Using Ranging for Collision-Immune IEEE 802.11 Rate Selection with Statistical Learning

Wojciech Ciezobka[a], Maksymilian Wojnar[a,*], Krzysztof Rusek[a], Katarzyna Kosek-Szott[a], Szymon Szott[a], Anatolij Zubow[b], Falko Dressler[b]

[a] *AGH University of Krakow, Poland*
[b] *TU Berlin, Germany*

Abstract

Appropriate data rate selection at the physical layer is crucial for Wi-Fi network performance: too high rates lead to loss of data frames, while too low rates cause increased latency and inefficient channel use. Most existing methods adopt a probing approach and empirically assess the transmission success probability for each available rate. However, a transmission failure can also be caused by frame collisions. Thus, each collision leads to an unnecessary decrease in the data rate. We avoid this issue by resorting to the fine timing measurement (FTM) procedure, part of IEEE 802.11, which allows stations to perform ranging, i.e., measure their spatial distance to the AP. Since distance is not affected by sporadic distortions such as internal and external channel interference, we use this knowledge for data rate selection. Specifically, we propose FTMRate, which applies statistical learning (a form of machine learning) to estimate the distance based on measurements, predicts channel quality from the distance, and selects data rates based on channel quality. We define three distinct estimation approaches: exponential smoothing, Kalman filter, and particle filter. Then, with a thorough performance evaluation using simulations and an experimental validation with real-world devices, we show that our approach has several positive features: it is resilient to collisions, provides near-instantaneous convergence, is compatible with commercial-off-the-shelf devices, and supports pedestrian mobility. Thanks to these features, FTMRate outperforms existing solutions in a variety of line-of-sight scenarios, providing close to optimal results. Additionally, we introduce Hybrid FTMRate, which can intelligently fall back to a probing-based approach to cover non-line-of-sight cases. Finally, we discuss the applicability of the method and its usefulness in various scenarios.

Keywords: Wi-Fi; 802.11; data rate selection; fine timing measurement; statistical learning

1. Introduction

In IEEE 802.11 (Wi-Fi) networks, the goal of rate selection is to choose the modulation and coding scheme (MCS), prior to each data transmission, to achieve the highest possible data rate given current radio conditions. The design of rate selection algorithms for Wi-Fi devices is a significant area of research because of the direct impact on Wi-Fi performance. The need for new algorithms is due to two factors. First, new amendments to the IEEE 802.11 standard increase the selection space with more MCS values, guard interval (GI) lengths, channel widths, etc. For IEEE 802.11ax, Table 1 presents the available data rates for a single spatial stream. Since 802.11ax supports up to eight spatial streams, the number of currently available Wi-Fi data rates increases to 1056. Second, the availability of new tools, such as those based on machine learning (ML) [1], can improve rate selection performance.

We notice that most rate selection algorithms, including those that apply ML, measure the transmission success rate for a given MCS (Section 2). However, under high station density (including hidden stations) and/or under cross-technology interference (e.g., ZigBee, Bluetooth) such a probing approach fails due to frame collisions, which will be misinterpreted as erroneous rate selection. Obviously, transmitters want to adapt their rates to the signal strength at the receiver and not to collision-causing interference from other transmissions. Therefore, an alternative closed-loop approach is to measure channel quality (such as the received signal strength, RSS) at the receiver and send feedback to the transmitter.

