SampleLite: A Hybrid Approach to 802.11n Link Adaptation

Lito Kriara
The University of Edinburgh
Edinburgh, United Kingdom
l.kriara@sms.ed.ac.uk

Mahesh K. Marina
The University of Edinburgh
Edinburgh, United Kingdom
mahesh@ed.ac.uk

ABSTRACT

We consider the link adaptation problem in 802.11n wireless LANs that involves adapting MIMO mode, channel bonding, modulation and coding scheme, and frame aggregation level with varying channel conditions. Through measurement-based analysis, we find that adapting all available 802.11n features results in higher goodput than adapting only a subset of features, thereby showing that holistic link adaptation is crucial to achieve best performance. We then design a novel hybrid link adaptation scheme termed SampleLite that adapts all 802.11n features while being efficient compared to sampling-based open-loop schemes and practical relative to closed loop schemes. SampleLite uses sender-side RSSI measurements to significantly lower the sampling overhead, by exploiting the monotonic relationship between best settings for each feature and the RSSI. Through analysis and experimentation in a testbed environment, we show that our proposed approach can reduce the sampling overhead by over 70% on average compared to the widely used Minstrel HT scheme. We also experimentally evaluate the goodput performance of SampleLite in a wide range of controlled and real-world interference scenarios. Our results show that SampleLite, while performing close to the ideal, delivers goodput that is 35–100% better than with existing schemes.

1. INTRODUCTION

802.11n [1] is the currently deployed version of the IEEE 802.11 wireless LAN standard offering application-level throughput (goodput) in excess of 100Mbps through a combination of new PHY and medium access control (MAC) layer enhancements that include the use of multiple antennas (or MIMO), channel bonding and frame aggregation. *Link adaptation* in the 802.11n context concerns adapting these various MAC and PHY features in response to varying channel and interference conditions. Link adaptation has a bigger impact on application performance in 802.11n WLANs than with legacy 802.11a/b/g WLANs – a suboptimal choice of feature settings can lead to a goodput loss in the order of 100s of Mbps as opposed to 10s of Mbps with 802.11a/g. However, as shown in [2], 802.11a/b/g link adaptation schemes turn out to be ineffective when applied in the 802.11n setting because the latter breaks some of the assumptions underlying legacy schemes. This observation has led to research on new link adaptation schemes specifically targeting 802.11n in the last few years. Most of these are open-loop schemes (e.g., [2, 3, 4]) that rely on some form of sampling at the sender side to identify the best performing settings for 802.11n features. As there are a large number of feature setting combinations to search across with 802.11n[1], existing schemes incur high sampling overhead or tend to use sub-optimal settings for extended periods. Closed-loop approaches (e.g., [5]) that measure channel quality on receiver side and feed it back to the sender, would be effective in theory, but they face practical implementation related hurdles.

In this paper, we propose a novel *hybrid* link adaptation approach termed SampleLite that leverages passive Received Signal Strength Indicator (RSSI) measurements on the sender side to identify a very small subset of feature setting combinations to sample for each link. SampleLite is a hybrid link adaptation scheme in the sense that it employs sampling (like open-loop schemes) and makes use of channel quality information (like closed-loop ones), but avoids their limitations. Unlike existing open-loop schemes, SampleLite dramatically reduces the search space to sample without risking the use of sub-optimal settings. And differently from typical closed-loop schemes, it relies on channel quality information (RSSI) already available at the sender side and therefore is easily implementable. Elaborating further, we make the following key contributions in this paper:

- To start with, we quantify the benefit of adapting multiple 802.11n features using testbed measurements that span a diverse set of scenarios differing in channel and interference conditions. We find that it is indeed crucial to adapt *all* features to obtain the best goodput.
- We then design SampleLite, a novel hybrid link adaptation scheme that adapts all available 802.11n features. It is driven by the insight that the maximum goodput yielding *setting* of each 802.11n PHY feature (MIMO mode, channel bonding and MCS) exhibits monotonicity with respect to RSSI (measured at sender side). We exploit this insight to limit the feature settings that need to be sampled. While the existing work that raises concerns about the utility of RSSI focuses on the relationship between performance (e.g., goodput) and RSSI, we focus on the relationship between *the setting of a feature* and the RSSI and that too in indoor environments. We implement SampleLite in the ath9k driver [6].
- We evaluate SampleLite in comparison with the state of the art solutions using an indoor 802.11n WLAN testbed across a wide range of scenarios (including experiments with real-world interference and mobility). Results demonstrate its superior performance. SampleLite delivers throughput close to ideal, and provides improvements of around 100% and up to 35% compared to RAMAS [4] and Minstrel HT, respectively. Moreover, it reduces sampling overhead by over 70% on average compared to Minstrel HT [3] (that does random exhaustive sampling).

2. RELATED WORK

Pefkianakis et al. [2] investigated the link adaptation problem in 802.11n WLANs and observed the non-monotonicity between

[1] 256 combinations — 4 spatial streams$\times$2 channel widths$\times$2 guard intervals$\times$8 MCSs$\times$2 frame aggregation levels (ON/OFF).

Link Type	RSSI	Line of Sight (LoS)?	Characteristic
A	[-33.5, -43.8] dBm	Yes	Strong
B	[-51.1, -60.2] dBm	No	Medium
C	[-64.8, -71.2] dBm	No	Medium–Weak
D	[-73.5, -81.1] dBm	No	Weak

Table 1: The range of average RSSI values (in dBm) for each link type in the testbed.

