RainbowRate: MIMO Rate Adaptation in 802.11n WiLD Links

Zenghua Zhao[1], Fucheng Zhang[2], Shaoping Guo[1], Xiang-Yang Li[3], Junze Han[3]
[1]School of Computer Science and Technology, Tianjin University, China
[2]School of Computer Software, Tianjin University, China
[3]Department of Computer Science, Illinois Institute of Technology, USA

***Abstract*—Legacy 802.11b/g protocols have been successfully applied in wireless long-distance (WiLD) networks to provide low-cost network connectivity for rural areas and developing countries, by designing TDMA MAC protocols. In contrast, we investigate the performance, and enhancement of using 802.11n for WiLD networks still using CSMA/CA. Our extensive experiments from our testbed show that, although 802.11n with CSMA/CA can provide network connectivity in WiLD environment, it suffers from poor performance. We thus propose RainbowRate, an efficient rate adaptation mechanism specified for WiLD links. RainbowRate leverages unique features found in WiLD links. We implement RainbowRate in off-the-shelf platforms and evaluate various aspects of our design using a testbed consisting of 6 mesh routers and 7 long-distance links, with the longest link length 20 km. Our experiment results show that RainbowRate out-performs state-of-the-art RA algorithms (MiRA, Ath9k and Minstrel) significantly in terms of throughput.**

I. Introduction

IEEE 802.11-based wireless long-distance (WiLD) mesh networks are emerging as a low-cost connectivity solution and are increasingly being deployed in rural regions with low population density [1]. These networks, together with large scale sensor networks [2], [3] enable many innovative applications. Such networks are featured with long-distance point-to-point links and high-gain directional antennae. The link length is usually over tens of kilometers and each link has direct line-of-sight. As in previous work [1], we assume that there is no significant external interference in WiLD networks, as such networks are usually deployed in remote rural areas with low population density and low wireless penetration. Many challenges exist to improve their performances, and coexist with other devices [4]–[7].

The legacy 802.11b/g has been applied successfully in such networks. However, the MAC protocol has been modified to TDMA type primarily due to inter-link interference [1], [8], [9]. To meet the increasing bandwidth requirements, it is natural to ask whether we can use higher data rate 802.11n for WiLD networks. In this work, we address several technical challenges of using 802.11n in WiLD networks. To the best of our knowledge, this is the first work to investigate 802.11n in WiLD networks. In contrast to the prior work, we still keep CSMA/CA to access the channel rather than TDMA-type MAC protocols for the following reasons.

Firstly, CSMA is inherently simple to implement and adaptive to dynamic traffic due to its random access. In contrast, TDMA-type MAC protocols suffer from the large overhead in centralized global synchronization, and low link utilization under a fixed time slot allocation when the traffic is asymmetric and fluctuates dramatically over time [9].

Secondly, inter-link interference can be avoided while using 802.11n. In majority WiLD networks, the number of links adjacent to a node is typically at most 7 (due to the limitation of directional antennae) with average of about 3 [1]. In prior work, the adjacent links operate at the same channel in WiLD networks, which leads to the inter-link interference. It is primarily because that 802.11b/g has only 3 orthogonal channels, and thus insufficient to be assigned to the adjacent links. However, 802.11n has 12 orthogonal channels when operating at 5 GHz, which is enough to be allocated to adjacent links such that each adjacent link has different orthogonal channels. Therefore, inter-link interference can be removed [10].

Our technical focus is on improving the performance of 802.11n WiLD networks based on CSMA/CA. During our design and implementation, we have encountered and conquered following challenges. Firstly, CSMA/CA is severely inefficient in long-distance links, due to the large value of $bandwidth \times delay$. The inefficiency makes retransmission very costly. CSMA/CA uses stop-and-wait ARQ mechanism, where the next packet would not be transmitted until the current packet has been delivered or dropped after certain number of retries [11]. The packet is retransmitted after without receiving an ACK for a period of $ack\_timeout$. The value $ack\_timeout$ should be modified to a large value in WiLD networks. Take a 20 km link for example, the $ack\_timeout$ increases to $171\mu s$ and $slot\_time = 65\mu s$, which are much longer than that in WLAN (where $slot\_time = 9\mu s$ for 802.11n). Even if the packet is received successfully, the next packet is transmitted at least after one RTT (Round-Trip Time) when the ACK is fed back. The RTT is much longer in WiLD networks due to long propagation delay. All the above makes long delay for transmitted packets. Along with high bit rate of 802.11n, CSMA/CA experiences low link utilization. Table I summarizes the theoretical maximum link utilization for several other scenarios. Notice that retransmission will

The research is partially supported by NSFC (National Natural Science Foundation of China) under Grant No. 61172063. The research of Li is partially supported by NSF CNS-1035894, NSF ECCS-1247944, NSF CMMI 1436786, and NSFC under Grant No. 61170216, No. 61228202.
978-1-4799-7575-4/14/$31.00 ©2014 IEEE

further reduce the link utilization.

The second challenge is that the existing RA (Rate Adaptation) algorithms are ill-suited in long-distance links, which degrades the link performance. RA is critical for link performance in multi-rate networks such as 802.11. We examine the well-known MiRA [12] and other two RAs (Ath9k and Minstrel) popular in the wireless card driver in our testbed. Our experimental results show that the maximum throughput with these three RA algorithms is worse than the maximum throughput achieved using the best fixed data rate (by searching all possible pairs of fixed FA and MCS). For example, when the RSSI is 21dB, the maximum throughput among these three RA algorithms is only 68.3 Mbps for 2×2 MIMO 802.11n; while the maximum throughput is 90 Mbps (achieved with MCS 11 and FA=32) at the similar channel states (*i.e.*, RSSI is 21 dB). A key reason here is that, these algorithms do not properly consider long RTT in WiLD links, where FA and retransmission have severe impact on the link utilization.

