utilizes the hard and soft pilots to reduce the residual channel effect, and thus improves the accuracy of channel estimation. Specifically, the estimation obtained from the above pilots is used to calibrate the initial estimation from the preamble. The calibration process is as follows. Assume that the extracted pilots have a position set $\boldsymbol{S}_k$ in the kth OFDM symbol. A pilot at the pth subcarrier has transmitted value $x_k(p)$ and received value with channel fading and noise expressed as:

$$y_k(p) = h(p)x_k(p) + n_k(p). \tag{9}$$

With the assist of pilots, the LS channel estimation for the k^{th} OFDM symbol is obtained as:

$$\hat{h}_k(p) = \frac{y_k(p)}{x_k(p)} = h(p) + n_k(p), p \in \boldsymbol{S}_k, k = 1, ..., N. \tag{10}$$

The channel estimation is calibrated by weighting the previous and current estimation, which is expressed as:

$$\tilde{h}_k(p) = \begin{cases} \alpha_k(p)\tilde{h}_{k-1}(p) + (1-\alpha_k(p))\hat{h}_k(p), & \text{for } p \in \boldsymbol{S}_k, \\ \tilde{h}_{k-1}(p), & \text{otherwise.} \end{cases} \tag{11}$$

where $\alpha_k(p)$ is the corresponding weight, and $\tilde{h}_0(p) = \hat{h}_0(p)$. Therefore, the residual channel effect after calibration can be expressed as:

$$\text{mse}_k(p) = E\{||\tilde{h}_k(p) - h(p)||^2\}, \tag{12}$$

and we aims to choose the optimal weight by minimizing the residual channel effect for each subcarrier. The procedure is then expressed as:

$$\alpha_{k,opt}(p) = \underset{\alpha_k(p)}{\text{argmin}}\ \text{mse}_k(p). \tag{13}$$

To derive the optimal weight, we assume that there are S_p^k pilots at the p^{th} subcarrier in k OFDM symbols, and $\mathbf{P}$ contains the index of OFDM symbols with S_p^k pilots. Without loss of generality, the minimum mse can be obtained when assuming the channel estimation has the same accuracy with the same noise variance $\sigma_n^2/2$. In this way, Equation (10) can be expressed as:

$$\tilde{h}_k(p) = \begin{cases} \beta_k(p)\hat{h}_0(p) + \frac{1-\beta_k(p)}{S_p^k}\sum_{n=1}^{S_p^k}\hat{h}_{\mathbf{P}_k}(p), & \text{for } S_p^k > 0 \\ \hat{h}_0(p), & \text{for } S_p^k = 0. \end{cases}$$

where $\beta_k(p)$ is the combined weight, and is expressed as:

$$\begin{aligned} \beta_{k,opt}(p) &= \underset{\beta_k(p)}{\text{argmin}}\ \text{mse}_k(p) \\ &= \underset{\beta_k(p)}{\text{argmin}}\ \beta_k(p)^2\sigma_{n'}^2 + S_p^k\left(\frac{1-\beta_k(p)}{S_p^k}\right)^2\sigma_n^2 \\ &= \underset{\beta_k(p)}{\text{argmin}}\ \sigma_n^2 \cdot \\ &\left(\left(\frac{1}{2} + \frac{1}{S_p^k}\right)\beta_k(p)^2 - \frac{2}{S_p^k}\beta_k(p) + \frac{1}{S_p^k}\right). \end{aligned} \tag{14}$$

Equation (14) infers that the optimal weight can be easily obtained by finding the minimum point of the quadratic function, which is expressed as:

$$\beta_{k,opt}(p) = \frac{2}{S_p^k + 2}. \tag{15}$$

Therefore, we have:

$$\alpha_{k,opt}(p) = 1 - \frac{1-\beta_{k,opt}(p)}{S_p^k} = \frac{S_p^k + 1}{S_p^k + 2}. \tag{16}$$

With the optimal weight $\beta_{k,opt}(p)$, the residual channel effect can be improved by:

$$\text{mse}_k(p) = \frac{S_p^k + 1}{S_p^k(S_p^k + 2)}\sigma_n^2 < \sigma_{n'}^2. \tag{17}$$

As the number of pilots increases, CSI becomes more reliable.