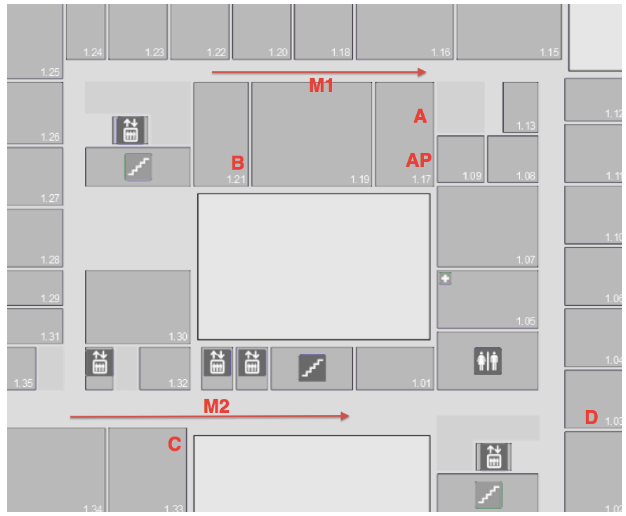


Figure 1: A snapshot of the 802.11n WLAN testbed.

frame error rate and bit-rate across different MIMO modes (number of spatial streams). Based on this observation, they proposed a scheme called MiRA that zig zags across different MIMO modes to search for the rate providing the maximum goodput. Minstrel HT [3] is the default link adaptation scheme in the commonly used open-source 802.11n wireless driver, ath9k [6]. It does random and exhaustive sampling of all 802.11n feature settings to update the expected throughput and loss rate of each feature setting combination. The combination that provides the highest expected throughput is chosen for data transmissions. However, if this selected rate turns out to be too lossy then it lowers the rate by reducing the number of streams. Differently from the schemes outlined above, RAMAS [4] takes a credit-based approach. It divides the 802.11n features into two groups: the modulation group with different MCS values and the enhancement group that consists of number of streams, channel widths and guard intervals. RAMAS uses credit-based algorithms to adapt these groups independently of each other and combines the results together to decide the overall feature setting. Combes et al. [7] explore the fundamental limits of sampling based rate adaptation algorithms and design a family of algorithms called ORS that learn the optimal settings as fast as possible (not necessarily the same as the optimizing the overhead due to sampling). Underlying ORS algorithms are certain unrealistic assumptions such as prior knowledge of the speed at which the environment is changing, which makes them more of theoretical interest.

While all the above are open-loop schemes relying on some form of sampling, the alternate closed-loop approach has also been investigated. Even though the availability of complete channel state information (CSI) would make the 802.11n link adaptation straightforward [8], sampling and feeding it back to the sender is expensive [9]. This has motivated the work in [10, 5]. The work in [5] additionally observes that CSI is not widely supported across all chipsets (which we confirm is still mostly true) and devises a more universally obtainable channel quality metric called diff-SNR. The authors in [5], however, report several practical hurdles for implementing the feedback (non-compliance of commodity chipsets with 802.11n standard and retrieving feedback embedded in ACK frames at sender side driver level) required for their ARAMIS scheme.

3. METHODOLOGY

Testbed. We take an experimental approach throughout the paper for analysis and evaluation. For this, we use an indoor 802.11n WLAN testbed deployed in our office building on a floor spanning an area of $30 \times 50m^2$. A snapshot of the testbed layout on the floor map is shown in Fig. 1. Each node in the testbed is a Linux based laptop equipped with a 2×3 802.11n mini PCI express card with an Atheros AR9300 chipset and running the ath9k driver. We however use the 802.11n card only in a 2×2 MIMO configuration throughout. Transmit power is fixed at the default setting of 18dBm, and this is the case for all MCSs. We consider nearly all available 802.11n features — frame aggregation (FA), MIMO mode (i.e., number of spatial streams), channel bonding (CB), and modulation and coding schemes (MCS). The only exception is the short guard interval (SGI), which is only supported for 40MHz channels in the chipsets we use. We disabled SGI in our experiments for consistency.

For traffic generation, we employ the commonly used Iperf tool for creating UDP traffic sessions between AP and client stations. Packet size is fixed at the Iperf default value of 1500 bytes. All our controlled experiments are conducted during late night hours to minimize noise. Each data point in the results reported in this paper is obtained from averaging across multiple runs.

Metrics. The key performance metric we focus on is the *Goodput* (application-level throughput on the receiver side), which is calculated as bits delivered per unit of time excluding the overheads related to protocol headers and retransmissions. We also measure sampling related probing overhead per unit of time for different link adaptation schemes.

Link Types and Interference Scenarios. We realize a diverse set of link qualities by varying the locations of client stations while keeping the access point (AP) position fixed; links from several different client locations are grouped into 4 types based on their RSSI values and other characteristics (e.g., line of sight) as indicated in Table 1. RSSI here (and henceforth) refers to the value of the variable by the same name reported by the ath9k driver at runtime. Using a spectrum analyzer, we found that channels 149 – 161 in the 5GHz band were free of any activity at all times, so we use those for our controlled experiments.

Interference scenarios we consider in our controlled experiments are similar to those in [11, 12]. Specifically, we consider five different interference scenarios: no interference (NI), co–channel interference (CCI), co–channel legacy interference (CCLI), adjacent channel interference (ACI) and adjacent channel legacy interference (ACLI). In all these scenarios, the interfering link consists of another AP-station pair next to the AP and station A in Fig. 1, and its link quality falls under the link type A (see Table 1). CCI and ACI scenarios are illustrated in Fig. 2(a) and Fig. 2(b), respectively. CCLI and ACLI scenarios are similar to CCI and ACI, respectively, except that interfering link is a legacy 802.11a link in the CCLI and ACLI scenarios.

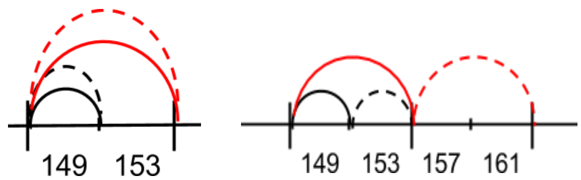


Figure 2: Illustration of co-channel (CCI) and adjacent channel (ACI) interference scenarios. Solid line represents the link under test while the dashed line corresponds to the interfering link. Black (red) coloured lines indicate the use of 20MHz (40MHz) channels.