In this vein, we have proposed FTMRate [2] – a closed-loop, context-aware and collision-immune rate selection algorithm, which relies not on directly measuring channel quality, but on ranging, i.e., measuring distance using the fine timing measurement (FTM) procedure introduced in IEEE 802.11-2016 [3] (and later extended in 802.11az). With FTM, stations can estimate their distance from the AP using round-trip time (RTT) measurements, predict the expected received signal strength (RSS) at the receiver from the distance, and finally select MCS values based on

[*] Corresponding author: Maksymilian Wojnar, AGH University of Krakow, Faculty of Computer Science, Electronics and Telecommunications, al. A. Mickiewicza 30, 30-059 Krakow, Poland

Email address: `maksymilian.wojnar@agh.edu.pl` (Maksymilian Wojnar)

Article published in Elsevier Computer Communications 225 (2024) 10-26. (10.1016/j.comcom.2024.07.001)

MCS	Modulation	Coding rate	Data rate [Mbit/s]
0	BPSK	1/2	7.3
1	QPSK	1/2	14.6
2	QPSK	3/4	21.9
3	16-QAM	1/2	29.3
4	16-QAM	3/4	43.9
5	64-QAM	2/3	58.5
6	64-QAM	3/4	65.8
7	64-QAM	5/6	73.1
8	256-QAM	3/4	87.8
9	256-QAM	5/6	97.5
10	1024-QAM	3/4	109.7
11	1024-QAM	5/6	121.9

Table 1: 802.11ax transmission rates for a single spatial stream in a 20 MHz channel with a 3.2 µs guard interval.

the expected RSS, as we have shown in [2].

The goal of this work is twofold: (i) to further study the performance of FTMRate and (ii) to address FTMRate's main limiting factor – reliance on line-of-sight (LOS) conditions. Regarding the first goal, the studies that we provide in this paper show that FTMRate has three desirable features: (a) rapid convergence to appropriate rates – since probing is not required, (b) compatibility with commercial-off-the-shelf devices including support for multiple input, multiple output (MIMO) transmissions and transmission power adaptation, and (c) mobility support – the method operates well under pedestrian mobility settings. Regarding the second goal, we introduce the Hybrid FTMRate algorithm, which can intelligently detect non-line-of-sight (NLOS) conditions and switch to a classic rate selection manager to maintain high performance.

The rest of this paper is organized as follows. We first provide a thorough review of the state of the art related to the topic of rate selection in IEEE 802.11 networks, with a special focus on ML-based solutions (Section 2), and a description of FTM operation in Section 3. Then, in Section 4, we describe the core of our unique proposal, FTMRate, as well as its extension – Hybrid FTMRate. Next, we present a thorough simulator-based performance evaluation in multiple scenarios (Section 5). Results show that in LOS scenarios FTMRate is both better than the tested baselines and operates close to the approximate upper bound. The gain depends on the exact scenario, e.g., for a network of 10 stations we provide a 40% gain over an existing non-ML method and a 20% gain over an ML-based method. In NLOS scenarios, Hybrid FTMRate can achieve performance similar to competing solutions. Next, we provide an experimental validation using physical hardware in Section 6 to show that the approach is viable in real-world settings. We address two limitations of FTMRate in Section 7 and discuss the benefits, drawbacks, and applicability of our approach in Section 8. We conclude the paper with a summary of our findings and outline future work in Section 9.

This article extends our previous results in [2]. Our new contributions include a significantly extended literature review (Tables 2 and 3), a detailed discussion of the trade-offs between the three variants of FTMRate (Section 4.2.2), new simulation scenarios highlighting the adaptability of FTMRate in dynamic transmission power settings (Section 5.2), high performance in hidden station scenarios (Section 5.3), and low impact of overhead when in-band signaling is used (Section 5.4). Furthermore, we introduce and evaluate the aforementioned Hybrid FTMRate (Sections 4.3 and 5.5). We provide an experimental validation and comparison with a state-of-the-art rate selection manager proving the validity of the proposed approach (Section 6), and address the outstanding challenges (Section 7), as well as thoroughly discuss the advantages and disadvantages (Section 8).

2. State of the Art

In this section, we review both non-ML and ML-based rate selection methods and state the novelty of our approach.

2.1. Non-ML-based Methods

The transmission rate selection problem has been investigated since the early days of 802.11. The two basic types of classical rate control are sampling-based (that uses statistical knowledge of transmission success rates) and measurement-based (that uses explicit channel measurements).