Furthermore, to simplify the implementation, an iterative calibrated method is derived. For the $(k-1)$th OFDM symbols, the optimal weighting is easily conducted using (16) as:

$$\alpha_{k-1,opt}(p) = \frac{S_p^{k-1}+1}{S_p^{k-1}+2}. \tag{18}$$

If there is a pilot at the pth subcarrier, e.g., $S_p^k = S_p^{k-1}+1$, then we have:

$$\begin{aligned}\alpha_{k,opt}(p) &= \frac{S_p^k+1}{S_p^k+2} = \frac{S_p^{k-1}+2}{S_p^{k-1}+3} \\ &= \frac{1}{2-\alpha_{k-1,opt}(p)}.\end{aligned} \tag{19}$$

Therefore, combing (11) and (19), the iterative channel estimation can be obtained. In this way, we can reduce the residual channel effect and approach the real channel status.

D. Greedy Rate Selection

Greedy Rate Selection (GRS) is an essential component in SmartPilot, because it directly links to the throughput. Previous rate selection algorithms barely consider multipath effect, where some subcarriers experience deep fading, and have rather poor channel quality. When choosing the optimal data rate for transmission, these subcarriers will drag the selected rate to a relatively low level, and ruin the opportunities for those "good" subcarriers to have higher rate for transmission. Adaptive modulation and coding could harness frequency diversity to improve the throughput, yet it complicate the system. We only consideration the system with one modulation and coding scheme. The goal is to choose the highest *averaging data rate* R_{opt} for all the subcarriers. To achieve this goal, we first define that for subcarrier i, its *affordable data rate* r_i is the highest data rate at which a subcarrier can successfully decode. According to [19], r_i can be easily obtained trough $CCSI_i$ based on the off-line training SNR-bitrate relationship. Therefore, the *averaging data rate* R_i when transmitting with r_i can be expressed as:

$$R_i = \frac{n_i \times r_i}{N}, \tag{20}$$

where N is the total number of subcarriers, and n_i is the number of subcarriers that have *affordable data rate* greater or equal to r_i.

Our goal is to find out the optimal data rate R_{opt} by maximizing R_i, which is modeled as:

$$\begin{aligned}R_{opt} &= \max_{\{i=1,\ldots,N\}} R_i \\ &= \max_{\{i=1,\ldots,N\}} \frac{n_i \times r_i}{N}.\end{aligned} \tag{21}$$

A brute force method is to compute all the possible R_i and select the largest one. However, it brings in high complexity in time and space domain. To simply the design, we adopt a top-down searching approach. We first classify the subcarriers with the same *affordable data rate* ri into one groups n_i, and sort all the groups $\mathbf{n} = [n_1, n_2, \cdots]$ with $\mathbf{r} = [r_1, r_2, \cdots]$ in descending order. Then we start from the group with highest rate r_i, and compute $R_1 = {}^{n_1 \times r_1}/_N$, $R_2 = {}^{n_2 \times r_2}/_N$, etc.. We compute all the possible R_i and find out the maximum value. This value will be set to R_{opt}, and will be fed back from the receiver to the sender through acknowledgement.

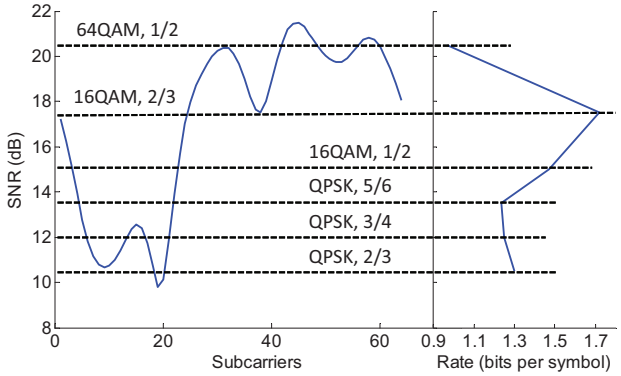


Fig. 7: Illustrated example of greedy rate selection. The left part denotes the subcarrier quality, and the right part is the averaging data rate under different modulation and coding schemes (MCS). Greedy rate selection always finds the optimal data rate and outperforms the existing rate selection algorithms.

In a frequency selective fading channel, e.g., some subcarriers experience deep fading, greedy rate selection strategically leaves them out of consideration, and chooses a much higher rate than they can afford. One possible question is how we deal with these deep fading subcarriers. To guarantee their transmission, we assign them the lowest data rate R_{bsc} (e.g., BPSK, 1/2, as preamble). Note that this is simple to achieve. After R_{opt} computation, the receiver will record the subcarriers with $R_i < R_{opt}$, and store their index into a table. This table is fed back along with the rate selection choice R_{opt} to the sender [7]. Upon receiving the feedback, the sender first inquiries the table and modulates the subcarriers with R_{bsc}. And then the rest of the subcarriers are transmitted with R_{opt}. Unlike adaptive modulation schemes that have different data rate at each subcarrier [19], we only differentiate the deep fading subcarriers, which greatly simplifies the system.