We have implemented and evaluated various aspects of our design in realistic 802.11n WiLD testbed. Our testbed consists of seven 802.11n long-distance links with length from 2.5 km to 20 km. Our experimental results confirm the aforementioned analysis and findings. We further observe that for the FDR-RSSI (Frame Delivery Ratio - Received Signal Strength Indicator) relationship, there exists a small transition window. Inside the transition window, the FDR varies from $10\% \sim 90\%$ quickly, whereas outside the window, the FDR is predictable. The transition window holds due to two unique characteristics in long-distance environment, (a) little multi-path fading effect, and (b) no significant external WiFi interference, which are different from WLAN [13]. The significance of the transition window lies in that it makes it possible to choose MCS according to the FDR-RSSI relationship.

In this paper, we propose RainbowRate, a novel RA mechanism to make 802.11n perform well in WiLD networks with CSMA rather than TDMA. RainbowRate leverages the transition window unique in long-distance environment to cut down the set of candidate MCSs efficiently. It adjusts bit rate in two levels, *coarse-grained level* and *fine-grained level*. In coarse-grained level, it selects candidate MCSs according to the RSSI, which makes it react to channel variation quickly. In fine-grained level, it adjusts the bit rate and FA level by probing MCSs only in MCS candidates set and its FA level, which protects the probing frames from high loss rate since the FDR of the MCSs in the set is over 90%. Fast switch and reliable-bit-rate retransmission further ensure low loss rate and few retransmission, thus improves the link utilization. We also use small CW (Contention Window) size and mFA (a pipelining ARQ with more frame aggregation level) to further improve the link utilization. RainbowRate is implemented as a shim layer in the driver protocol stack, without any modifications to hardware, so it can be employed on the off-the-shelf 802.11n wireless cards to retain its low cost.

We implement and evaluate RainbowRate in our testbed. Experiment results show RainbowRate outperforms MiRA, Ath9k and Minstrel in all the scenarios we tested. It improves UDP throughput by 107%, 159% and 34% respectively compared to MiRA, Ath9k and Minstrel, and improves TCP throughput by 31%, 153% and 38.5% respectively.

II. Related Work

802.11n measurements: Like legacy 802.11a/b/g, 802.11n is designed for short-range WLAN. The performance of 802.11n in WLAN has been studied well in theory and by measurements [14]. Sheth *et al.* [15] and Chebrolu *et al.* [16] studied 802.11b/g in long-distance wireless networks. In [17], the link length is no more than 1.8 km. To the best of our knowledge, our work is the first to study 802.11n in long-distance links in a systematic way via measurements.

RA algorithms: RA has been well studied for legacy 802.11b/g [18]–[20], and for 802.11n [12], [21]–[24]. RA algorithms proposed for 802.11b/g do not perform well in 802.11n networks since they are not designed for MIMO systems [12]. The major challenge to design RA for 802.11n is too many bit rates available to be selected. There are 32 MCS indices, and for each MCS index there are 4 different bit rates corresponding to SGI/LGI, 20 MHz/40 MHz.

MiRA [12] is the state-of-the-art RA for 802.11n, however its rate sorting schemes make it difficult to be extended to select candidate rates adaptively. RAMAS [21] separates rate adaptation into two groups and each group has its own rules for upgrading and downgrading indices. MiRA and RAMAS adopt information fedback from the receiver, whereas SGRA [19] adapts bit rate guided by SNR retrieved from PHY layer. However, SGRA is designed for legacy 802.11b/g networks with significant interference, therefore it focuses on how to calibrate FDR-SNR relationship under interference.

Ath9k and Minstrel HT (abbreviated as Minstrel throughout the paper) are RAs provided by the driver. Ath9K sorted all the MCS indices according to their bit rate without differentiating their stream mode. Therefore it is always stuck in a single stream mode or a double stream mode as illustrated in [12]. Minstrel uses random probing to select the optimal bit rate, and sometimes is stuck in sub-optimal bit rate. ERA [22] focuses on energy efficiency, which is beyond the scope of this paper.

None of the above are appropriate in long-distance environment, since they do not address the challenges unique in such networks, such as low link utilization.

III. Performance of 802.11n in WiLD Links

In this section, we first describe the testbed we have deployed, then present experiment observations to show the performance of 802.11n in WiLD links.

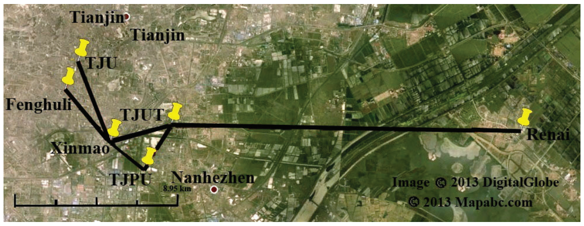
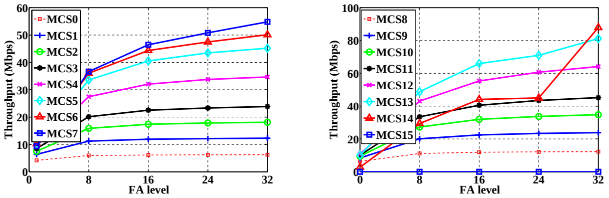


Fig. 1: The network topology of our testbed



(a) MCS 0-7 at 20 MHz

(b) MCS 8-15 at 20 MHz

Fig. 2: The impacts of FA on the throughput for MCS 0-15

A. *Experimental setup*

We conduct all the experiments in our campus testbed as shown in Fig. 1. Our testbed comprises of 7 long-distance links for our study from 2.5 km to 20 km in length. A long-distance link in our experiment consists of two 802.11n radios communicating with each other between two sites. Each node is deployed using water-proof boxes atop various buildings. All the links are line-of-sight.

We use a programmable platform, UBNT RouterStation Pro running Open-WRT open source OS, as our hardware platform. Each of them is equipped with a UBNT SR71-A outdoor 802.11n 3×3 MIMO mini-PCI wireless card. The directional antenna is UBNT RocketDish5G-30 dish antenna with high gain of 30 dBi, beam width of 5°.

