Verification of ns-3 Wi-Fi Rate Adaptation Models on AWGN Channels

Juan Leon
University of Washington
Seattle, Washington, USA
juanleon@uw.edu

Thomas R. Henderson
University of Washington
Seattle, Washington, USA
tomhend@uw.edu

Sumit Roy
University of Washington
Seattle, Washington, USA
sroy@uw.edu

ABSTRACT

The performance of Wi-Fi networks depends on the ability of devices to adapt their transmissions to dynamic channel/network conditions. Hence, "Rate Adaptation Algorithms (RAAs)" have been devised to allow nodes to select appropriate modulation and coding schemes (and other parameters) in response to varying channel/network conditions. These algorithms are neither standardized nor typically divulged by vendors, and devising a 'performance-optimal' RAA for specific scenario remains an active topic that necessitates a complex, multi-parameter cross-layer (PHY/MAC) approach. The ns-3 network simulator offers detailed models of the Wi-Fi medium access control (MAC) layer, including three reference RAA implementations; however testing and validation of these RAA models has been very limited to date. This paper reports on initial test and validation for ns-3 RAA models via 802.11n/ac/ax simulations. After describing the RAA scope and implementations, we explore and summarize insights from test results as to a) whether the ns-3 RAAs are able to achieve the correct rates as configuration is varied and b) how they respond to step changes in the received signal-to-noise ratio (SNR) as a means for exploring their convergence properties.

CCS CONCEPTS

• **Networks** → **Network simulations**.

KEYWORDS

ns-3, network simulation, Wi-Fi rate adaptation algorithms

ACM Reference Format:
Juan Leon, Thomas R. Henderson, and Sumit Roy. 2023. Verification of ns-3 Wi-Fi Rate Adaptation Models on AWGN Channels. In *2023 Workshop on ns-3 (WNS3 2023), June 28–29, 2023, Arlington, VA, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3592149.3592162

1 INTRODUCTION

Rate Adaptation Algorithms (RAAs) are an integral part of wireless communications, to enable reliable and stable link operations in *dynamic* conditions arising due to node mobility and/or time-varying channel and interference scenarios. RAAs enable reactive updating of Modulation and Coding Scheme (MCS) based on current assessment of link/network conditions, as indicated by Received Signal Strength (RSS), Packet Error Rates (PER) or other suitable measurement metrics. Hence RAAs are fundamental to optimizing network operations and continue to be actively studied via network simulation in ns-3 (e.g., [5], [7]). As of the current ns-3.37 release, ns-3 contains models for three RAAs for 802.11n/ac/ax simulations: IdealWifiManager, MinstrelHtWifiManager and ThompsonSamplingWifiManager. Despite continuing interest, no structured campaign has been conducted to rigorously test the ns-3 RAA implementations and to cross-validate with implementations or prior simulation works. The lack of such benchmarking is underscored by the fact ns-3 currently has little to no test code for either MinstrelHt or ThompsonSampling models, and the project's issue tracker lists several open issues. While some test code is available for the IdealWifiManager, it has been reported [1] that its performance is suboptimal. This work aims to start to fill the above gaps, by capturing results from our efforts in evaluating the various RAAs as applied to some basic Wi-Fi scenarios. After initial descriptions of their operations, we describe results from tests of each in controlled single-link (two-node) scenarios involving the change of link SNR over time, for various parameter configurations. Using this simple scenario, we explore the following questions:

(1) Is each RAA able to reach the correct transmission rate for all possible parameter configurations?
(2) How fast do the RAAs converge to a step change in received SNR, and what differences do we observe?

The literature on RAAs is significant and spans decades; for an example of a recent taxonomy and survey, see [5]. One reason that RAA continues to be a fertile area of active research is that the problem is inherently cross-layer, multi-parameter and multi-objective optimization, inevitably involving multiple tradeoffs. A key element of any RAA is the ability to sense current channel/network state and incorporate filtered past observations, as predictors of future channel/network conditions. Implementation constraints (ability to sample good measurement from devices, resultant algorithmic complexity subject to real-time operation, etc.) also impact the design of RAAs. ns-3 only provides a (non real-time) simulation implementation and lacks necessary sophisticated PHY abstraction for fast fading channels, channel state information feedback and other hardware imperfections, etc. Nonetheless, given the continuing interest in RAAs, this paper seeks to characterize the existing ns-3 implementations for simple, baseline PHY scenarios as benchmark. A thorough study of ns-3 RAA performance would involve more sophisticated scenarios and time-varying channels (fading channels with coherence times allowing for adaptation), but we offer this work as a starting point for basic ns-3 validation. We expect that as the PHY abstractions are improved, so too will the capabilities for more interesting RAA research with ns-3.