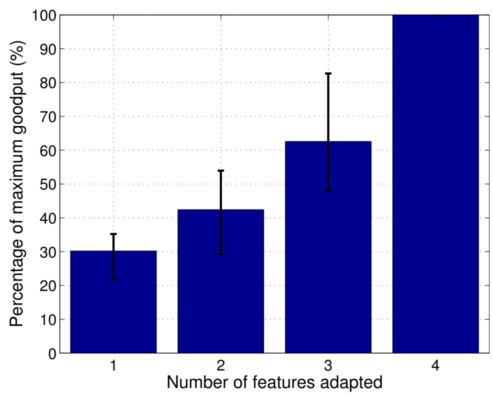


Figure 3: Percentage of maximum goodput obtained from adapting any 1, 2, 3 and all 4 features of FA, MIMO, CB and MCS. While the individual bars show the average gain in each case, the error bars indicate the minimum and maximum gain of each case.

Acronym	Definition
FA	Frame Aggregation
CB	Channel Bonding
MIMO	Multiple Input Multiple Output mode (STBC, SDM)
STBC	Space-Time Block Coding
SDM	Spatial Division Multiplexing
MCS	Modulation and Coding Scheme
GP	Goodput
NI	No Interference
CCI	Co-Channel Interference
CCLI	Co-Channel Legacy Interference
ACI	Adjacent Channel Interference
ACLI	Adjacent Channel Legacy Interference

Table 2: Acronyms used throughout the paper.

4. BENEFIT OF ADAPTING MULTIPLE 802.11N FEATURES

Several 802.11n link adaptation schemes focus on a subset of features (e.g., [2, 7, 11]). For example, MiRA [2] and ORS [7] do not consider channel bonding, whereas it is the sole focus of [11]. To quantify the importance of holistic link adaptation, we examine the benefit of adapting all features relative to cases when only a subset of features are considered. Towards this end, we obtain goodput measurements using our testbed for each possible feature setting combination in different link type and interference scenarios. Essentially, we get a set of tuples $[FA, CB, MIMO, MCS, GP]$ for each link type and interference scenario (see §3). The various acronyms used are summarized in Table 2. Here FA, CB, MIMO and MCS are the four different features we examine in this work, and GP is the goodput given by each specific combination experimentally. Moreover, using this extensive dataset, we identify the feature setting combination that yields the maximum goodput in each link and interference scenario. We use the notation $maxGP_{ij}$ to refer to the maximum goodput with link type i and interference scenario j.

We now assess the fraction of goodput obtained from adapting a subset of all available features. Consider the case when only one feature x is adapted out of the four examined in this study. For this case, we find the average goodput $x1_{ij}$ and $x0_{ij}$, when x is enabled and disabled, respectively, averaged across all possible settings of other features. Then we compute the fraction of goodput obtained from adapting only the feature x, $gain(x_{ij}) = max(x1_{ij}, x0_{ij})/maxGP_{ij}$. Similar computation is done for each of the other features. Then we average all these goodput fractions and across all link i and interference j scenarios. This is the result we report for adapting just one feature. Similarly, we repeat this process for when two, three or all four available features are simultaneously adapted. Note that the detailed impact, behaviour and interdependencies of each feature are described in detail in our previous work [12].

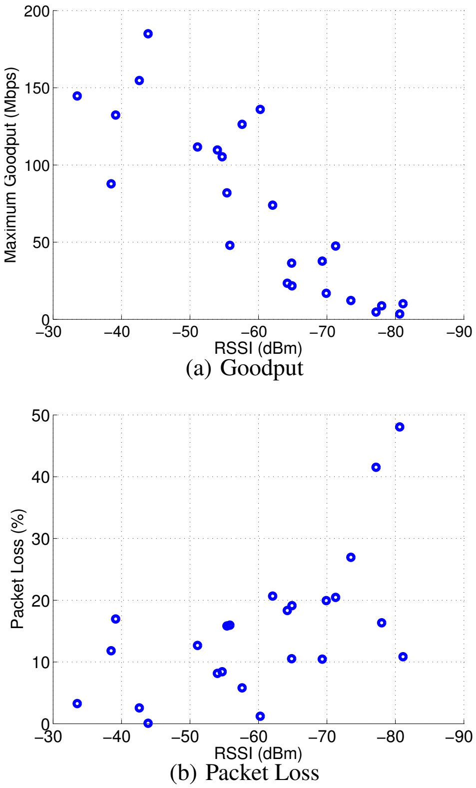
Results are shown in Fig. 3. Clearly, considering all 4 features gives 100% of the maximum goodput. We observe that considering any 3 features as opposed to all 4 yields only around 60% of the maximum goodput, on average. This drops to around 40% when any 2 features are adapted and to 30% when only any 1 of the features is adapted. These results demonstrate that it is vital to adapt all available features to achieve maximum goodput performance.

5. SAMPLELITE

We have just shown that adapting all 802.11n features is essential for maximizing goodput. From the earlier discussion of related work, we observe that sampling based open-loop approaches can be inefficient while the closed-loop approaches relying on direct measurement of channel quality face practical hurdles. In this section, we present a new hybrid solution called SampleLite that is holistic and has aspects of these two existing approaches but avoids their limitations.

5.1 Key Insight & Potential Overhead Savings

Our key insight is that the link quality as inferred by RSSI observations on the sender side can serve as a guide in significantly reducing the sampling space for open-loop schemes. While the RSSI as a channel quality indicator is simpler and easily accessible when used on the sender side, it is also shown to be an unreliable measure of packet delivery success [5, 8]. In fact, we observe this exact behavior in our evaluation, as depicted in our packet loss and goodput results in Fig. 4. Each data point in these results represents a different link and interference type scenario, and corresponds to the feature setting that gives the maximum goodput. As we can see, there is no noticeable correlation between RSSI and goodput or loss. However, for the same data, interestingly we find that setting of each feature in the maximum goodput yielding configuration shows a monotonic behavior with respect to RSSI, as shown in Fig. 5. This observation on monotonicity is valid across different hardware platforms even though the specific thresholds differ. For Figs. 5(a), 5(b), 5(c) we use hardware platform (with chipset AR9300) described in Section 3, whereas for Figs. 5(d), 5(e), 5(f) correspond to the use of the setup described in [12] and chipset



(a) Goodput

(b) Packet Loss

Figure 4: Maximum goodput and corresponding packet loss versus average RSSI of a link. Each data point represents a different link and interference type scenario, and corresponds to the feature setting that gives the maximum goodput.