Fig. 7 presents the basic idea of greedy rate selection. The left part denotes the channel quality for 64 subcarriers, while the right part is the averaging data rate under different modulation and coding schemes (MCS). The effective SNR is 11.2dB, thus ESNR chooses QPSK, 2/3 code rate as its transmission data rate, which is 1.33 bits per symbol. On the contrary, Our GRS chooses the *highest averaging data rate*, which is 16QAM, 2/3 code rate, resulting in a data rate of 1.72 bits per symbol. Furthermore, the subcarriers with *affordable data rate* lower than 16QAM, 2/3 is set to BPSK, 1/2 code rate. Thus the overall data rate is further increased.

V. Performance Evaluation

In this section, we evaluate the performance of SmartPilot through extensive experiments and simulations. Our 802.11a/g like PHY layer is built on top of OFDM modules on GNU

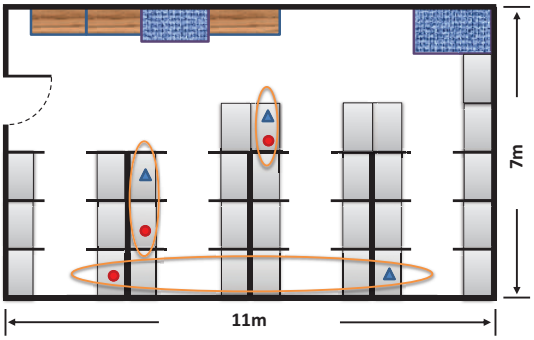


Fig. 8: Experimental environment. 3 sets of nodes are distributed in different locations.)

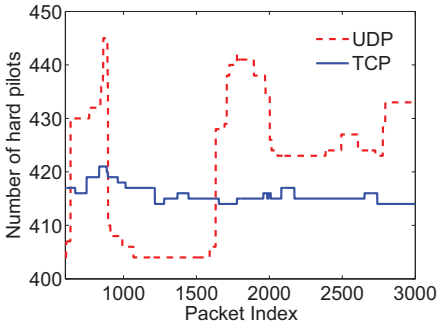


Fig. 9: The number of extracted hard pilot in the first 60 bytes from 3000 packets under different traffic.

radio platform [9]. The universal software radio peripheral 2 (USRP2) uses the RFX2450 daughterboard as RF frontend, which operates in the 2.4-2.5GHZ range. As depicted in Fig. 8, different sets of USRP2 nodes were tested to verify the experiment. In the following evaluations, the combinations of channel coding and modulation schemes in our implementation follows the standards in 802.11n [20]. LDPC encoder and sum-product decoder are adopted as the basic coding scheme. We use a 64-point FFT on a 20Mhz channel.

Our idea of utilizing soft and hard pilots to calibrate CSI is implemented on the USRP2 nodes by replaying real-life packet traces in our office. Yet due to the latency constraint, greed rate selection can be only realized through trace-driven simulations. Specifically, the SIGCOMM'08 trace [18] was replayed and collected to conduct greedy rate selection on our interconnected simulator with C++ and Matlab. SampleRate [3], the de facto rate selection algorithm used today, SoftRate [5] and ESNR [7], two research algorithms with the best published results, are selected as comparisons. Their simulations are also based on the traces we have collected in our office. Finally, we compare SmartPilot with 802.11 standard in terms of optimal throughput under AWGN channel and the trace-driven link.

A. Evaluation Methodology

Real Channel Status: To evaluate to what extent SmartPilot can approach the real channel status, we conduct experiments to train the link using known PN sequence. Two static usrp2 nodes are consistently transmitting 50 PN sequence. Then we compute the CSI based on the received PN sequences as the ground truth information.

SampleRate: SampleRate uses implicit feedback, e.g., packet reception ratio for rate adaptation. It computes the expected airtime to send a error-free packet (e.g., BER=0) using the delivery statistics for all the rate choices. When the current chosen rate has a airtime that exceeds the airtime of a lower rate, it falls back to the lower rate for adaptation. To decide whether or not to switch to a higher rate, every 10 packets the sender will transmit one packet to probe 1 or 2 higher rate. The main weakness of SampleRate is the low response to rapid channel changes due to frame-level rate adaptation metric.

SoftRate: SoftRate is an explicit feedback scheme that utilize estimated BER for rate selection. The receiver predicts the per-packet BER from convolutional decoder, and uses it to select the data rate according to the relationship between BER and modulation and coding schemes. The rate choice is then feedback to the sender through ACK. The main drawback of SoftRate is that, when BER = 0, it is difficult to determine to which level we should increase the data rate. Also, it cannot reflect the frequency diversity due to multipath effect.