WNS3 2023, June 28–29, 2023, Arlington, VA, USA

ACM ISBN 979-8-4007-0747-6/23/06.
https://doi.org/10.1145/3592149.3592162

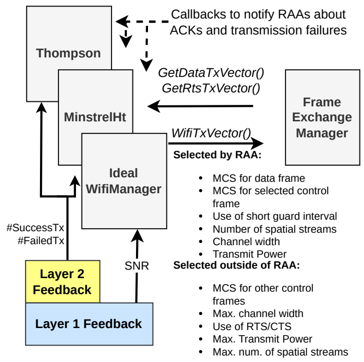


Figure 1: ns-3's Implementation of RAAs

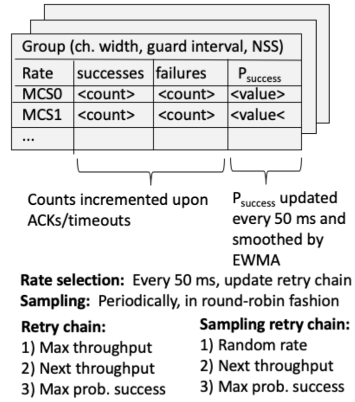


Figure 2: Overview of MinstrelHtWifiManager Algorithm

2 RATE ADAPTATION IN NS-3 WI-FI

RAAs can be classified into two categories: *measurement-based* and *sampling-based*. Measurement-based algorithms use physical layer (PHY) measurements, such as averaged Received Signal Strength Information (RSSI) for received frames, to estimate current channel/network condition and react accordingly. Measurements that deliver per-subcarrier signal-to-noise (SNR) ratio samples offer most fine-grained PHY information but are likely difficult to realize in practice. Sampling-based algorithms on the other hand, do not make use of measurement data at the physical layer but instead probe the Layer-2 packet channel by trying different rates according to some heuristic. Both Layer-1 (PHY) measurement and Layer-2 sampling is typically filtered by low pass filters such as an exponentially weighted moving average (EWMA). For modern Wi-Fi standards (IEEE 802.11n and later), ns-3 has only one model of a measurement-based approach, the so-called *IdealWifiManager*, and two sampling-based models, the *MinstrelHtWifiManager* and the *ThompsonSamplingWifiManager*.

In principle, an RAA could dynamically adapt multiple transmission parameters - transmit power, RTS/CTS (Request To Send/Clear To Send) handshake, spatial diversity or multiple streams, dynamic channel bonding, etc. However in ns-3 implementation, the current RAAs only adapt the modulation and coding scheme (MCS), the use of short guard intervals for 11n/ac (if both stations support), the channel width, and the number of spatial streams, for both data and selected control frames. As depicted in Figure 1, when channel access is obtained, the *FrameExchangeManager* asks the RAA to provide a *WifiTxVector* object conforming to the available channel width and other transmission constraints that may exist (such as maximum number of spatial streams supported).

The ns-3 *WifiRemoteStationManager* is the base class for ns-3 RAA models, responsible for selecting *physical layer parameters* for the upcoming data, control, and management frame transmissions, and for storing information regarding link capabilities to each remote station, such as whether a short guard interval is supported. To track network dynamics and adapt transmission rate, the *WifiRemoteStationManager* incorporates results from each frame transmission (based on feedback from Wi-Fi unicast recipients) to create a summary of current network state. A *ConstantRateWifiManager* is also available for static assignment of MCS.

The operation of RAAs in ns-3 is driven by the underlying PHY error models governing whether receptions fail or succeed. For example, later in the paper, the plots of Figure 7 depict 'waterfall' curves corresponding to the error rate performance of different MCS values corresponding to IEEE 802.11ac standard and binary convolutional code (BCC) encoding, for an additive white Gaussian noise (AWGN) channel. As the SNR increases, higher MCS rates become available. The exact breakpoint between the selection of adjacent MCS values depends on the details of the RAA.