AR9220. Note that majority of the links (about 80%) in our indoor testbed environment are non line of sight (NLoS) links, which could be a factor behind the usefulness of RSSI for inferring best feature settings.

We exploit this relationship between feature settings and RSSI in our approach as elaborated in the following. In the rest of the paper, we will focus on the hardware platform described in Section 3 (using AR9300 chipset) for simplicity of exposition. From Figs. 5(a), 5(b), 5(c), we can identify reasonably clear RSSI thresholds that separate the RSSI regions where each feature should take different values to yield the maximum goodput. This suggests that we could use the current RSSI as a guide in choosing a small subset of feature setting combinations to sample, thereby drastically reduce the sampling overhead in comparison with most existing schemes that resort to exhaustive sampling. Through simple analysis, we now estimate the potential savings in overhead from this idea, starting with the MIMO mode.

Fig. 5(a) suggests that when the average RSSI is more than a particular threshold (-79dBm) sampling should focus on MIMO mode set to two spatial streams and single stream otherwise. A similar observation can be made from Fig. 5(b) for sampling of channel bonding alternatives though for a different RSSI threshold. This indicates a possible sampling space reduction by half with respect to MIMO mode (when only up to 2 spatial streams are supported as is the case with our hardware) and further halving of sampling space in relation to channel bonding. Even greater savings in sampling overhead can be achieved with regard to MCS as its setting

	802.11n	**802.11ac**
MIMO mode (# spatial streams)	75%	87.5%
Channel Bonding	50%	75%
MCS	62.5%	70%
Total	95.3%	99%

Table 3: Potential sampling overhead reduction from exploiting the monotonic relationship between best feature setting and average RSSI for 802.11n and 802.11ac cases relative to exhaustive sampling based approaches.

can be chosen from more possibilities. We do notice some outliers in Fig. 5(c), possibly due to the known challenges with using RSSI as an indicator [8]. A simple and efficient way to handle such outliers would be to sample not just the specific MCS value suggested by the average RSSI measurement but also its neighboring ones. More generally, suppose that measured RSSI value maps to sampling the MCS value n, then for robustness, instead of just n, we could sample MCS values $\in [n-i, n, n+i]$ where i is a small number (e.g., 1, 2, 3). We experimentally determine that i = 1 (i.e., sampling 3 MCS values) works well as shown shortly. For 802.11n with 8 different MCS values, this suggests a reduction in sampling space by 62.5% with respect to MCS feature.

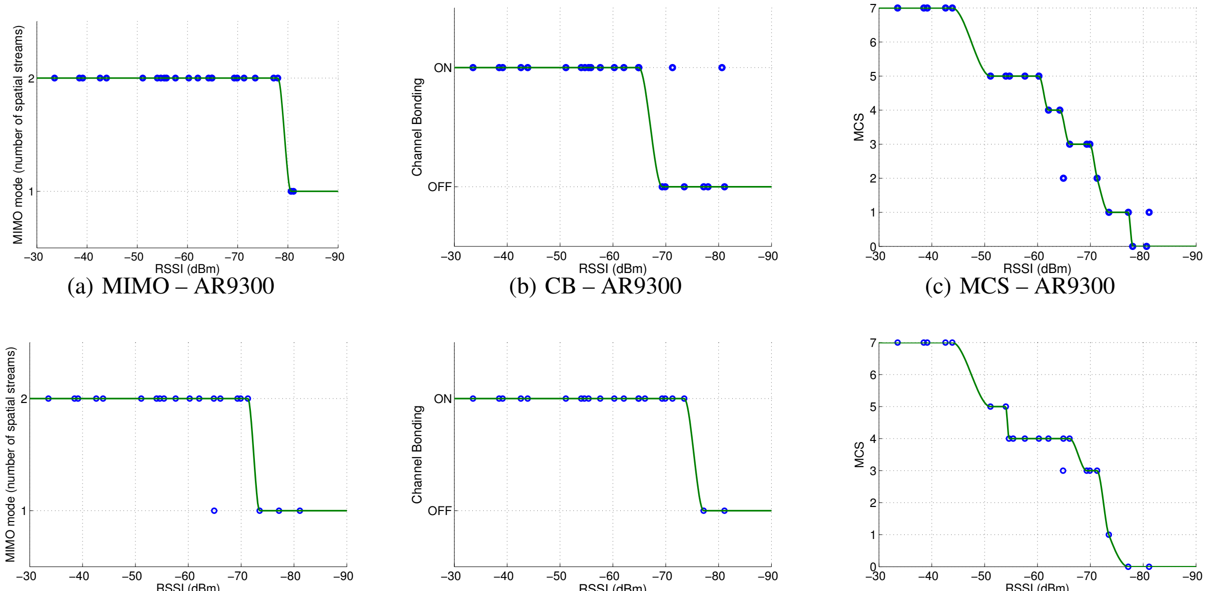
All the above points taken together, as few as 3 feature setting combinations need to be sampled with 802.11n from a total of 64 (8 MCS values x 2 channel bonding x 4 spatial streams), indicating a potential reduction in sampling by 95.3% overall. The 802.11ac with 320 combinations in total (10 MCS values x 4 channel bonding options x 8 spatial streams) offers even higher potential savings by over 99% across all features. These potential savings in sampling overhead are summarized in Table 3. For 802.11n, we experimentally verify that estimated savings in Table 3 can be achieved while a similar experimental validation for 802.11ac is left as future work.

Note that we did not explicitly consider the frame aggregation feature here as we find that existing schemes like Minstrel HT [3] already have an efficient way to adapt this feature that is implemented in the ath9k driver. The approach taken to adapt the degree of frame aggregation is guided by previous analytical studies (e.g., [13]) — higher the bit-rate as determined by the settings chosen for the underlying 802.11n PHY features (MIMO mode, channel bonding and MCS), the larger the size of the aggregated frame.

5.2 Design & Implementation Details

Our proposed solution for 802.11n link adaptation is based on the insight just described on the monotonic relationship between best setting of each feature and average RSSI of a link, and is named SampleLite. The average RSSI is measured on the sender side from the k recent RSSI measurements; k is set to 10 in our implementation. The use of sender side RSSI is common. For example, CHARM [14] relies on such measurements and exploits the channel reciprocity in a practical SNR-based rate adaptation scheme for legacy 802.11 WLANs in dynamic environments.