We restrict our description to two prominent sampling-based algorithms, both later used as baselines, Minstrel [10] and Intel's `iwl-mvm-rs` [11]. Minstrel, the default rate control algorithm of the Linux kernel, calculates the transmission success ratio and constructs a table with per-frame statistics. Additionally, it performs exponential weighted moving average calculations to process the success history of each rate. After that, the rates with the best throughput, second-best throughput, highest success probability, and lowest base rate are considered for transmission (in descending order, with several retry attempts per each rate).

Modern Intel chipsets, which operate using the `iwlwifi` driver, use `iwl-mvm-rs` as the rate selection algorithm. The available rates are divided into several groups and the rates are selected in two interleaving phases [11]: MCS scaling and column scaling. During MCS scaling, the algorithm tries to maximize throughput by changing MCS. During column scaling, `iwl-mvm-rs` looks for a better combination of modes (single or multiple spatial streams), GIs, antenna configurations. First, the lowest parameters are used (which correspond to the lowest throughput and highest reliability) and then the search cycle starts (i.e., the interleaving of the two phases begins). A decision is made based on the success ratio. After the search cycle ends, `iwl-mvm-rs` performs MCS scaling until a new search cycle begins (triggered either by too many successful transmissions from the previous cycle, too many unsuccessful transmissions from the previous cycle, or after 5 s since the previous cycle has elapsed).

Key	ML type	Year	Eval. method	Wi-Fi type	Scenario	RTS/ CTS	Novelty	Improvement	Input	Output
[4]*	NN	2013	S (Qualnet)	802.11b	Two, five, and ten stations. Single AP. Uplink traffic.	Yes	Apply NN for Tx rate selection	Higher throughput than two SOA methods.	Number of contending stations, channel conditions (indicated by BER), traffic intensity.	Consecutive failed and successful transmission thresholds.
[5]	Random forests	2013	S (ns-3)	802.11p	Five stations. Uplink traffic.	No	Apply SL for Tx rate selection.	Higher throughput than three SOA methods.	SNR, speed, propagation distance.	Packet success ratio per data rate.
[6]	Random forests	2018	E, S (Matlab)	802.11ac	Single link.	N/A	Apply SL for channel classification to support Tx rate selection.	Higher spectral efficiency	Magnitude of reference symbol, received preamble.	Channel classification (residential or office).
[7]	NN	2020	E	802.11ac	Five stations. Two APs. Uplink traffic.	N/A	Provide extensible rate selection framework.	Higher throughput than three SOA methods.	Link quality (RSSI, subframe loss rate), features (MCS, MIMO, bandwidth).	Scores of available rates.
[8]	NN	2020	E, S (T-SIMn)	802.11n	Several devices.	N/A	Apply NN for transmission rate selection based on effective throughput.	Higher throughput than two SOA methods.	Effective throughput values of the rates in the sampling set.	Estimated throughput of supported rates.

Table 2: Supervised learning-based rate selection algorithms. Evaluation methods: S – simulation, E – experiments. Papers marked with * involve deep learning. SOA – state of the art, Tx – transmission [9].

2.2. ML-based Methods

ML-based rate control is mainly based on supervised learning (SL) or reinforcement learning (RL) and often uses link quality indicators (e.g., RSS, collision probability) as input (observations). A summary of papers (in chronological order) using supervised and reinforcement learning to improve transmission rate selection is provided in Tables 2 and 3, respectively. Evidently, SL-based approaches are less popular than RL-based ones.

2.2.1. Supervised Learning

Designing an SL-based solution requires defining both an input (feature) set and an output. We discuss the various approaches in the following.