2.1 IdealWifiManager

The *Ideal rate adaptation* algorithm in ns-3 takes its name from the use of an (idealized) feedback channel that reports, to the transmitter, the most recently observed SNR value from that transmitter at the receiver. The transmitter uses this reported SNR, as well as error tables, to pick an MCS that meets a specific performance threshold; namely, to pick an MCS such that the long-term BER will be lower than a prescribed BER threshold (which defaults to 10^{-6} BER). Due to selecting a BER threshold, this algorithm may not be truly "Ideal" because it might be too conservative depending on the channel state. This algorithm is based on the Receiver Based Auto Rate (RBAR) approach in the literature [2] and is a measurement-based RAA that performs no filtering of past SNR history– only using the most immediate history. The Wi-Fi ACKs (or block ACKs) are used to deliver the received SNR to the transmitter through the use of an ns-3 packet tag, a data field that is not available in a real 802.11 ACK frame. The IdealWifiManager uses the last observed SNR for a given station, along with the error model tables being used for the Wi-Fi channel model, to select the highest-rate MCS and supported configuration that satisfies the BER criteria. For RTS frames, this algorithm will pick the highest rate within the (legacy) basic rate set; the use of High Throughput (HT) modes is not supported for RTS, although it is allowed in the standard. If both stations can use short guard intervals, then typically the use of short guard interval will be selected because it offers a higher rate.

2.2 MinstrelHtWifiManager

MinstrelHt (Figure 2), extending the legacy *Minstrel* model, is based on the Linux implementations of the same name. MinstrelHt is

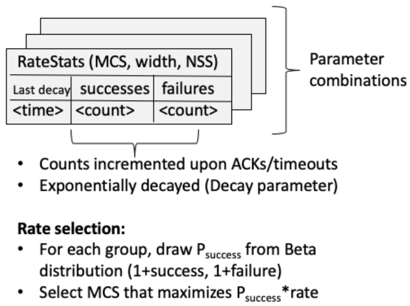


Figure 3: Overview of ThompsonSamplingWifiManager

a sampling-based approach that organizes possible combinations of rate control parameters into 'groups', and tracks the history of successful and failed receptions for each MCS in each group. MinstrelHt is difficult to succinctly summarize (Chapter 5 of [1] is a thorough overview of the ns-3 model), but Figure 2 captures the essence of the approach. MinstrelHt uses sampling and maintains statistics of its groups and rates within, to identify three rates in particular: 1) the maximum throughput rate, 2) the next highest throughput rate, and 3) the rate with the highest probability of success. MinstrelHt uses these three rates in what is called a 'retry chain', corresponding to the initial and (possible) retransmissions of a frame. Most frames are sent according to this retry chain, but MinstrelHt also uses a small number of frames to replace the maximum throughput rate with a randomly selected rate (to explore performance of additional groups and rates). Randomly selected rates are selected more-or-less in a round-robin fashion. Statistics on the probability of success for each MCS, and on the average Aggregated MAC Protocol Unit (A-MPDU) length, are aged out every 50 ms by an Exponentially Weighted Moving Average (EWMA) that may be represented [1] by the following equation:

$$y[n] = EWMA/100 * y[n-1] + (100 - EWMA)/100 * x[n] \quad (1)$$

In Equation 1, $y[n]$ is the probability of success for the current MCS. $y[n-1]$ is the previous probability of success for the current MCS. $x[n]$ is the proportion of success for the current MCS. The default value of $EWMA$ in ns-3 is 75; a smaller value will cause older values to age sooner. This algorithm has evolved in Linux due to concerns about convergence, including replacing the EWMA with a different low-pass filter, as described in [1], but the ns-3 implementation has not been updated. Like IdealWifiManager, when choosing parameters for an RTS frame, MinstrelHt picks a basic rate less than or equal to the rate in use for data frames, with a normal guard interval and using a single channel and spatial stream. Short guard interval, if configured for both stations, can also be exploited.