Relationships between best feature settings and RSSI (Fig. 5 using the AR9300 chipset) serve as reference curves in deciding which feature setting combinations to sample as discussed before. These curves need to be calibrated for different types of hardware and radio configurations (e.g., transmit power). In SampleLite, this calibration is done by tracking the packet error rates. We leave a detailed exploration of this calibration component in SampleLite as part of future work. It is however worth noting that the calibration in our case is simpler than with schemes like CHARM [14]. This



(a) MIMO – AR9300 (b) CB – AR9300 (c) MCS – AR9300

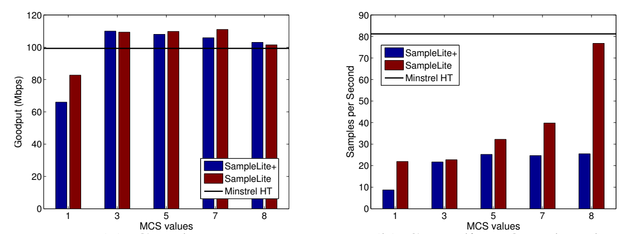
(d) MIMO – AR9220 (e) CB – AR9220 (f) MCS – AR9220

Figure 5: Monotonic relationship between feature settings providing maximum goodput and RSSI, using two different hardware platforms. Each data point corresponds to a different experiment scenario in terms of link and interference type.

is because CHARM and other similar schemes rely on RSSI measurements to select the actual rate for data transmissions, whereas we use them to decide only the sampling subspace; thus there is more room to err from our use of RSSI.

We define two variants of SampleLite. In the vanilla SampleLite, we only exploit the MCS related monotonic relationship with RSSI (Fig. 5(c)), whereas in SampleLite+ we exploit all three monotonic relationships (Fig. 5(a)–5(c)) also including MIMO mode and channel bonding. We implement both SampleLite and SampleLite+ in the ath9k driver. We experimentally determine the parameter i in SampleLite and SampleLite+ that reflects the range of MCS values to sample. Fig. 6 illustrates the impact of using different values for i on goodput and sampling overhead for one of our experiment scenarios. Range of MCS values = 1 implies i = 0, range of MCS values = 3 implies i = 1 and so forth. We also include the case of sampling all MCS values as in Minstrel HT. We see that choosing the smallest value for i or sampling all MCS values leads to lower goodput or higher sampling overhead. So we set i = 1 in our implementation of SampleLite and SampleLite+.

With SampleLite+, the sampling space is reduced by a factor of four compared to SampleLite with the hardware we use because it only samples one setting each for MIMO modes (number of spatial streams) and channel widths. And SampleLite with i = 1 has a sampling overhead that is 3/8ths of what is needed with exhaustive sampling schemes like Minstrel HT. Goodput and sampling overhead with Minstrel HT (for the same experiment scenario) are also shown in Fig. 6; results that span all scenarios are presented in the next section. Note that, unlike Minstrel HT, SampleLite and SampleLite+ do not downgrade the bit-rate in response to high rate of losses. In addition, since they only focus on a small sampling subspace with fewer feature setting combinations, the rate of sampling can also be correspondingly lower. In our implementation, sampling frequency of SampleLite and SampleLite+ is reduced in comparison with Minstrel HT by a factor of four and five, respectively.



(a) Goodput (b) Sampling Overhead

Figure 6: Goodput and sampling overhead of SampleLite and SampleLite+ with different range of MCS values of i: 0, 1, 2, 3, compared to Minstrel HT.

The pseudo-code for SampleLite+ scheme is presented in Algorithm 1 (we get the SampleLite algorithm by removing lines 8–17 and sampling all MIMO modes and channel widths in lines 18–19). The sampling frequency used is as explained above. The statistics table update involves estimating the probability of successful transmission, as well as the estimated goodput to be provided with the specific setting combination. Finally, the optimal setting that is selected, is the one with maximum goodput. Note that once the statistics information is considered old (i.e., using an exponential weighted moving average similarly as in Minstrel HT), the specific entry is given a lower weight. RSSI thresholds obtained from our hardware platform to focus the sampling of MIMO modes, channel widths and MCS values (see Figs. 5(a), 5(b) and 5(c)) are shown in Tables 4, 5 and 6.

Although we present SampleLite and SampleLite+ as separate schemes, they could be combined into a scheme that allows switching between them to achieve further robustness in highly dynamic and interference-prone environments. This idea is along the same

Algorithm 1: SampleLite+ Algorithm

```
while 1 do
  if waitTime ≥ samplingFrequency then
    for MCSi = 7 to 0 do
      if avg(RSSI) ≥ Threshold(MCSi) then
        n = MCSi;
      end if
    end for
    if avg(RSSI) ≥ Threshold(MIMO) then
      streams = 2;
    else
      streams = 1;
    end if
    if avg(RSSI) ≥ Threshold(ChannelBonding)
    then
      width = 40;
    else
      width = 20;
    end if
    sampleRandomSetting S' from :
    [MCS ∈ [n − 1, n + 1], streams, width];
    updateStatisticsTable(S');
    reset waitTime;
  else
    waitTime + +;
  end if
  chooseOptimalSettingforTx(statisticsTable);
end while
```

RSSI Threshold	MIMO: Number of Streams
$avg(RSSI) \geq -79$	2
$-79 > avg(RSSI)$	1

Table 4: RSSI thresholds to infer the MIMO mode to sample for SampleLite+, based on Fig. 5(a).

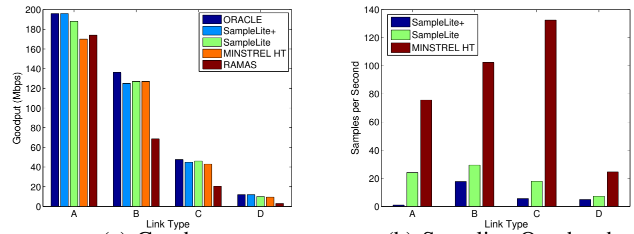
RSSI Threshold	Channel Bonding
$avg(RSSI) \geq -67$	40 MHz
$-67 > avg(RSSI)$	20 MHz

Table 5: RSSI thresholds to infer the channel width to sample for SampleLite+, based on Fig. 5(b).