A multilayer perceptron artificial neural network (NN) is adopted in [4] to model the correlation function between the optimal number of consecutive failed/successful uplink transmissions and a set of traffic metrics (number of contending stations, channel conditions measured using the bit error rate (BER), and traffic intensity). In [5], the random forests method is used in IEEE 802.11p networks to select transmission rates based on the predicted probability of successful transmission. Signal to noise ratio (SNR) samples are used to characterize the propagation environment and position information (obtained from satellite navigation) is used to increase the prediction accuracy. Furthermore, [6] uses random forests to classify the channel type (residential or office) to select the appropriate MCS, since MSC strongly depends on the propagation channel quality. Furthermore, [7] implements an ANN that learns the mutual influence of rates, throughput, and channel quality. The data rate is selected based on the observed congestion level (measured as data frame service time).

2.2.2. Reinforcement Learning

In RL, agents learn by iteratively interacting with the environment. As described below, most solutions focus on improving throughput, though some also concentrate on decreasing delay.

Several works implement *stochastic approaches.* In [12], a stochastic learning automata-based rate adaptation algorithm is proposed to guarantee a given packet success ratio by observing successful transmission attempts. In [13], a distributed algorithm is proposed, which is based on the stochastic multi-armed bandit (MAB) approach. It explores different configuration options (e.g., channel bandwidth, MCS) and observes their impact on network performance under varying channel conditions. The frame success ratio is used as a reward and the optimal configuration (i.e., the one with the highest ratio) is used to calculate the regret. Graphical optimal rate sampling is proposed in [14]. The authors assume that the throughput is a unimodal function of the selected (rate, MIMO mode) pair. They show that graphical unimodality can be used to efficiently learn and track the best transmission rates. In [15], Thompson sampling (TS) is used to model the acknowledgment probability for each MCS and maximize throughput. In [16], the transmission rate is adjusted based on channel quality, represented by the signal-to-interference ratio (SINR) and the transmission power.

Key	ML type	Year	Eval. method	Wi-Fi type	Scenario	RTS/CTS	Novelty	Improvement	Observation	Action
[17]	MAB	2011	S (custom), E (SDR)	802.11a/g	Single AP (downlink)	N/A	Apply MAB for best channel and Tx rate selection.	Higher throughput.	Avg. success ratio.	Channel number, Tx rate.
[13]	MAB	2016	S (ns-3)	802.11ac	Up to 20 stations. Uplink traffic.	N/A	Apply MAB for link adaptation (Tx rate selection).	Higher throughput, lower frame loss, and delay than three SOA methods.	Frame success ratio.	Tx rate.
[18]	MAB, TS	2017	E	802.11ac	Six APs, 20 stations.	N/A	Apply MAB for link adaptation in mobile environments (select Tx rate).	Better performance than two SOA solutions.	Packet error rate, SNR value calculated using exponential weighted moving average, statistical table.	Channel bonding, MIMO spatial streams, MCS.
[14]	MAB	2018	S, E	802.11	Nine downlink streams.	Yes	Apply MAB for Tx rate adaptation.	Better results than two SOA methods.	Success rate for given Tx rate.	Tx rate.
[15]	MAB, TS	2018	S (custom)	802.11	Single link	N/A	Apply classical TS for link adaptation, select Tx rate.	N/A	Transmission success/ failure.	Tx rate.
[16]	TS, PF, MAB	2020	S (ns-3)	802.11ax	Three scenarios: single link, two overlapping networks (each with a single link), residential building. Downlink traffic.	N/A	Apply TS and PF for statistical-based MCS selection.	Lower delay, higher throughput.	Tx rate, success rate.	MCS level.
[19]*	Deep NN	2020	E	802.11ac	Eight APs and 30 stations. Uplink and downlink traffic.	N/A	Apply DRL for link adaptation (select Tx rate), the DRL component is placed at APs.	Higher throughput, lower packet loss and delay than three SOA methods.	Channel noise, selective fading under MIMO, channel interference.	MCS level, number of spatial streams, channel bonding.
[20]	QL	2021	S (ns-3)	802.11a	Single link.	No	Use packet timeouts to train RL model to select the best Tx rate.	Higher throughput than a SOA method.	Packet timeout events.	MCS level.
[21]	Latent TS	2021	S (custom)	LTE, but applies also to 802.11	Single link.	N/A	Apply latent TS to model probability distribution of channel SINR, select Tx rate.	Better results than for unimodal TS and outer loop link adaptation.	SINR, ACK/NACK.	MCS level.
[22]*	Deep QL	2021	E	802.11ac	Office environment w/ and w/o interference. Uplink and downlink traffic.	N/A	Apply DRL for Tx rate selection.	Higher throughput than two SOA methods.	Per-rate sub-frame loss, RSSI, and service time ratio.	Moving directions along 3D maze: bandwidth, MCS, MIMO mode.
[23]	QL	2022	S (custom)	802.11ax	Overlapping networks: 16 APs and 16 stations (one per AP). Uplink and downlink traffic.	No	Apply RL for per-cell Tx rate selection in OBSS scenarios.	Better results than [24, 25].	Throughput (frame-by-frame, period-by-period).	Tx rate.
[26]	QL	2023	S (custom), E	802.11ax	Pair of nodes in different environments: office, hallway, classroom.	No	Automatic parameter tuning based on Q-learning.	Better results than [24, 27].	Packet error statistics.	Automatic parameter tuning.