We run all the experiments at 5 GHz frequency band to simulate the scenarios in rural areas without significant external WiFi interference.

B. *Performance of 802.11n in WiLD links*

After increasing $ack\_timeout$ and $slot\_time$ to avoid ACK timeout, we want to know the performance of 802.11n in long distance links. We therefore conducted extensive experiments. The results show that 802.11n still experiences poor performance. We try to find the fundamental reasons that impact its performance in this section.

FA impact on the throughput: In our experiments at the 20 km link, the sender sends UDP traffic to the receiver at all possible combinations of parameter values for (MCS, FA, bandwidth), where MCS is from 0 to 15, FA level is from 1 to 32 in steps of 2 (but only show the results of step 8 for sake of clarity), and bandwidth is 20 MHz/LGI and 40 MHz/SGI.

Fig. 2 shows average throughput for each MCSs at 20 MHz. The results are similar at 40 MHz. We can see that when FA level is 1, the average throughput among all MCSs is no more than 10 Mbps. For lower MCSs (MCS 0-1), the throughput is near saturated after FA level 8 or 16, however for higher MCSs, the throughout increases with the increment of FA level

In a nutshell, FA has significant impact on the throughput of 802.11n at WiLD links.

Analysis from link utilization perspective: Why FA impact on 802.11n in long-distance links so severely? To answer the question, we analyze its link utilization. CSMA/CA uses DCF (Distributed coordination function), which adopts stop-and-wait ARQ [11].

We calculate the maximum link utilization over different FA level for 20 km and 50 km links regardless of retransmission and backoff time, as listed in Table I. We only consider 40 MHz/SGI for limited space. The payload length is 1400 bytes for a frame. SIFS is $16\mu s$ as in [11], DIFS depends on link length, $146\mu s$ at 20 km link and $278\mu s$ at 50 km link. From the table, we can see that for MCS 15 40 MHz/SGI, the link utilization of 20 km and 50 km links is 77% and 62% respectively. The larger the FA level, the higher the utilization.

TABLE I: FA impact on 802.11n maximum link utilization at 20 km links regardless of retransmission and backoff time

	20 km	20 km	20 km	50 km	50 km	50 km
MCS	FA 1	FA 16	FA 32	FA 1	FA 16	FA 32
0	0.52	0.90	0.93	0.42	0.88	0.91
2	0.36	0.86	0.91	0.27	0.82	0.87
4	0.22	0.79	0.88	0.16	0.72	0.81
7	0.16	0.73	0.85	0.11	0.64	0.75
8	0.36	0.86	0.91	0.27	0.82	0.87
10	0.22	0.79	0.88	0.16	0.72	0.81
12	0.13	0.67	0.82	0.08	0.58	0.7
15	0.09	0.59	0.77	0.06	0.48	0.62

Retransmission is costly: However, retransmission time in long-distance links cannot be ignored due to long propagation delay. Firstly, if the number of retransmission is n, the total time to transmit a frame increases by n times, so the utilization of the link decreases to $1/(n+1)$. Take MCS 12 at 40 MHz, 20 km link for example, if the retransmission is 3, the link utilization deceases to $20\%(= 1/4 \times 0.80)$. In other words, the corresponding throughput decreases from 180 Mbps to 36 Mbps. Secondly, when we consider the backoff time, the utilization is even lower. Since for each retransmission, the CW size will double. The minimum and maximum CW size for best effort traffic is (15, 1023) in 802.11n [11]. The maximum backoff time is then $1023 \times slot\_time = 1023 \times 65\mu s = 66ms$, where $slot\_time$ increases to $65\mu s$ instead of $9\mu s$ to eliminate protocol collisions. That is to say, retransmission will probably enlarge backoff time and thus enlarge total transmission time.

C. *RA algorithms: a case study*

802.11n supports various MCSs, and each MCS corresponds to a different physical layer bit rate. For such multi-rate wireless networks, link RA is critical to improve link performance. Does existing RA algorithms help to improve 802.11n link performance?

To answer this question, we choose three popular RA algorithms, including the state-of-the-art MiRA [12], and other two RA algorithms provided by the driver, Minstrel and Ath9k. We implemented MiRA prototype in our testbed, and conducted experiments to examine the performance of the above RA algorithms. The experiments run at our 20 km link and at 40 MHz. The results are shown in Table II. To our surprise, none of the above RA algorithms can achieve a good performance. The average UDP throughput of MiRA, Minstrel and Ath9k is 60.0 Mbps, 68.3 Mbps and 36.7 Mbps respectively. We also run fixed rate algorithm, the average throughput is 90 Mbps for MCS 11 with no loss, whereas MCS 12 gives a throughput of only 62.4 Mbps with loss rate of 48%. The RSSI varies very slowly with the average of 21 dB during these experiments. In this case, 90 Mbps is the optimal throughput for RA algorithms to reach since RA algorithms have to adapt to the optimal MCS, resulting in more or less overheads. However even the highest throughout obtained by Minstrel is only about 75% of the optimal throughput.

TABLE II: Results of existing RA algorithms

RA Algorithms	Primary MCSs	Other MCSs	Throughput (Mbps)	Loss rate	Average FA level
Minstrel	MCS 11 (99.4%)	0.6%	68.3	2.3%	16.5
MiRA	MCS 11 (95.9%)	4.1%	60.0	12.4%	31.4
Ath9k	MCS 11 (70.9%) MCS 12 (28.6%)	0.6%	36.7	27.6%	11.9

We analyzed the traces and the algorithms. We found that fundamental reason for poor performance is that these algorithms leverage probing to estimate the current link quality, which will result in much overhead due to MCSs probed. Probing at higher bit rate results in higher loss rate, and that at lower bit rate leads to lower throughput directly. High loss rate causes heavy retransmission, which degrades the throughput severely as analyzed in Section III-B. Ath9k randomly selects MCS candidates to probe and then sorts the rates providing the best throughput considering packet error rate. Because of the random selection, rates are stuck in sub-optimal rates in some cases. In this case, it is stuck at MCS 12 with high loss rate, which makes it suffer from high loss rate of 27.6%. Minstrel also randomly chooses MCS candidates to probe. It uses more MCSs but much less probing frames compared to Ath9K, which is why Minstrel outperforms Ath9k at most time. However, the average length of A-MPDUs of Minstrel is only 16.5, which impacts on its throughput. MiRA is based on zigzagging and probing between intro-mode and inter-mode. MiRA should have better behavior than Minstrel and Ath9k, since it does not probe randomly. However in this case, it shows MiRA uses more probing frames which is 4.1%. In addition, its loss rate is as high as 12.4%.