RSSI Threshold	MCS value $\in [0, 7]$
$avg(RSSI) \geq -45$	7
$-45 < avg(RSSI) \geq -49$	6
$-49 < avg(RSSI) \geq -61$	5
$-61 < avg(RSSI) \geq -65$	4
$-65 < avg(RSSI) \geq -70$	3
$-70 < avg(RSSI) \geq -73$	2
$-73 < avg(RSSI) \geq -78$	1
$-78 < avg(RSSI)$	0

Table 6: RSSI thresholds for choosing the MCS value n to sample for SampleLite and SampleLite+, based on Fig. 5(c).

lines as slight widening of sampling space in SampleLite to handle outliers. Detailed investigation of an unified version that switches adaptively between the SampleLite variants is left for future work.



(a) Goodput (b) Sampling Overhead

Figure 7: Performance in controlled experiments with varying link quality and no interference.

6. EVALUATION

In this section, we evaluate the performance of SampleLite using the indoor 802.11n WLAN testbed across diverse link types and interference scenarios (see §3) relative to Minstrel HT [3], RAMAS [4] and an ORACLE. Minstrel HT is the default scheme with the ath9k driver, whereas RAMAS was shown in [4] to outperform other link adaptation schemes including MiRA [2] and original ath9k 802.11n rate control algorithm [15]. To realize ORACLE, we measure the goodput obtained from using each feature setting combination and pick the maximum among them. We could not include ARAMIS [5] because its implementation is not available. Although effective SNR metric from [8] could in theory provide the performance upper bound in any scenario, as already noted in Section 2, it relies on CSI which is not widely implemented including on Atheros chipsets we used in our testbed.

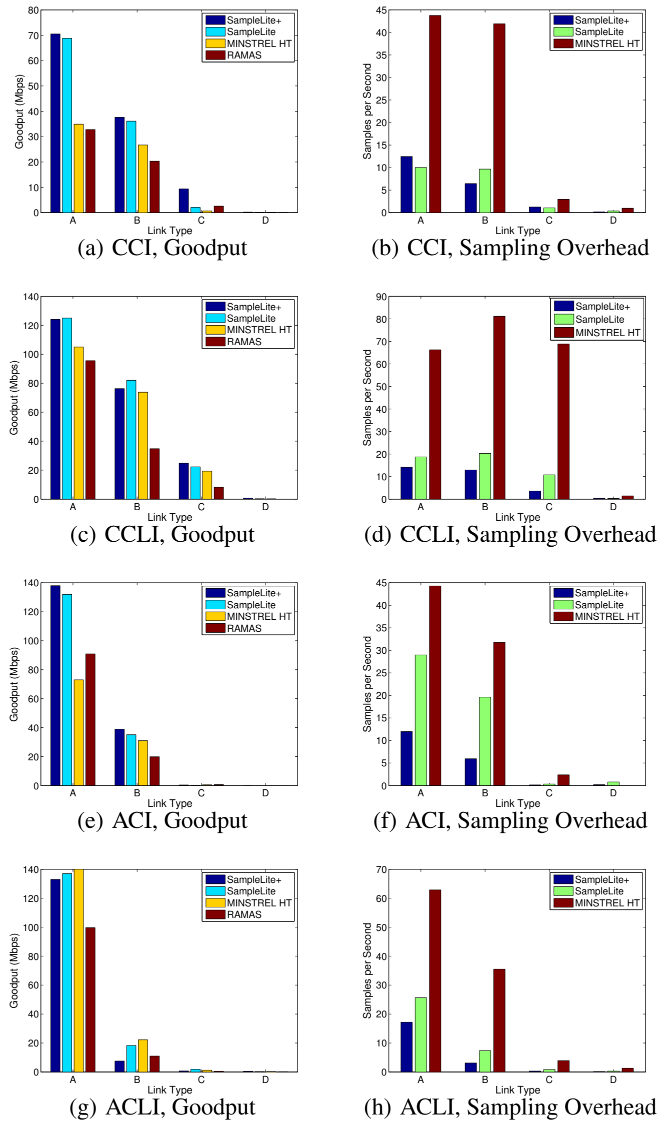
6.1 Controlled Experiments

6.1.1 Effect of Link Type

Fig. 7 shows the impact of different link qualities in the absence of any interference. We observe that both SampleLite and SampleLite+ achieve goodput quite close to the ORACLE. Compared to Minstrel HT, SampleLite+ improves goodput by up to 25% and reduces the sampling overhead on average by 90% because of its targeted sampling. Relative to RAMAS, both SampleLite and SampleLite+ provide up to a 3-fold goodput improvement (which happens for link type D). Reasons for the poor performance of RAMAS are elaborated shortly. Note that from here on we do not include the ORACLE performance. This is because interfering link uses a link adaptation algorithm (specifically, Minstrel HT) in the interference scenarios, making these scenarios dynamic and realization of the ORACLE impractical.

6.1.2 Controlled Co-Channel and Adjacent Channel Interference Effect

Fig. 8 studies the effect of interference. We see that SampleLite mostly outperforms both Minstrel HT and RAMAS in terms of goodput on average by 27.2% and 63.3%, respectively. Average goodput improvement with SampleLite+ compared to Minstrel HT and RAMAS is 33.7% and 94.3%, respectively. Worse performance with Minstrel HT in the presence of interference is because it responds frequently and rapidly to increase in frame losses by reducing the number of streams and rate. This compounds the effect of interference as transmissions take longer and increase the contention level and likelihood of collisions. SampleLite and SampleLite+ avoid this problem by relying on RSSI for choosing the feature setting combinations to sample and select the one providing maximum expected goodput for data transmissions. RAMAS performs