2.3 ThompsonSamplingWifiManager

Finally, the *ThompsonSamplingWifiManager* (Figure 3) implements a rate adaptation algorithm based on the Thompson sampling algorithm. Thompson sampling is a Bayesian statistical technique used to solve the multi-armed bandit (MAB) problem. The work by [6] proposed to use this algorithm as an RAA, and in the work by Krotov [3], an ns-3 implementation was developed and used as a performance baseline for another research algorithm; both references can be consulted for more information about Thompson sampling in general. The problem of rate selection for Wi-Fi can be framed as a MAB problem with the objective to select the MCS that maximizes throughput despite uncertainty in the success probability for any MCS (either initially, or due to changing channel conditions). For each transmission opportunity, ThompsonSampling will sample from the underlying Beta random variable for each MCS, and calculate a value which is the product of this (probabilistic) frame success rate and the PHY rate that results from that rate. It then picks the maximum value from among all of the MCS. The *alpha* and *beta* parameters of the underlying Beta distribution are simply the number of successes and failures experienced for each MCS. These counts are incremented upon ACK or block ACK receptions, and are aged by an exponential decay factor:

$$count(t) = count(t - \Delta t) * e^{-Decay * \Delta t} \quad (2)$$

where the default ns-3 value of $Decay$ is 1. Unlike MinstrelHt, ThompsonSampling will not explore groups with lower than the allowed channel width or number of spatial streams allowed for the remote station (focusing its sampling only on varying the MCS). ThompsonSampling will use the most robust MCS to send RTS and other control frames and will also limit them to use only one spatial stream and a single channel. ThompsonSampling will configure the use of short guard interval if both stations support it.

2.4 Discussion

In all of the above algorithms, the use of short guard interval is not consistent with how it may be used in practice. In practice, short guard interval is used for environments that exhibit a low amount of multipath. It has been reported [4] that short guard interval is not typically employed unless a station has reached the best MCS available. Only if the highest MCS is reached will a station try to enable short guard interval once it has experienced success with the normal guard interval. In ns-3 RAAs, short guard interval is directly and immediately tried as an option if it is supported by both stations, and since the ns-3 channel and error models do not typically penalize the choice (there is no high delay spread channel environment modeled), there is only benefit and no risk to enabling it. Future refinement of the physical layer for channels with long delay spread should provide different performance to different guard interval selection, and in addition, ns-3 should consider to adopt the procedure defined in [4] of only enabling it at the highest MCS.

3 SISO VERIFICATION RESULTS

We present results to verify correct operation of the three ns-3 RAAs for single input, single output (SISO) links, with an exploration of convergence time over non-fading (AWGN) channels.

3.1 Basic Operation

To confirm the basic operation, we created a script that configures each possible combination of channel width, spatial streams and guard interval for a given standard (802.11n, 802.11ac, or 802.11ax OFDM). Our script is derived from that developed by Grunblatt [1] and is similar to the *wifi-manager-example.cc* script in the ns-3 codebase. We observe the highest rate selected by each manager and

Table 1: ns-3 Parameters for Variable Distance Test

Simulation Parameter	Value
ns-3 Version	3.37
Wi-Fi Standard	802.11ac
Rate Adaptation Algorithm	Ideal, MinstrelHt or ThompsonSampling
Spatial Streams	1
Channel Width	20 MHz
Short Guard Interval	Enabled
Propagation Loss Model	Log Distance
Application	Saturating 1420 byte packets
Preamble Detection Model	disabled
Simulation Duration	30 seconds (five trials averaged)

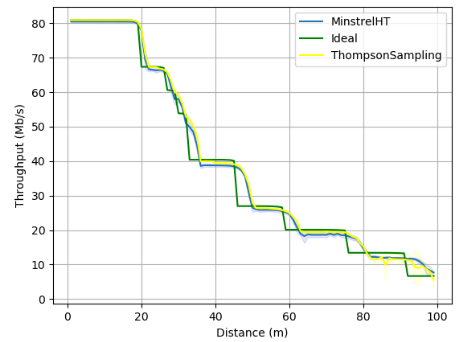


Figure 4: SISO Variable Distance Test

compare them across possible configurations and with published MCS tables. The ns-3 RAAs are able to select appropriate MCSes (in alignment with MCS tables and AWGN error tables) in most cases. The exception is the ThompsonSampling RAA when low SNR is used. Unlike Ideal and MinstrelHt implementations, the ns-3 ThompsonSampling has no heuristic to quickly drop to a robust MCS when faced with high packet losses (due to low SNR). This problem has already been reported in the ns-3 issue tracker[1]. A heuristic favoring more robust rates for certain retransmission events might remedy this problem.