ESNR: Our ESNR implementation follows the configurations indicated in [7]. The receiver measures CSI from OFDM preambles, and computes the highest rate configuration that has error-free packet deliveries (e.g. BER=0). It returns the rate choice to the sender through lightweight ACK. If the effective SNR has not changed, but consecutive packets are retried, the data rate will fall back to a lower level to protect the poor choices near rate boundary. The drawback of ESNR is the inaccurate measurement of CSI from OFDM preamble, which directly influences the rate selection. Also, when some subcarriers experience deep fading, they will damage the entire packet transmission.

SmartPilot: We implement SmartPilot as described in Sec. III. The sender takes the MPDU from MAC layer as input to rearrange the upper layer headers. The receiver keeps receiving the network packets, and extracts the soft and hard pilots during and after LDPC decoding. Then it performs CSI calibration and feeds CCSI to the greedy rate selection algorithm to compute the optimal data rate that ensures error-free packet deliveries (e.g., BER=0). The receiver also returns the rate changes through ACK as in [7].

B. Pilot Extractor

To begin with, we evaluate the performance of pilot extractor on our GNU Radio platform. The number of available pilots directly links to the performance of rate estimation. Thus, it is an essential component in the entire system. Pilot extractor consists of two components: hard pilot extraction and soft pilot extraction. We replayed the SIGCOMM'08 trace

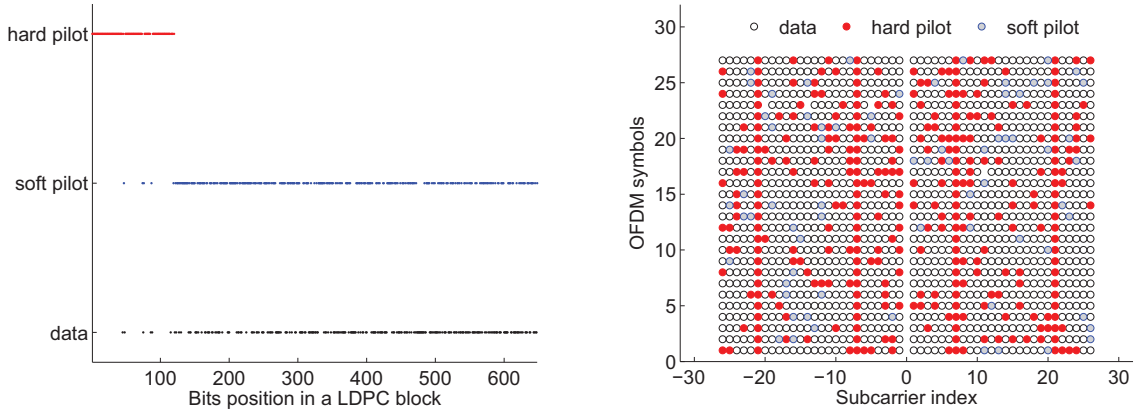


Fig. 10: The distribution of the extracted soft pilot and hard pilot. (a) Distribution in a 802.11 LDPC block (b) Distribution in all OFDM symbols within a packet.

in our office, and collected the UDP and TCP traffic for evaluation.

The hard pilots are extracted from the first 60 bytes of packets, which comprise of protocol headers up to transport layer. Fig. 9 shows the number of hard pilots in the first 60 bytes in 3000 packets. The history size is set to 600, with the misprediction threshold of 0.005. The hard pilot remains a constantly high quantity throughout the entire transmission. Specifically, the UDP traffic has an average number of 420-bit hard pilots, which is approximate 87% of the header bits (480bits). In the meanwhile, the TCP traffic has bit less number of hard pilots. That is because TCP header has more control fields that changed on packet level. However, TCP's hard pilot still yields about 410-bit and 85% of the entire header bits. Therefore, hard pilots from protocol headers are great help for CSI calibration.

Fig. 10 illustrates the positions of all pilots and data in one packet. The packet length is 324bytes. With QPSK modulation and rate-2/3 LDPC code, 27 OFDM symbols are generated in this packet. As shown in Fig. 10(a), with the help of MAC interleaver before LDPC encoding, hard pilots are distributed in all blocks. Meanwhile, soft pilots also spread across the entire packet. Fig. 10(b) gives the position of pilots in all OFDM symbols. They are uniformly distributed across OFDM subcarriers.

C. Residual Channel Effect

After obtaining the smart pilots, we see how they work on CSI calibration. In order to set a baseline, we trained the link between two USRP nodes by transmitting known 50 long PN sequences. We use the training sequences to compute the CSI as the ground truth information. Then we conduct normal data transmission to extract smart pilots. We follow the 802.11n standard to use least-square estimation (LS) channel estimation to compute the original CSI, and apply hard pilot and soft pilot to calibrate CSI.