Implications: The probing mechanisms used in RA algorithms make them inefficient in terms of throughput in long distance networks, moreover costly retransmission and small FA level are key challenges too.

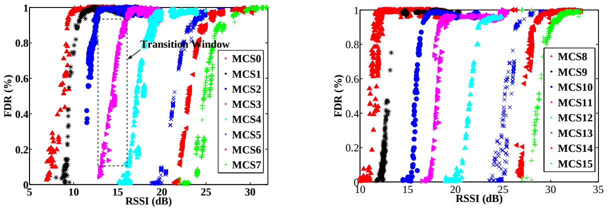
IV. RSSI-PDR Relationship in WiLD Links

Probing is a common method to select the best bit rate in RA algorithms. However, probing causes more overhead in WiLD links. Can we predict the link quality and select bit rate with little or even no probing? We hence conducted experiments to address this problem. We consider the following parameters: FDR (Frame Delivery Ratio) and RSSI.

A. FDR vs. RSSI: Transition Window

We carried out extensive experiments in our testbed to measure FDR and RSSI relationship at different MCSs. It is worth to note that FDR and RSSI are collected at the MAC layer. We turned off MAC-layer ACKs and set the maximum retries limit to zero. This allows us to measure the real frame loss rate in absence of any MAC-layer acknowledgements and retries. In each run, the sender transmits saturated UDP traffic in a fixed MCS and transmit power for 120 seconds. MCS varies from 0 to 15, and the bandwidth is across 20/40 MHz.

Fig. 3 shows the plot of FDR in different MCSs versus RSSI at our 20 km link, MCS 0-15 at 20 MHz. The results is similar at 40 MHz. It is clear that different MCSs have different RSSI thresholds. Generally, higher rate requires higher RSSI to sustain. The same rate at 40 MHz needs higher RSSI than that at 20 MHz. We observe a characteristic transition region for FDR versus RSSI. In transition window, the FDR rises from 10% (lossy) to 90% (reliable). For example, Table III gives the width of the transition widow for MCS 0-7. The others are similiar. We see that most link transitions happen within a window of around 2 dB for MCS 0-4 with BPSK or QPSK modulation. For MCS 5, 6 and 7 with QAM modulation, the transition window spreads wider to 3-4 dB. The observation of transition window is consistent with the link abstraction found in [25] and the results in scenarios without interference in [19]. This is to say, channel quality is predictable outside of the transition window.



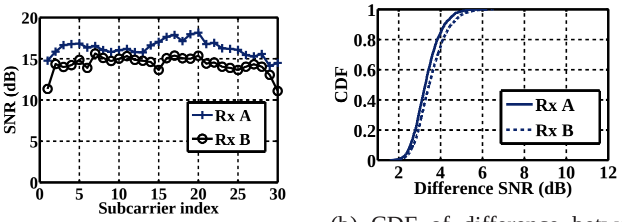
(a) MCS0-7 at 20MHz

(b) MCS8-15 at 20MHz

Fig. 3: FDR vs. RSSI

TABLE III: Transition window width for MCS 0-7 at 20 MHz

MCS	Rate (Mbps)	Transition) window width (dB)
0	6.5	2
1	13.0	2
2	19.5	2
3	26.0	2
4	39.0	3
5	52.0	3
6	58.5	3
7	65.0	4



(a) CSI of a frame at one link

(b) CDF of difference between min and max per subcarrier SNR

Fig. 4: CSI analysis

We further evaluate the impact of locations and environments on FDR-RSSI relationship. We calculate the transition window for traces from all the links in our testbed. We see the transition windows exist in all traces, and for one link, the transition window is stable with time. It suggests that RSSI threshold exists regardless of locations and environments.

B. Why RSSI is a good link quality indicator in WiLD links?

It is well-known that RSSI cannot be used to predict link quality in WLAN [13]. The possible reasons for the existence of narrow-width transition window in 802.11n WiLD links could be two folds as follows.

Firstly, the long-distance link is line-of-sight, and long of course, which makes little multipath fading effect [26]. In this case, each subcarrier of OFDM leveraged by 802.11n PHY layer experiences almost the same channel, therefore RSSI calculated from all the subcarriers can be a good indicator for the channel.

We further verify this by collecting CSI measurements of the link using CSI tools [27]. Fig. 4a shows the CSI of a MPDU across subcarriers at MCS 0. We can see that the distribution is quit flat. For each received MPDU, we compute the difference between minimum and maximum SNR across different subcarriers. The mean difference value for one stream is about 4 with standard deviation of 0.6. We further calculate the cumulative distribution (CDF) of the difference for MCS 0 as shown in Fig. 4b. It says that the variation of each subcarrier is very small, 80% difference of SNR across all the subcarriers is less than 4 dB. That is to say, the multipath fading effect is not significant in WiLD environment.

Secondly, in contrast to WLAN, there is no significant external WiFi interference in rural areas [25]. In the experiments, we run experiments at 5 GHz to avoid external WiFi interference, simulating the case in rural areas. We scanned WiFi sources at 5 GHz band, only found one source which works at 161 channel, whereas our nodes work at 44 channel. There is no interference at all.