Table 3: Reinforcement learning-based rate selection algorithms. Evaluation methods: S – simulation, E – experiments. Papers marked with * involve deep learning. SOA – state of the art, Tx – transmission.

Additionally, a particle filter (PF) is used to estimate channel quality (SINR). Finally, with latent TS [21] SINR is modeled in terms of its probability distribution over a range of SINR values, updated based on the feedback received (ACK or NACK). Additionally, the SINR probability distribution is relaxed in every step with a smoothing function to adapt to fading channels.

Other works consider *more complex approaches.* The authors of [23] apply Q-learning (QL) for rate adaptation in overlapping network scenarios. In [20], packet timeouts are used to train the agent. Choosing an MCS is an action and the resulting contention window (CW) size is a state. Additionally, in [26], QL is used to automatically tune the parameters of an adaptive outer loop link adaptation algorithm, which selects MCS based on packet error rate measured in a given time window. In [22], deep QL allows adapting to the current link quality and channel conditions. Rate selection features (MCS, MIMO mode, channel width) are treated as different coordinates of a 3D maze. Each 3D cell of the proposed maze represents a rate. Then, a deep reinforcement learning (DRL) model is used, in which the action space "includes two moving directions along three dimensions", the state includes six elements (channel width, MIMO mode, MCS, sub-frame loss, received signal strength indication (RSSI), service time ratio), and the resulting goodput is treated as a reward.

2.3. Novelty of Proposed Approach

Based on the above literature review, we conclude that our core proposal, FTMRate, differs from other state-of-the-art approaches based on machine learning, e.g., TS [28] and DRL [29]. These general methods assume no knowledge about how the channel behaves, and the only feedback comes from each frame transmission, either successful or not. While an FTM measurement could be yet another signal from the environment (called the 'context' in RL literature) that would improve the accuracy of these methods, it also would pose some issues. In the case of TS, its simplicity is lost when contextual bandits are introduced since we have no simple posterior formula. Meanwhile, a general DRL agent could exploit FTM information but such an approach is computationally expensive and the DRL agent needs to learn what is already known, i.e., have prior knowledge of how the communication channel works. In this paper, working in the paradigm of context-aware communication, we fully embrace knowledge of the wireless channel to build a simple yet efficient (collision-immune, power-adaptable, computationally inexpensive, rapidly converging) agent based on probabilistic modeling.