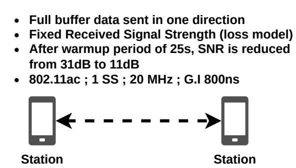
We also observed that MinstrelHt samples from a wider range of possible configurations than does ThompsonSampling. For example, if the allowed channel width is 80 MHz, ThompsonSampling will tend to sample only from 80 MHz channel width configurations, but MinstrelHt will also explore 40 MHz and 20 MHz options. As a result, MinstrelHt convergence could take longer for 802.11ax than for other standards, due to more combinations to sample.

We adapted a scenario developed by Grunblatt [1], in which a station as source is separated by a given distance from an AP operating as sink, and transmits saturating traffic. Each simulation trial lasts 30 seconds, and uses the ns-3 FlowMonitor to record throughput over the duration of the data transfer. An external script calculates and plots the sample average and standard deviation of throughput observed in five simulation runs. The distance is increased in steps of 1 meter and the experiment is repeated. Results are provided for parameters in Table 1. The minimum and maximum AP-STA distance are 1 and 99 meters, respectively. Figure 4 shows that as the distance increases, all ns-3 RAAs provide comparable throughput.

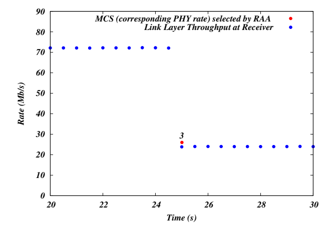
3.2 Convergence

One key aspect of RAAs is their ability to quickly converge to an appropriate MCS in response to changing channel conditions. Upon a change in the channel that affects reception statistics, both the ns-3 MinstrelHt and ThompsonSampling models require time to sample possible alternative configurations. This section examines the response of both models to the most basic exploratory scenario: a step change in the received SNR of the channel.

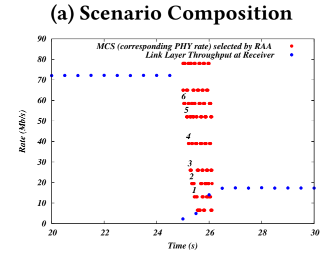
3.2.1 General Behavior. In the following time-series figures, we illustrate how different RAAs respond and converge when a sudden change is applied to the channel, as depicted in Figure 5a. RAA performance is quantified via the link layer throughput between two nodes, one of which is transmitting data to the other in saturation mode. The Received Signal Strength (RSS) is set to a constant value (SNR of 31 dB, which leads to an 802.11ac MCS of 8) for a duration of 25 seconds. At 25 seconds, a 20 dB SNR drop occurs, and each RAA is then responsible for adapting to the new channel.



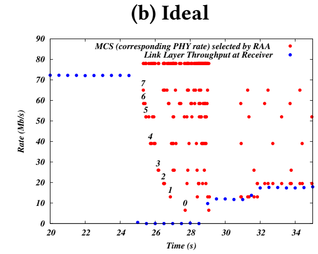
(a) Scenario Composition



(b) Ideal



(c) MinstrelHt



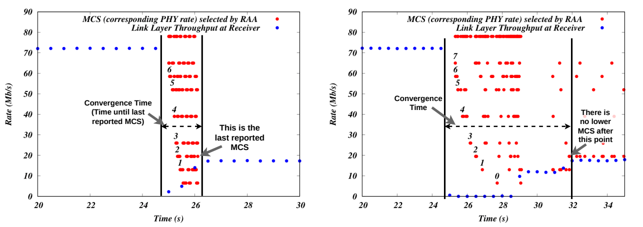
(d) ThompsonSampling

Figure 5: Time Series Plots of Convergence to a New MCS

Ideal RAA immediately selects the expected MCS at the start of the simulation by using the received SNR tag piggybacked onto control frames. Ideal RAA quickly converges (convergence time depends on the delay to receive the new SNR tag as shown in Figure 5b), after a change in channel, by making a single new MCS selection for new channel condition.