Fig. 11 plots the residual channel effect of CSI and CCSI. CCSI achieves at least 7dB improvement over CSI across the SNR range from 0dB up to 35dB. Furthermore, with higher SNRs, CSI suffers from more critical noise enhancement. That is because noise at the deep fading channel conditions has more influence on channel estimation. On the contrary, calibrated channel estimation is not affected by such influence. It verifies that with the assistance of smart pilots, CCSI is more reliable for rate adaptation.

Fig. 12 presents a comparison among the channel response of ground truth, CSI and CCSI. We use BPSK modulation, and the SNR is around 10dB. Compared with CSI, CCSI is more approaching to the ground truth. These results verify that SmartPilot can effectively reduce the residual channel effect, and approach the real channel statues.

D. Greedy Rate Selection

Due to the latency constraint of USRP2, we are not allowed to conduct the real time evaluation for rate adaptation protocols. Therefore, we conduct trace-driven simulations to evaluate the performance of smart pilot. We replayed the SIGCOMM'08 trace in our office, and collected all the data traffic for simulation. SampleRate SoftRate and ESNR are chosen as three comparisons.

SmartPilot Rate Adaptation. In this step, we compare the performance of rate adaptation among SmartPilot, SampleRate, SoftRate and ESNR over the traces we gathered through USRP. As shown in Fig.14 and Fig.13, the chosen rate is the one computed through rate adaptation metric, where is the frame loss rate in SampleRate, estimated BER in SoftRate, effective SNR in ESNR. As for SmartPilot, two metrics are evaluated, ground truth information and calibrated CSI. The achieved rate is the average data rate for all the packets, which is computed from the transmitted error-free packets. Whenever the $BER > 0$, we set the achieved rate to 0.

The evaluation is divided into two steps. We first verify

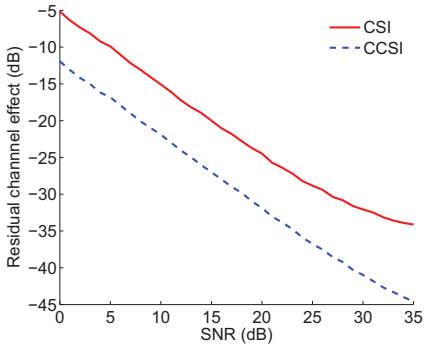


Fig. 11: The residual channel effect between CSI - ground truth and CCSI - ground truth.

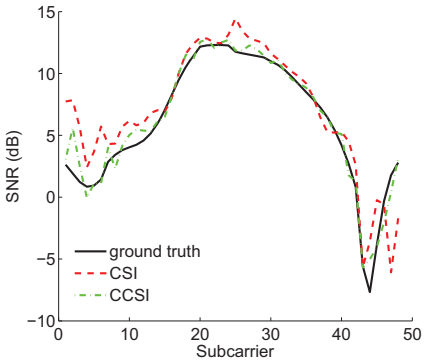


Fig. 12: The subcarrier SNR comparison of ground truth, CSI and CCSI. BPSK modulation, SNR=10dB.

the performance of greedy rate selection with ground truth channel information. As shown in Fig. 14(a), with the real channel status, greedy rate selection can choose relatively high data rate in a frequency selective fading channel. The achieved data rate perfectly traces the chosen data rate in practice, and archives a rate of over 4 bits per symbol, which verifies the effectiveness. Then we replace ground truth information with our calibrated CSI. The performance still remains promising. As shown in Fig. 14(b), with the assistance of hard pilot and soft pilot, the greedy rate selection also quickly tracks the chosen data rate, and archives a average data rate of over 3.6 bits per symbol.

In the second step, we compare the performance of SmartPilot with SampleRate, SoftRate and ESNR. As shown in Fig. 13, the chosen rates are not satisfactory in all the above three protocols. Among them, ESNR outperforms the other two protocols, which traces the chosen rate pretty well in fast time varying channel with a rate of 2 bits per symbol. Yet it still cannot catch up with SmartPilot since it cannot remove the residual channel effect. As for SampleRate and SoftRate, they exhibit very irresponsive reaction, and only achieves a rate of 1.9 and 1.72 bits per symbol due to deep fading subcarriers. We dig into the detail of these protocols and find out that, the reason for their bad behavior results from the incorrect channel estimation. They are likely to generate the estimated error and thus move on to an incorrect rate.