Implications: The experimental results indicate that the FDR-RSSI shows narrow transition window for each MCS and the relationship exists in all the links tested, which motivates us to develop an efficient RA algorithm guided by RSSI to improve 802.11n link performance.

V. RainbowRate Design

A. Algorithm design

In this section we propose RainbowRate to improve 802.11n throughput in long-distance links, and to provide high bandwidth network connectivity for remote rural areas. The RSSI-based bit rate adaptation consists of two modules to adjust MCS and its FA level to the link quality: coarse-grained adaptation and fine-grained adaptation. In coarse-grained adaptation, it selects the MCS candidates set quickly according to RSSI ranges. In fine-grained adaptation it adjusts MCSs in the set and its FA level by probing to select the best MCS and FA level. The former module makes it find available MCSs efficiently so as to reduce excessive probings, while the latter leads it to reach the best bit rate accurately. In contrast to the existing RA algorithms, the probing used in RainbowRate has very little overhead. In addition, we propose small CW size and mFA to improve the link utilization and further improve the throughput.

We now are ready to delve into the details of RainbowRate.

1) Coarse-grained adaptation: We select MCS candidates set corresponding to the RSSI estimated on line. This operation is triggered if the estimated RSSI changes to another RSSI range. We now describe MCS candidates set and RSSI estimate.

MCS candidates set: Based on the relationship of FDR-RSSI, there is a transition window for each MCS. Outside the transition window, the FDR of the MCS is predictable, *i.e.*, when RSSI is larger than a threshold $rssi_a$, its FDR is over 90%. Therefore, given a RSSI range, the MCSs whose RSSI threshold falls in this RSSI range form MCS candidates set. In the MCS candidates set, the MCSs are sorted by their bit rates in descending order regardless of their stream modes (*i.e.*, single-stream mode or double-stream mode).

We also select a reliable MCS for each MCS candidates set, the reliable MCS is the MCS with near 100% FDR in that RSSI range. The reliable MCS is used as the MCS for retransmission, which will be described in Section V-A2.

2) Fine-grained adaptation: In fine-grained adaptation, we use probing to adjust MCS and its FA level. In contrast to probings in existing RAs, we only probe MCSs in the MCS candidate sets, in which MCSs usually have a high FDR over 90%, therefore the probing MPDUs will not suffer from high loss rate. Moreover, we leverage fast switch and reliable-bit-rate retransmission to further ensure a low-level loss rate.

The probing is triggered every 2048 MPDUs. 2048 MPDUs are selected as the probing period instead of a fixed time period for two reasons: (a) it is easy to operate since it is powers of 2 and (b) it corresponds to a dynamic period in terms of time for different MCS. For example, it takes about 104 ms to transmit 2048 frames at MCS 13, whereas about 138 ms at MCS 12. Dynamic probing period is more flexible.

In each probing period, at most 4 groups of MPDUs are used as probing frames, each with 64 frames, since 64 frames probing is typically sufficient to collect loss statistics. Where group 1 is transmitted at the primary MCS with current FA level minus $step$, group 2 at the primary MCS with current FA level plus $step$, group 3 at adjacent MCS with higher bit rate if there is any, and group 4 at adjacent MCS with lower bit rate if there is. $step = 2$ based on our experiments, since this will not cause throughput fluctuation. When the MCS candidates set changes, the primary MCS is set to the one with the highest bit rate in the new set. In this way, there are at most $256 (= 4 \times 64)$ MPDUs used as probing among 2048 MPDUs, which is only 12.5%, therefore the overhead is little. Furthermore, there are 2 groups of probing MPDUs transmitted at primary MCS but with different FA level, they are not overhead actually since FA level with small difference will not cause much throughput degradation. In this sense, the overhead is only $6.25\% (= 2 \times 64/2048)$.

Fast switch: At the end of the probing period, the primary MCS will be set to the one with the highest estimated throughput and the FA level is set to the one used in that MCS correspondingly. However, if the FDR of the primary MCS is less than 80% and not the one with the highest throughput probed, then the primary MCS is switched to the one with the highest throughput immediatly without waiting until the end of the probing period. This is called fast switch. Fast switch ensures to adapt quickly to the bursty link quality variation, thus protects MPDUs against loss.

Reliable-bit-rate retransmission: As we pointed out that retransmission is very costly in long-distance links, we then design a reliable-bit-rate retransmission rule. In this rule, when a MPDU has been retransmitted for once and not been delivered yet, it means that the frame has been transmitted for twice and is lost at both times at this MCS, then the MCS is regarded as in risk of high loss rate. To ensure low loss rate, the MPDU is then retransmitted using the reliable MCS with almost 100% FDR in this MCS candidate set. The total number of retransmission of the MPDU is 5, where 1 retransmission is at its MCS, the others are at the reliable MCS. 5 retransmission is enough to ensure almost frames delivery since MCSs in the MCS candidates set have high FDR over 90% and the reliable MCS has FDR of almost 100%. Actually, in our experiments most MPDUs are delivered at most after 2 retransmissions. This rule is applied to all the MPDUs transmitted at primary MCS and the probing ones.

In summary, the RSSI-based rate selection leverages the MCS candidates set to adapt to the link quality quickly and uses fast switch and reliable-bit-rate retransmission to keep lower loss rate and less retries, thus achieves high throughput.

B. Other design options

CW size impact: From the analysis in Section III-B, we can see that backoff time contributes to the low link utilization, and the value of backoff time depends on the CW size. In DCF, when the medium transitions from busy to idle, multiple nodes may be ready to send data [11]. To minimize collisions, nodes wishing to initiate transfer select a random backoff count and defer for that number of slot times. The random backoff count is selected as a pseudo-random integer drawn from a uniform distribution over the interval $[0, CW]$, where CW, an integer value, is the contention window size.