When MinstrelHt builds its sample table, it acquires statistics on the channel state for each group and MCS, and then selects the best MCS, spatial streams, guard interval, and channel bandwidth based on its channel history. The convergence process of MinstrelHt after stepping down from MCS 8 is shown in Figure 5c. When there is a drop in SNR at time 25 seconds, the previous history becomes

[1] https://gitlab.com/nsnam/ns-3-dev/-/issues/414



(a) MinstrelHt Convergence Definition **(b) ThompsonSampling Convergence Definition**

Figure 6: Convergence Definition

invalid, and thus MinstrelHt samples other MCS values and evolves the history using an EWMA. During the transition, all lower MCS values are selected at some point because the history suggests that they could offer the highest throughput, since all the MCS values previously had worked during sampling. However, after the channel degrades, MCS 7, 6, 5, and 4 all result in transmission failures, leading to a momentarily lower throughput because the highest supportable MCS is 3. After learning enough about the new channel, MinstrelHt converges on the highest sustainable throughput offered by MCS 3, as observed in the plot by no further changes in the selected MCS after around 26 seconds.

In ThompsonSampling, the algorithm strongly prefers the MCS value that leads to the highest PHY rate, so long as the frame success ratio is high enough, as explained in previous sections. Initially, because all of the underlying Beta random variables are similarly distributed, the algorithm will favor the highest supported MCS, and will not subsequently sample lower MCS values. However, at time 25 seconds, the channel conditions change and the success probability of the highest MCS drops significantly, leading to the selection of lower MCS values. In general, ThompsonSampling has a preference for higher rates due to the following heuristic: $SelectedMCS = MAX(FrameSuccessProbability * PHYrate(MCS))$. Figure 5d shows the response of ThompsonSampling to the change in channel conditions. After 28 seconds, the algorithm continues to periodically vary the selected MCS; the periodicity is controlled by the Decay factor, a configurable parameter that determines how quickly ThompsonSampling forgets its previous history and attempts higher rates again.

3.2.2 Convergence Experiment. To explore convergence time behavior, we constructed an experiment based on the scenario described above in Figure 5a, checking the ability of the RAA to respond to a drastic change in SNR. However, this experiment differs in that both managers (ThompsomSampling and MinstrelHt) were evaluated by sweeping the magnitude of the SNR change across a wide range, and by starting from both a high SNR and a low SNR initial starting point. Specifically, the first configuration involves a decrease in SNR from an initial signal-to-noise ratio (SNR) of 32 dB. This is accomplished by decreasing the SNR with a granularity of 0.1 dB, and the time taken to converge is recorded. This process is repeated in 100 randomized trials to obtain an average value, and the resulting average convergence times are displayed in the plots. The second configuration differs only in that it involves an improvement in channel conditions, starting from an initial SNR of 1 dB. Each simulation trial lasts for 20.5 seconds, allowing for warmup of 0.5 seconds, followed by 10 seconds of data transfer at the initial SNR, and 10 seconds of observing the behavior after the change in channel at 10.5 seconds.

The primary objective of this experiment is to explore convergence trends in a very simple, controlled experiment. Although MinstrelHt and ThompsonSampling are both sampling-based algorithms, they do have substantial differences in implementation. For instance, MinstrelHt chooses and prefers the 'best' rate for an interval, whereas ThompsonSampling does not officially converge on a best rate but operates on dynamically changing probability tables. Although not presented herein, we also have observed that the speed of convergence is affected by the configuration of the MinstrelHt EWMA and the ThompsonSampling Decay factor. Because of the differences between MinstrelHt and ThompsonSampling, we defined convergence times in slightly different ways. For MinstrelHt, we define convergence time (Figure 6a) as the time, after a channel change, until the last reported MCS selection. This can be done in ns-3 by leveraging the fact that MinstrelHt will not report MCS changes for sampled rates; only changes to the selected MCS are reported. Unlike MinstrelHt, ThompsonSampling will report all sampled MCS values through its trace source, so we cannot use a similar convergence definition. Instead, for ThompsonSampling, we define convergence time (Figure 6b) by first looking at the lowest MCS value selected during the last second of simulation time, and then looking backward from that time, finding the point in time after which no lower MCS value is selected.