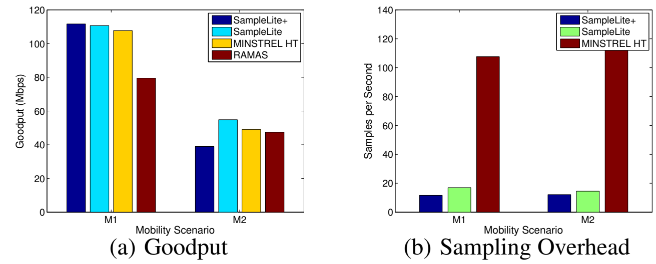
(a) CCI, Goodput
(b) CCI, Sampling Overhead
(c) CCLI, Goodput
(d) CCLI, Sampling Overhead
(e) ACI, Goodput
(f) ACI, Sampling Overhead
(g) ACLI, Goodput
(h) ACLI, Sampling Overhead

Figure 8: Performance in controlled experiments with different interference scenarios and link types.

poorly because its credit based scheme is conservative in adapting the number of streams, and aggressive in adapting the MCS. This mismatch, also noted in [5], causes RAMAS to often operate at sub-optimal settings with single stream and high MCS values, leading to higher losses and reduced performance. This is more apparent as link quality deteriorates.

As per the performance of the interfering link, note that its link quality is similar to link type A (Table 1) and it is placed very close to the AP (Fig. 1). Therefore, its goodput performance is halved when competing against link type A in CCI scenario (Fig. 8(a)) in the absence of any hidden terminals. Even when competing against hidden terminals like the weak links (link types C and D), the interferer's goodput performance remains largely unaffected (over 150Mbps).

In terms of sampling overhead, SampleLite achieves an average reduction of 70.5% compared to Minstrel HT, and a maximum of 86.5%. SampleLite+ reduces it even more by 83% on average



(a) Goodput
(b) Sampling Overhead

Figure 9: Performance in scenarios with mobility.

and up to 98.7%. This significant decrease in sampling is a result of RSSI-guided sampling approach adopted in SampleLite+ and SampleLite that is fundamentally different from the random and exhaustive sampling (e.g., Minstrel HT); the savings in sampling overhead contribute to the goodput improvements with SampleLite+ and SampleLite.

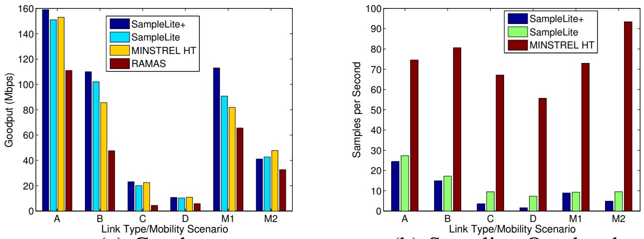
Unlike previous work, SampleLite minimizes the sampling overhead by initiating the sampling only when the current RSSI information varies from the recent average one. During sampling, the expected packet delivery probability and the expected goodput are stored for each possible setting combination. In all hidden terminal cases (link types C and D in Fig. 8), we notice that as Minstrel HT samples more (more frequently and also more MCS values that are not sampled in SampleLite due to RSSI based filtering), it disrupts the stability of the system and results in lower goodput performance compared to SampleLite.

6.1.3 *Mobile Scenarios*

Next, we study the performance of SampleLite variants with station mobility that makes the environment dynamic and causes frequent channel fluctuations. Specifically, we create mobility scenarios M1 and M2 (Fig. 1), where the mobile user walks at a pedestrian speed ($\sim 1m/sec$). M1 exhibits better link qualities with RSSI $\in [-39, -52]$ dBm compared to M2 which observes RSSI $\in [-56, -68]$ dBm. In Fig. 9 we see that even in these dynamic scenarios SampleLite manages to deliver competitive goodput relative to Minstrel HT and RAMAS while providing significant reduction in sampling overhead (87% on average) compared to Minstrel HT. SampleLite provides higher goodput than SampleLite+ in the more challenging of these two scenarios as it samples more widely.

6.2 In the Wild Experiment

We now study the performance of SampleLite variants in the wild with several APs (and their associated clients) belonging to other operational WLANs that share channels with our testbed nodes in an uncoordinated manner. Specifically, we configured our testbed AP and associated clients during peak office hours (2-4pm) to use channel 44 in the 5GHz band on which we found that there are 12 other APs operating on the same channel. In Fig. 10(a), we observe that, even in this challenging environment, SampleLite+ provides up to 38% goodput compared to Minstrel HT and SampleLite variants outperform RAMAS by over 100% in goodput. As per sampling overhead (Fig. 10(b)), SampleLite+ and SampleLite respectively reduce it by up to 97% and 87% in this real-world scenario. However, in scenario M2, SampleLite variants are up to 15% worse than Minstrel HT, in goodput. To understand why, we look into the lower layer statistics, specifically the number of frame retransmissions as a percentage of total number of transmissions. These statistics are summarized in Table 7 and correspond



(a) Goodput (b) Sampling Overhead

Figure 10: Performance in a scenario with several real-world, uncontrolled interferers.

	A	B	C	D	M1	M2
SampleLite+	2%	6%	22%	46%	14%	5%
SampleLite	3%	14%	51%	23%	14%	16%
Minstrel HT	3%	17%	48%	29%	18%	21%
RAMAS	2%	15%	35%	26%	13%	23%

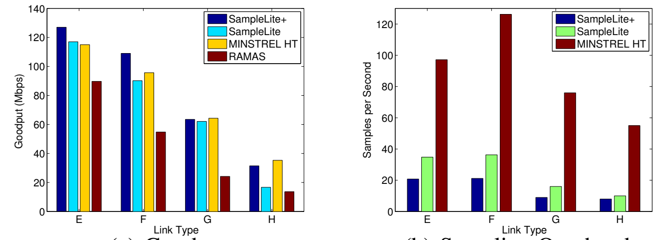
Table 7: Average percentage of retransmissions per link type in the real-world interference scenario shown in Fig. 10.

to Fig. 10. We see that SampleLite+ has the least retransmissions whenever it outperforms all other algorithms. However, it is surprising that though it has only 5% retransmissions in the weak mobile scenario (M2), it gives worse goodput than Minstrel HT with about 22% retransmissions. By looking into the details even further, we find that SampleLite+ had successful transmissions using lower rates based on low measured RSSI, whereas Minstrel HT used on average higher MCS rates, resulting in losses, but also transmitting more data. More specifically, for the M2 scenario in Fig. 10,SampleLite+, SampleLite and Minstrel HT mostly use MCS 3, 4 and 7, respectively.