The CW parameter takes the initial value CW_{min} and effectively doubles on each unsuccessful frame transmit. If the CW reaches CW_{max} it remains at that value until it is reset. The CW is reset to CW_{min} after every successful frame transmit. In 802.11n, (CW_{min}, CW_{max}) for the best effort traffic is $(15, 1023)$. Large CW is important to avoid collisions when there are many nodes sharing the channel. However it is very inefficient in long-distance links since backoff time becomes much longer than that in WLAN. This happens because $slot\_time$ is modified longer according to the link distance, and the backoff time is $k \times slot\_time$, where k is randomly selected from (CW_{min}, CW_{max}). However, large CW is not necessary in WiLD links at all, since there are only two nodes sharing a point-to-point link which is the most common link type in WiLD networks. Even for point-to-multipoint links, the number of nodes sharing a channel is no more than 3 to 4 [1]. We therefore want to use small CW size to reduce backoff time.

We consider (CW_{min}, CW_{max}) of $(7, 15)$ and $(15, 31)$, and calculate their collision probability after n retransmissions. In this case, CW is CW_{min} when a frame is transmitted for the first time, and CW is CW_{max} for all the retransmission. The collision probability after n retransmission is $\frac{1}{CW_{min}+1} \times \frac{1}{(CW_{max}+1)^n}$. The collision probability after 2 retransmissions is 0.05% and 0.006% for $(7, 15)$ and $(15, 31)$ respectively. To verify the performance of small CW, we carried out a simple experiment on our 20 km link. In this experiment, one node sends TCP/UDP traffic to the other node at MCS 12 with FA level 32. The results are listed in Table IV. We can see that the throughput of TCP and UDP at all the small CW sizes tested increase compared to that using the original CW size (15, 1023) at comparable link quality.

TABLE IV: Impact of CW size on fixed MCS

CW size	(1, 7)	(7, 15)	(15, 31)	(15, 1023)
TCP (Mbps)	54.7	49.4	48.2	39.1
UDP (Mbps)	184.00	160.85	155.11	122.3

mFA: more frame aggregation: 802.11n experiences poor performance in long-distance links due to long propagation delay. Frame aggregation can help to improve link utilization, but the maximum FA level is 32 which is insufficient for high MCSs to achieve high link utilization. For example, the link utilization of MCS 12 at 40 MHz, FA level 32, 20 km link is only 82% according to Table I. RA algorithms cannot improve the maximum link utilization although they can optimize the throughput. Therefore we design mFA, a pipelining ARQ to improve link utilization by using large FA level $m, m > 32$. It transmits consecutive m frames before waiting for a block acknowledgement. We calculated the maximum link utilization for MCSs at 40 MHz for 20 km link, and found that when $m = 256$, the maximum link utilization is over 98%. Since 802.11n does not support FA level larger than 32, we hence implement a shim layer above the MAC layer for mFA. The MAC layer ACK and retransmission are disabled.

VI. Implementation and Evaluation

We implemented RainbowRate in UBNT RouterStationPro, a programmable platform used in our testbed. We evaluate whether RainbowRate can improve link throughput. We choose MiRA [12], and other two RAs provided with 802.11n driver, Minstrel and Ath9k, as comparison. The experiments were conducted in the 20 km link in our testbed. The traffic is TCP and UDP. Since the channel changes slowly for long-distance links, we also change the transmit power to simulate the channel variation to see the reaction of the RAs. We first evaluate RainbowRate with default CW size and FA level for comparison, and then discuss the impact of CW size and mFA.

A. UDP traffic

We conduct experiments at various transmit powers to test UDP performance under different link qualities. We then collect the average throughput for each of the RAs. The results are shown in Fig. 5. RainbowRate outperforms other RAs in all the scenarios we tested. RainbowRate works well even at low link quality. For example, at average RSSI of 14 dB, RainbowRate improves the throughput by 134%, 107% and 27% over Ath9k, MiRA and Minstrel respectively. From Table V, we can see that the loss rate of RainbowRate is less than 1%, which is much lower than any of other RAs. The low loss rate makes RainbowRate delivery frames with little retransmission, therefore it achieves higher throughput.

Table VI lists the MCSs distribution of RAs at average RSSI of 26 dB. The average FA level of RainbowRate is

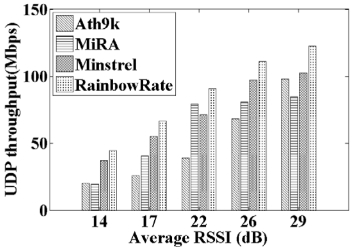


Fig. 5: UDP throughput over various link qualities

TABLE V: Loss rate of RAs

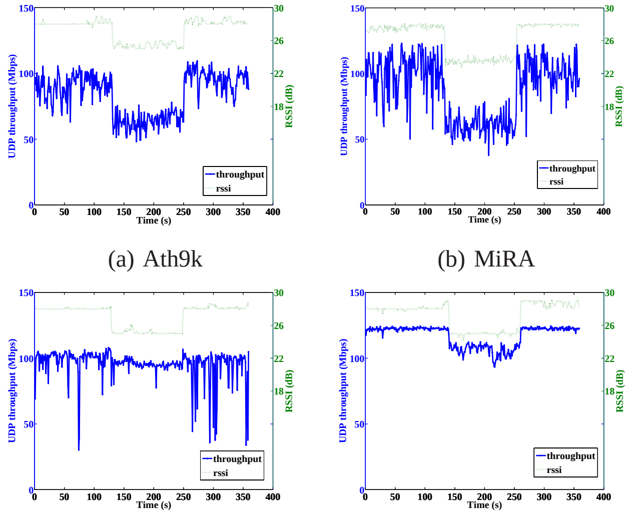
SNR (dB)	Ath9k	MiRA	Minstrel	RainbowRate
14	8.15%	55.58%	4.2%	0.21%
17	27.6%	12.4%	2.35%	0.33%
22	8.93%	1.57%	1.14%	0.16%
26	11.8%	2.36%	3.69%	0.71%
29	7.7%	28.5%	6.57%	0.12%