Optimal throughput. Finally, we evaluate the performance of SmartPilot in terms of optimal throughput compared with Shannon rate and the state-of-the-art 802.11 standard.

Shannon rate provides the upper bound that a system can achieve in AWGN channel. We use it as a baseline to see to what extend we can approach it. Shannon rate is calculated as follows [21]:

$$\mathrm{C} = \log_2(1 + \frac{P}{N_0}), \tag{22}$$

where P is the average received signal power over the bandwidth and N_0 is the average noise or interference power over the bandwidth. They are measured in watts.

The optimal throughput is measured by the envelope of all available rates, and is expressed by the product of modulation order and code rate. Note that the real throughput can be easily obtained by only weighting it with a fixed utilized bandwidth. To ensue the fair comparison, 802.11 standard adopts optimal rate selection algorithm, which means that it knows the true highest rate at any given time.

We first evaluate the performance of SmartPilot over AWGN channel, where white noise follows Gaussian distribution, and the energy spreads across the entire channel. Ideally, there is no fading or frequency selectivity. Fig. 15 demonstrates the evaluation results. Specifically, in Fig. 15(a) we gauge the achieved data rate of SmartPilot and 802.11 standard compared with Shannon rate. It is seen that SmartPilot has a better performance than the 802.11 standard when taking advantages of pilots. The achieved data rate is linearly increased as a function of SNR. When exceeding a certain threshold, i.e., 5 bits/symbol for 64QAM with 5/6 code rate, it cannot utilize higher SNR to achieve higher data rate, since there is no such combination of channel coding and modulation schemes. SmartPilot obtains higher rate than 802.11 standard. It outperforms 802.11 by 56% (-5dB to 5dB), 25% (5dB to 15dB) and 5% (above 15dB) achieved data rate, as shown in Fig. 15(b). These gains are benefited from hard pilot and soft pilot in LDPC decoder since there is no channel effect over AWGN channel. Fig. 15(a) also reveals the individual gain of hard pilot and soft pilot. Generally hard pilot contributes more than soft pilot. Yet these two kinds of pilots together have an synergy effect on the overall SmartPilot performance.

Fig. 16 gives a performance comparison between SmartPilot and 802.11 standard over a real link. The link is gathered from our office using two USRP2 nodes.We transmit 10min packets for every rate. Furthermore, SNR is controlled by the parameters of *tx_amplititude* and *tx_gain* in GNU Radio.. At the receiver, we calculate the achieved data rate of each available modulation and code type, and choose the envelope as the optimal throughput. To simplify the comparison, we still use Shannon rate in AWGN channel to show the upper bound. Fig. 16(a) shows SmartPilot greatly outperforms 802.11. It can reach error-free transmission even at lower SNRs, e.g., 3dB for BPSK modulation and 1/2 code rate. At the lower SNRs, i.e.,

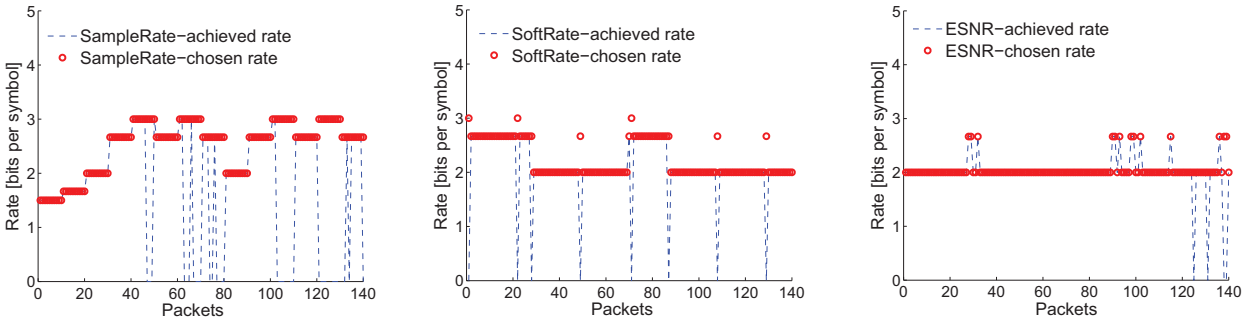


Fig. 13: Comparisons of rate selection algorithms under a time-varying channel. (a) SampleRate (b) SoftRate (c) ESNR. The chosen rates are not satisfactory in all the above three protocols.