Figure 7 presents the results of the experiment. Four subplots are shown according to the combination of RAA and direction of SNR change. Each subplot shows two datasets and two Y axes on a shared X axis. The X axis represents the resulting SNR value after the SNR was changed. The colored data series represent the average convergence times (left Y-axis), and the gray dashed lines (right Y-axis) show the error rate performance of each MCS value (ranging from 0 to 8) as a function of SNR and packet size, with MCS 0 as the left-most curve and MCS 8 the right-most curve. We included the error rate curves because they explain some of the behavior of the convergence curves.

All subplots demonstrate underlying trends of increase or decrease in convergence time as a function of the resulting SNR value. However, they also exhibit very strong peaks at several SNR points. Each peak corresponds to a transition between two MCS values, as the algorithms have difficulty converging when there is a tradeoff between the higher throughput offered by a higher MCS and the better frame success ratio offered by a lower MCS at these regions.

Focusing first on the experiment shown in Figure 7a, notice that the convergence time increases with the size of the SNR drop (i.e., as the resulting SNR value is lower). This trend can be attributed to the way in which MinstrelHt statistics evolve over time. Specifically, when fast rates are used, or when sampling occurs, a larger number of attempts can be made for a higher MCS (i.e., a specific number of MPDUs sent using that MCS), compared to slower rates, which limit the number of attempts that can be made within a given time interval. Consequently, as more attempts are made for a particular MCS, MinstrelHt can accumulate more information on that MCS, leading to a faster convergence time. An additional interesting aspect of the plot is the wider peaks observed in regions where the

SNR is less than 10.8 dB. We hypothesized that an additional effect was coming into play in these regions. Wi-Fi acknowledgments are sent at selected basic rates of 6, 12, or 24 Mbps, and at certain SNR regions, the selection of an appropriate acknowledgment rate can be difficult, leading to lost acknowledgments as well. To confirm this hypothesis, we conducted an experiment in which control frames were forcibly set to use 6 Mbps to ensure their robustness. The results are not shown here due to space, but this modified experiment also resulted in narrow peaks for these low SNR regions.

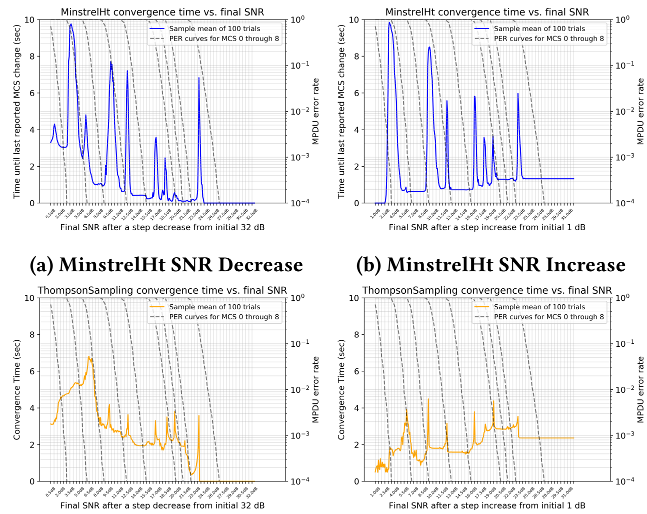
Figure 7b presents results from an increase of SNR from an initial value of 1 dB, for MinstrelHt, and shows similar peaks in the convergence times. Interestingly, a slight trend is observed, where larger increases in SNR correspond to slower convergence times. This behavior can be attributed to MinstrelHt first converging to a supportable but less optimal MCS. Moreover, the trend is found to be weaker than the previously described trend, due to MinstrelHt's avoidance of re-sampling already-used rates or very robust rates (those with an empirical weighted moving average probability (EWMAProb) of greater than 95%). These heuristics allow MinstrelHt to sample other rates, even if the currently used rate is performing successfully, and converge to a better rate quickly.