31.9, almost reaching the maximum value of 32. The primary MCS is MCS 12. MPDUs transmitted by MCS 12 are 99.96%. MiRA has 2 primary MCSs: MCS 12 and MCS 5. 50.5% and 15.7% MPDUs are sent at MCS 12 and MCS 5 respectively. MPDUs sent by other MCSs are 33.8%. The bit rate of MCS 5 is 120 Mbps, whereas the bit rate of MCS 12 is 180 Mbps. MCS 5 needs more RSSI to sustain its high throughput than MCS 12. Therefore MiRA suffers more loss rate than RainbowRate. Minstrel's loss rate is 3.69%, 99.3% MPDUs are sent at its primary MCS 12. It probes all the other MCSs due to its random probing. Although MPDUs sent at other MCSs is only 0.7%, they more or less degrades its performance. The average FA level of Minstrel is 20.1, which has impact on its throughput too. Ath9k uses MCS 12 and MCS 13 as its primary MCSs, 56.6% and 43.3% MPDUs are sent at them respectively. The use of MCS 13 results in its high loss rate up to 11.8%. High loss rate leads to many retransmissions. Therefore its throughput is only 68 Mbps, whereas the throughput of RainbowRate is 111 Mbps. Ath9k experiences low throughput among them, since it is stuck in sub-optimal bit rate MCS 13.

We change transmit power during experiment to check how quickly RAs react to the channel dynamics. We plot the throughput with time in Fig. 6. The results show that MiRA, Ath9k and RainbowRate can adjust their bit rate according to the link quality, whereas Minstrel does not show clear throughput transition due to random bit rate probing. Ath9k and MiRA suffer from throughput fluctuation, in contrast, RainbowRate is smooth during the experiment. RainbowRate can react to the channel quickly because it changes its bit rate candidates once the link quality varies. In addition, it only probes at a small set of bit rates, which reduces the probing overhead.

TABLE VI: MCSs distribution of RAs: UDP

RA Algorithms	Primary MCS (%)	Other MCSs	Average FA level
RainbowRate	MCS 12(99.96%)	0.04%	31.9
Minstrel	MCS 12 (99.3%)	0.7%	20.1
MiRA	MCS 12 (50.5%), MCS 5 (15.7%)	33.8%	30.3
Ath9k	MCS 12 (56.6%), MCS 13 (43.3%)	0.1%	14.4



(a) Ath9k

(b) MiRA

(c) Minstrel

(d) RainbowRate

Fig. 6: RA UDP throughput with dynamic transmit powers.

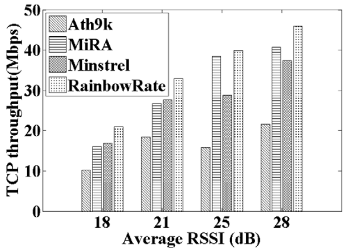


Fig. 7: TCP throughput of RAs over different link quality

B. *TCP traffic*

TABLE VII: MCSs distribution of RAs: TCP

RA Algrihms	Primary MCS(%)	Other MCSs	throughput	Average FA level
RainbowRate	MCS 12(94.4%) MCS 13 (3.8%) MCS 11 (1.7%)	0	43.8	10.6
Minstrel	MCS 12 (90.5%)	9.5%	35.2	7.4
MiRA	MCS 1 (51.9%) MCS 12 (31.2%)	33.8%	38.9	10.2
Ath9k	MCS 13 (79.2%) MCS 14 (11.0%)	9.8%	19.6	8.4

TCP results are shown in Fig. 7. RainbowRate outperforms other RAs in all the scenarios we tested. We analyzed traces of all the experiments, and take RSSI 28 for example since they are similar. The MCS distribution, throughput and average FA level are listed in Table VII. For RainbowRate, 94.4% MPDUs are sent at MCS 12, and 3.8% MPDUs are sent at MCS 13. MCS 11 is only reliable MCS to retransmit MPDUs, never used as probing. MiRA uses MCS 11 and MCS 12 as its primary MCSs, it also probes MCS 10 and MCS 14. The use of MCS 11 contributes to its lower throughput. Minstrel also uses MCS 12 as its primary MCSs, but it probes all the other MCSs randomly, and its average FA level is only 7.4 which degrades its throughput. For Ath9k, its throughput is the lowest among others. It uses MCS 13 as its primary MCS although the FDR at MCS 13 is only 70.2%. It even uses MCS 14 to send 11% MPDUs with FDR of 36.6%.

We also changed transmit power during experiment to check

TCP performance of RAs under bursty channel variation. The results are not shown here for limited space. For TCP traffic, RainbowRate keeps a higher throughput too compared to other RA algorithms, and reacts to the change more quikly. The fundamental reason is that RainbowRate leverages fast switch and reliable-bit-rate retransmission to ensure low loss rate, which improves the throughput of TCP.

C. *CW size experiments*

We set CW size (CW_{min}, CW_{max}) to different combinations of (1, 7), (7, 15) and (15, 31) to test the performance of RainbowRate. As shown in Table VIII, the throughput of UDP improves to 136.43 Mbps at (1,7), however, the UDP throughput at other two CW sizes are comparable to that at original (15, 1023). That is because that RainbowRate reduces the number of retransmission efficiently, most of the MPDUs can deliver in 3 retries, therefore the CW (15, 31) seems have no effect on it. For TCP, the throughput at three small CW sizes are all higher than that at original CW size.

TABLE VIII: CW size impact on RainbowRate

CW size	(1, 7)	(7, 15)	(15, 31)	(15, 1023)
TCP (Mbps)	61.84	61.29	59.17	45.95
UDP (Mbps)	136.43	127.83	116.52	122.62

D. *mFA experiments*

We conducted experiments to run RainbowRate under mFA level of 128 and 256, the average throughput is 148.63 Mbps and 160.68 Mbps respectively. While the throughput is 122 Mbps with mFA level 32. The throughput increases after using more frame aggregation.