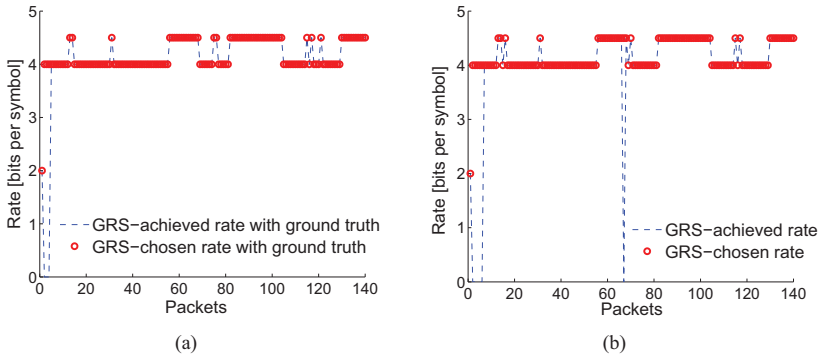


Fig. 14: Greedy rate selection (GRS). (a) over ground truth information (b) over calibrated CSI. With the assistance of hard pilot and soft pilot, the greedy rate selection can quickly tracks the chosen data rate, and achieves a promising performance as good as it knows the real channel status.

less than 5dB, SmartPilot can achieve data rate 39.5 times than 802.11 standard. That is because the rate is unavailable for 802.11 at the low SNRs, while SmartPilot can successfully recover the message with the aid of pilots. Furthermore, as shown in Fig.16(b), compared wit 802.11 standard, SmartPilot achieves $1.7\times$, $1.3\times$ and $1.06\times$ performance gains at the range of 5 to 15dB, 15 to 25dB and above 25dB.

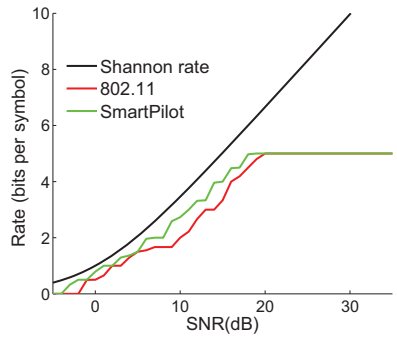
VI. Conclusion

In this paper, we propose a novel rate adaptation protocol termed SmartPilot. It aims at exploiting the potential data bits in PHY layer decoder and upper layer protocol header for rate adaptation. We first identify the problem in the existing rate adaptation protocols, that is, the lack of sufficient information to reduce residual channel effect. By investigating soft and hard pilots for channel estimation, SmartPilot can calibrate the channel state information (CSI) without extra overhead. With calibrated CSI, a Greedy Rate Selection algorithm is proposed, which leverages frequency diversity to obtain the optimal data rate. We have verified the efficiency of SmartPilot using GNU/USRP2 platform. The experiment results show that the residual channel effect has been reduced 87% using SmartPilot. We also simulate greedy rate selection algorithm based on CCSI using interconnected Matlab and C++. Extensive results show that SmartPilot achieves $2.1\times$, $1.9\times$ and $1.8\times$ throughput gain over SampleRate SoftRate and ESNR.

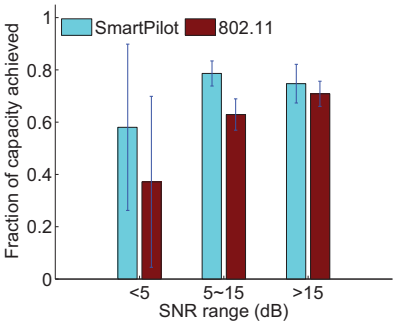
The design of SmartPilot provides a new panel to obtain reliable information. In next stage, we propose to exploit smart pilots under more communication scenarios [22] [23] [24] [25], such as Multiple Input Multiple Output (MIMO) and mobile environment.

Acknowledgment

This research is supported in part by HKUST Research Grants Council (RGC) 613113, grants from 973 project 2013CB329006, China NSFC under Grant 61173156, RGC under the contracts CERG 622613 and HKUST6/CRF/12R, grant from Huawei-HKUST joint lab, Program for New Century Excellent Talents in University (NCET-13-0908), Guangdong Natural Science Funds for Distinguished Young Scholar (No.S20120011468), New Star of Pearl River on Science and Technology of Guangzhou (No.2012J2200081), China NSFC Grant 61202454.