The next two subplots repeat the previous experiments but for ThompsonSampling. Figure 7c shows a noticeable trend in which an increase in the magnitude of the drop in SNR leads to a corresponding increase in the convergence time. This trend can be attributed to the fact that prior to the change in the channel, Thompson had already built up successful statistics for all rates up to and including the supported MCS 8. However, after the channel change, the statistics for the previously successful MCS values need to be lowered by attempting those rates and failing, until the newly supported MCS is learned. Consequently, a larger drop in SNR will necessitate that Thompson Sampling gather information on more MCSs to ensure that the optimal MCS is not erroneously chosen. It is noteworthy that the plot deviates from this trend when the resulting SNR is less than 5dB. This is because when the supported MCS is greater than MCS 0, Thompson Sampling may occasionally lock onto a less than optimal MCS due to its robustness. Then, it takes some time for the statistics to decay enough for ThompsonSampling to attempt a higher rate that may be successful. In contrast, if the supported MCS is MCS 0, there are no other MCSs for ThompsonSampling to potentially lock onto.

Observations made from Figure 7d reveal a pattern that is similar to that observed in Figure 7b. Specifically, an increase in SNR results in an increase in the convergence time. This phenomenon can be attributed to the tendency of ThompsonSampling to converge on suboptimal rates that are robust, thereby necessitating the decay of probabilities to a point where a better rate can be selected. The only region that doesn't follow this trend corresponds to resulting SNR greater than 23 dB. We have yet to explain this region.

4 CONCLUSIONS

This paper summarized the ns-3 RAA implementations and described their current behavior and limitations via experiments in simple two-link scenarios with variable SNR. All ns-3 RAAs were verified to achieve appropriate MCS in most situations; the ThompsonSampling algorithm could use better heuristics to cope with low SNR channels, and MinstrelHt may benefit from more heuristics to improve convergence for 802.11ax OFDM. We observed that the use of 802.11n/ac/ax short guard interval was not in line with reported practice, and could be an area of improvement for all algorithms. Due to space limitations, we did not present results on the sensitivity of convergence behavior to the exponential decay factors in use by the sampling algorithms; this topic could be a subject of future work. All simulation scripts and instructions are available[2]. We expect future improvements to the ns-3 PHY abstraction to focus on incorporating channel state information feedback for MIMO, and models for operation in time-varying fading channels, which may lead to new or improved RAAs. New RAAs will also be needed in ns-3 for 802.11ax/be features such as OFDMA and spatial reuse.



(a) MinstrelHt SNR Decrease **(b) MinstrelHt SNR Increase**

(c) Thompson SNR decrease **(d) Thompson SNR increase**

Figure 7: Convergence Times as Function of SNR Step Changes

REFERENCES

[1] Rémy Grünblatt, Isabelle Guérin-Lassous, and Olivier Simonin. 2019. Simulation and Performance Evaluation of the Intel Rate Adaptation Algorithm. In *MSWiM 2019 - 22nd ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. ACM, Miami Beach, United States, 27–34.

[2] Gavin Holland, Nitin Vaidya, and Paramvir Bahl. 2001. A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking* (Rome, Italy) *(MobiCom '01)*. Association for Computing Machinery, New York, NY, USA, 236–251.

[3] Alexander Krotov, Anton Kiryanov, and Evgeny Khorov. 2020. Rate Control With Spatial Reuse for Wi-Fi 6 Dense Deployments. *IEEE Access* 8 (01 2020), 168898–168909. https://doi.org/10.1109/ACCESS.2020.3023552

[4] Eldad Perahia and Robert Stacey. 2013. *Next Generation Wireless LANs: 802.11n and 802.11ac* (2nd ed.). Cambridge University Press, USA.

[5] Ibrahim Sammour and Gerard Chalhoub. 2020. Evaluation of Rate Adaptation Algorithms in IEEE 802.11 Networks. *Electronics* 9, 9 (2020).

[6] Viktor Toldov, Laurent Clavier, Valeria Loscrí, and Nathalie Mitton. 2016. A Thompson Sampling Approach to Channel Exploration-exploitation Problem in Multihop Cognitive Radio Networks. In *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. Valencia, Spain, 1–6.

[7] Dong Xia, Jonathan Hart, and Qiang Fu. 2013. Evaluation of the Minstrel Rate Adaptation Algorithm in IEEE 802.11g WLANs. In *2013 IEEE International Conference on Communications (ICC)*. Budapest, Hungary, 2223–2228. https://doi.org/10.1109/ICC.2013.6654858

[2] https://gitlab.com/juanvleonr/ns-3-dev/-/tree/raa-verification