VII. Conclusion

In this work, we systematically studied the 802.11n characteristics in WiLD links through extensive experiments on our testbed. We proposed RainbowRate, a novel RA algorithm specific for WiLD networks. It leverages FDR-RSSI relationships unique in WiLD networks to assist cutting down bit rate candidates, which makes it react to the channel variation quickly. It also uses probing among bit rate candidates and FA level in a fine-grained way to make it accurately. Our work is among the first to examine MIMO RA in long-distance links in a practical setting. In the future work, we will verify RainbowRate in multi-hop WiLD networks.

References

[1] B. Raman and K. Chebrolu, “Design and evaluation of a new mac protocol for long-distance 802.11 mesh networks,” in *MobiCom ’05*, pp. 156–169.

[2] X. Mao, X. Miao, Y. He, T. Zhu, J. Wang, W. Dong, X. Li, and Y. Liu, “Citysee: Urban co2 monitoring with sensors,” in *Proceedings of IEEE INFOCOM*, 2012.

[3] Y. Liu, Y. He, M. Li, J. Wang, K. Liu, and X. Li, “Does wireless sensor network scale? A measurement study on greenorbs,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 24, no. 10, pp. 1983–1993, 2013. [Online].

[4] Z. Zhao, X. Wu, J. Zhao, and X.-Y. Li, “Zigbee vs wifi: Understanding issues and measuring performances of ieee 802.11n and ieee 802.15.4 coexistence,” in *IEEE IPCCC 2014*

[5] Y. Yan, P. Yang, X.-Y. Li, Y. Tao, and L. You, “Zimo: Building cross-technology mimo to harmonize zigbee smog with wifi flash without intervention,” in *ACM Mobicom 2013.*

[6] P. Yang, Y. Yan, X.-Y. Li, L. You, J. Wang, J. Han, and Y. Xiong, “Wizbee: Wise zigbee coexistence via interference cancelation in single antenna,” *IEEE Transactions on Mobile Computing*, 2014.

[7] Y. Tao, X. Li, and C. Bo, “Performance of coexisted wifi and zigbee networks,” in *IEEE ICDCS 2013 Workshops*, pp. 315–320.

[8] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, “Wildnet: Design and implementation of high performance wifi based long distance networks,” in *ACM NSDI ’07*, pp. 87–100.

[9] S. Nedevschi, R. K. Patra, S. Surana, S. Ratnasamy, L. Subramanian, and E. Brewer, “An adaptive, high performance mac for long-distance multihop wireless networks,” in *ACM MobiCom ’08*, pp. 259–270.

[10] S. Avallone and I. F. Akyildiz, “A channel assignment algorithm for multi-radio wireless mesh networks,” *Comput. Commun.*, vol. 31, no. 7, pp. 1343–1353, May 2008.

[11] “IEEE standard for wireless local area networks: 802.11n.” [Online]. Available: http://www.ieee802.org/11

[12] I. Pefkianakis, Y. Hu, S. H. Wong, H. Yang, and S. Lu, “Mimo rate adaptation in 802.11n wireless networks,” in *ACM MobiCom*, 2010, pp. 257–268.

[13] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris, “Link-level measurements from an 802.11b mesh network,” in *SIGCOMM ’04*. New York, NY, USA: ACM, 2004, pp. 121–132.

[14] V. Shrivastava, S. Rayanchu, J. Yoonj, and S. Banerjee, “802.11n under the microscope,” in *ACM IMC*, 2008, pp. 105–110.

[15] A. Sheth, S. Nedevschi, R. Patra, S. Surana, E. Brewer, and L. Subramanian, “Packet loss characterization in wifi-based long distance networks,” in *IEEE INFOCOM ’07*, pp. 312 –320.

[16] K. Chebrolu, B. Raman, and S. Sayandeep, “Long-distance 802.11b links: performance measurements and experience,” in *ACM MobiCom*, 2006, pp. 74–85.

[17] U. Paul, R. Crepaldi, J. Lee, S.-J. Lee, and R. Etkin, “Characterizing wifi link performance in open outdoor networks,” in *IEEE SECON ’11*, pp. 251–259.

[18] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, “Robust rate adaptation for 802.11 wireless networks,” in *ACM MobiCom*, 2006, pp. 146–157.

[19] J. Zhang, K. Tan, J. Zhao, H. Wu, and Y. Zhang, “A practical snr-guided rate adaptation,” in *IEEE INFOCOM*, 2008, pp. 2083–2091.

[20] M. Vutukuru, H. Balakrishnan, and K. Jamieson, “Cross-layer wireless bit rate adaptation,” in *ACM SIGCOMM ’09*, 2009.

[21] D. Nguyen and J. J. Garcia-Luna-Aceves, “A practical approach to rate adaptation for multi-antenna systems,” in *IEEE ICNP*, pp. 331–340.

[22] C.-Y. Li, C. Peng, S. Lu, and X. Wang, “Energy-based rate adaptation for 802.11n,” in *ACM MobiCom ’12*, pp. 341–352.

[23] W. Kim, O. Khan, K. Truong, S.-H. Choi, R. Grant, H. Wright, K. Mandke, R. Daniels, R. Heath, and S. Nettles, “An experimental evaluation of rate adaptation for multi-antenna systems,” in *IEEE INFOCOM ’09*, pp. 2313–2321.

[24] L. Deek, E. Garcia-Villegas, E. Belding, S.-J. Lee, and K. Almeroth, “Joint rate and channel width adaptation for 802.11 mimo wireless networks,” in *IEEE SECON ’13*, 2013, pp. 167–175.

[25] B. Raman, K. Chebrolu, D. Gokhale, and S. Sen, “On the feasibility of the link abstraction in wireless mesh networks,” *Networking, IEEE/ACM Transactions on*, vol. 17, no. 2, pp. 528 –541, april 2009.

[26] D. Tse and P. Viswanath, *Fundamentals of wireless communication*. Cambridge University Press, 2005.

[27] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Predictable 802.11 packet delivery from wireless channel measurements,” in *ACM SIGCOMM ’10*, pp. 159–170.