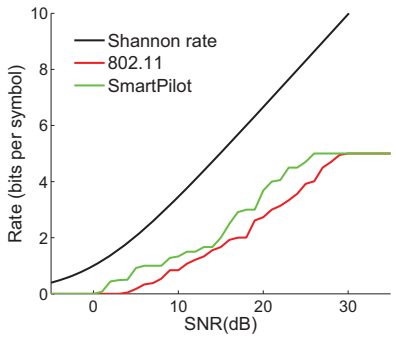


(a)

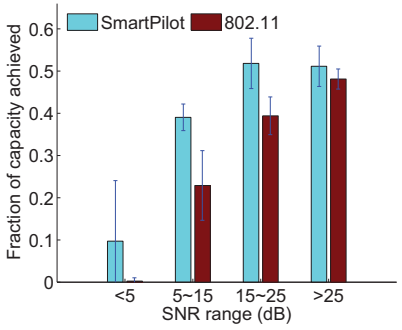


(b)

Fig. 15: Date rate achieved by SmartPilot and 802.11 standard over AWGN channel. (a) Achievable data rate (b) Achievable fraction of capacity.



(a)



(b)

Fig. 16: Data rate achieved by SmartPilot and the standard 802.11 over a frequency selective fading channel. (a) Achievable data rate (b) Achievable fraction of capacity.

REFERENCES

[1] I. . W. Group *et al.*, *IEEE 802.11n-2009: Enhancements for Higher Throughput*, 2009.

[2] C. Wong, R. Cheng, K. Lataief, and R. Murch, "Multiuser ofdm with adaptive subcarrier, bit, and power allocation," *Selected Areas in Communications, IEEE Journal on*, vol. 17, no. 10, 1999.

[3] J. C. Bicket, "Bit-rate selection in wireless networks," Ph.D. dissertation, Massachusetts Institute of Technology, 2005.

[4] S. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM Mobicom*, 2006.

[5] M. Vutukuru, H. Balakrishnan, and K. Jamieson, "Cross-layer wireless bit rate adaptation," in *ACM SIGCOMM*, 2009.

[6] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic media access for multirate ad hoc networks," in *Proceedings of the 8th annual international conference on Mobile computing and networking*. ACM, 2002, pp. 24–35.

[7] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4. ACM, 2010, pp. 159–170.

[8] J. Huang, Y. Wang, and G. Xing, "Lead: Lleveraging protocol signatures for improving wireless link performance," 2013.

[9] E. Blossom, "Gnu radio: tools for exploring the radio frequency spectrum," *Linux journal*, vol. 2004, no. 122, p. 4, 2004.

[10] M. Luby, "Lt codes," in *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*. IEEE, 2002.

[11] A. Shokrollahi, "Raptor codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 6, pp. 2551–2567, 2006.

[12] A. Gudipati and S. Katti, "Strider: Automatic rate adaptation and collision handling," *ACM SIGCOMM*, 2011.

[13] J. Perry, H. Balakrishnan, and D. Shah, "Rateless spinal codes," in *ACM HotNets*, 2011.

[14] J. G. Proakis, *Digital communications*, 5th ed. New York: McGraw-Hill, 2007.

[15] G. D. Forney Jr, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[16] J. Hagenauer and P. A. Hoeher, "A viterbi algorithm with soft-decision outputs and its applications," in *Global Telecommunications Conference, 1989, and Exhibition. Communications Technology for the 1990s and Beyond. GLOBECOM'89., IEEE*. IEEE, 1989, pp. 1680–1686.

[17] Y. Tong, L. Chen, and B. Ding, "Discovering threshold-based frequent closed itemsets over probabilistic data," in *Data Engineering (ICDE), 2012 IEEE 28th International Conference on*. IEEE, 2012, pp. 270–281.

[18] A. Schulman, D. Levin, and N. Spring, "Crawdad data set umd/sigcomm2008 (v. 2009-03-02)," 2009.

[19] H. Rahul, F. Edalat, D. Katabi, and C. Sodini, "Frequency-aware rate adaptation and mac protocols," in *ACM MobiCom*, 2009.

[20] *IEEE 802.11-2012: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, 2012.

[21] M. Fogiel, *The Handbook of Electrical Engineering*. Research & Education Assoc., 1996.

[22] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, "Mimo rate adaptation in 802.11 n wireless networks," in *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 2010, pp. 257–268.

[23] Y. Zhu, Z. Li, H. Zhu, M. Li, and Q. Zhang, "A compressive sensing approach to urban traffic estimation with probe vehicles," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 11, pp. 2289–2302, 2013.

[24] K. Wu, H. Li, L. Wang, Y. Yi, Y. Liu, D. Chen, X. Luo, Q. Zhang, and L. M. Ni, "hjam: Attachment transmission in wlans," *Mobile Computing, IEEE Transactions on*, vol. 12, no. 12, pp. 2334–2345, 2013.

[25] Y. Zheng and M. Li, "Pet: Probabilistic estimating tree for large-scale rfid estimation," *Mobile Computing, IEEE Transactions on*, vol. 11, no. 11, pp. 1763–1774, 2012.