Focusing again on the same "in the wild" scenario (Fig. 10), in Table 8 we present results on the amount of sampling done by Minstrel HT relative to SampleLite+ to appreciate the negative impact due to sampling. While these results demonstrate that Minstrel HT devotes greater percentage of transmissions to sampling, it does not reflect that some of these sampling related transmissions consume more airtime when sampling lower MCS values in the case of link types A–C and M1. We also see that sampling related transmissions constitute a greater fraction of overall transmissions as link quality deteriorates, as expected.

6.3 Robustness to Environments

Here we test the robustness of our approach by using it in a different type of environment (home). Different link types in this new environment are summarized in Table 9. As in the previous section, this environment is naturally subject to uncontrolled interference from neighboring home WLANs. Specifically, we identified that there are multiple APs present in the area, two of which are operating in the same or neighboring channels and in very close proximity to our AP. Fig. 11(a) shows that SampleLite variants can still perform similar or better than Minstrel HT and RAMAS even in a different environment, and that only hardware changes necessitate re-calibration of thresholds. In the case of position H (weak link with NLoS) we see that Minstrel HT outperforms SampleLite. We believe that the same reason as M2 scenario in the previous subsection applies here. Concerning the sampling overhead, SampleLite variants reduce it by more than 80% compared to Minstrel HT (Fig. 11(b)).



(a) Goodput (b) Sampling Overhead

Figure 11: Performance in the home environment with real-world, uncontrolled interferers.

	A	B	C	D	M1	M2
SampleLite+	0.18%	0.22%	0.39%	0.45%	0.10%	0.21%
SampleLite	0.22%	0.21%	0.68%	1.04%	0.12%	0.29%
Minstrel HT	0.60%	1.15%	4.70%	5.67%	1.14%	2.66%

Table 8: Average percentage of sampling packets per link type in the real-world interference scenario shown in Fig. 10.

Link	RSSI	LoS?	Link Type
E	-41 dBm	No	Strong
F	-58 dBm	No	Strong–Medium
G	-65 dBm	No	Weak–Medium
H	-78 dBm	No	Weak

Table 9: Average RSSI values (in dBm) for each link type in the home environment.

7. CONCLUSIONS

We have considered the link adaptation problem in 802.11n WLANs and showed that it is vital to adapt all key 802.11n features in order to maximize goodput. Observing that most existing schemes suffer from excessive sampling or implementation concerns, we design an efficient and practical scheme called SampleLite that takes a novel hybrid approach. SampleLite relies on easily accessible sender-side RSSI measurements to identify a small subset of rates to sample, thus reducing the sampling overhead. Through a testbed evaluation considering a wide range of controlled and real-world interference scenarios, we show both SampleLite variants significantly reduce the sampling overhead by over 70% compared to Minstrel HT. We also show that goodput-wise SampleLite provides substantial gain relative to Minstrel HT and RAMAS by up to around 35% and 100%, respectively. Our future work will focus on further improving the robustness and agility of SampleLite in highly dynamic and interference-prone environments by adaptively widening the sampling space, and its application to 802.11ac networks with newer features.

8. REFERENCES

[1] IEEE 802.11n. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Enhancements for Higher Throughput, 2009.

[2] I. Pefkianakis, Y. Hu, S. Wong, H. Yang, and S. Lu. MIMO Rate Adaptation in 802.11n Wireless Networks. In *Proc. ACM MobiCom*, 2010.

[3] F. Fietkau. Minstrel_HT: New Rate Control Module for 802.11n. http://lwn.net/Articles/376765/, March 2010.

[4] D. Nguyen and J. J. Garcia-Luna-Aceves. A Practical Approach to Rate Adaptation for Multi-Antenna Systems. In *Proc. IEEE ICNP*, 2011.

[5] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. Joint Rate and Channel Width Adaptation for 802.11 MIMO Wireless Networks. In *Proc. IEEE SECON*, 2013.

[6] ath9k 802.11n Wireless Driver. http://linuxwireless.org/en/users/Drivers/ath9k.

[7] R. Combes, A. Proutiere, D. Yun, J. Ok, and Y. Yi. Optimal Rate Sampling in 802.11 Systems. In *Proc. IEEE INFOCOM*, 2014.

[8] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Predictable 802.11 Packet Delivery from Wireless Channel Measurements. In *Proc. ACM SIGCOMM*, 2010.

[9] R. Crepaldi, J. Lee, R. Etkin, S. Lee, and R. Kravets. CSI-SF: Estimating Wireless Channel State using CSI Sampling and Fusion. In *Proc. IEEE INFOCOM*, 2012.

[10] X. Xie, X. Zhang, and K. Sundaresan. Adaptive Feedback Compression for MIMO Networks. In *Proc. ACM MobiCom*, 2013.

[11] L. Deek, E. Garcia-Villegas, E. Belding, S. Lee, and K. Almeroth. The Impact of Channel Bonding on 802.11n Network Management. In *Proc. ACM CoNEXT*, 2011.

[12] L. Kriara, M. K. Marina, and A. Farshad. Characterization of 802.11n Wireless LAN Performance via Testbed Measurements and Statistical Analysis. In *Proc. IEEE SECON*, 2013.

[13] T. Yoo, R. Lavery, A. Goldsmith, and D. Goodman. Throughput Optimization Using Adaptive Techniques. http://msw3.stanford.edu/ zhuxq/adhoc/letter_adaptive.pdf, 2004.

[14] G. Judd, X. Wang, and P. Steenkiste. Efficient Channel-Aware Rate Adaptation in Dynamic Environments. In *ACM MobiSys*, 2008.

[15] M. Wong, J. Gilbert, and C. Barratt. Wireless LAN using RSSI and BER Parameters for Transmission Rate Adaptation. *US patent 7369510*, 2008.