3. Distance Estimation Using Fine Timing Measurement

Before explaining the operation of our proposal, we first describe FTM as a standalone procedure to estimate the distance between communicating Wi-Fi devices. FTM is an active, point-to-point protocol for positioning, with an accuracy exceeding that of RSS-based positioning [30]. FTM operation is based on a burst of frame exchanges as shown in Fig. 1. First, the *initiating station* sends a trigger frame (FTM Request) to the responding station to initiate the measurement procedure. Then, a set of FTM frames and ACKs are exchanged between the responding station and the initiating station. The goal is for the initiating station to obtain four timestamps for the i-th frame exchange ($t_{1,i}$ to $t_{4,i}$) so as to compute the RTT:

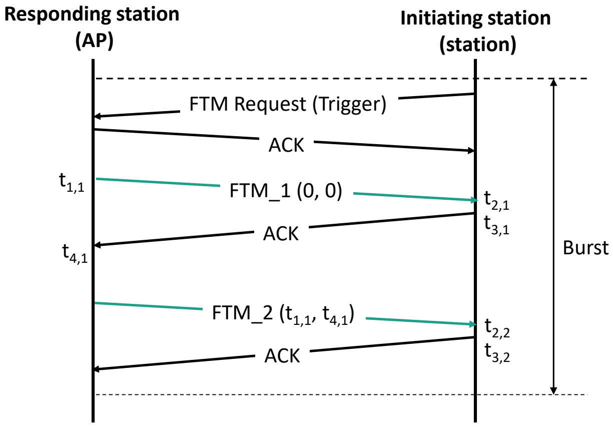


Figure 1: FTM operation for the shortest possible burst [9].

$$RTT_i = (t_{4,i} - t_{1,i}) - (t_{3,i} - t_{2,i}), \tag{1}$$

where $t_{1,i}$ is the time at which the i-th FTM frame was transmitted by the AP, $t_{2,i}$ is the time at which the i-th FTM frame was received by the initiating station, and $t_{3,i}$ is the time at which the initiating station transmitted the acknowledgment of the correct reception of the i-th FTM frame to the AP, $t_{4,i}$ is the time at which the acknowledgment of the i-th FTM frame was received by the AP [9]. FTM requires n frame exchanges to calculate $n-1$ RTTs. Therefore, the minimum number of the FTM frames exchanged within a burst is $n = 2$, as presented in Fig. 1, where a single RTT is measured in the burst, and the $t_{1,1}$ and $t_{4,1}$ timestamps are transferred by the AP to the initiating station in the second FTM frame (FTM_2).

The i-th distance estimation (ρ_i^{RTT}) can be calculated from the i-th RTT as

$$\rho_i^{RTT} = \frac{RTT_i}{2} c, \tag{2}$$

where c is the speed of light.

The signaling overhead required to perform an FTM operation depends i.a. on the size of the burst. Zubow et al. have found that "in scenarios with strong multi-path environments, e.g., indoors, there is no gain from using larger channel bandwidth and higher number of measurements" [31]. Thus, the airtime consumed by the shortest possible FTM burst (Fig. 1) consists of transmitting an FTM request frame, two FTM frames, and three ACK

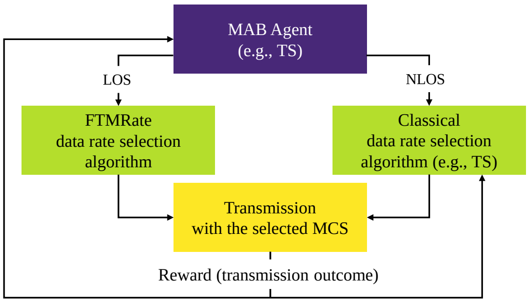


Figure 2: Hybrid FTMRate design.

frames. A promising solution to solve the overhead problem would be to perform FTM out-of-band (since the distance remains the same regardless of radio frequency). IEEE 802.11be will provide multi-link operation and it is easy to envision that while the main communication between stations and APs is done in the 5 or 6 GHz bands, FTM bursts can be relegated to the 2.4 GHz band. In the following, we mainly focus on rate selection performance improvements achieved by estimating the distance with FTM but also address the signalling overhead.

Regarding localization accuracy, FTM tends to slightly overestimate distances, with errors typically remaining under 1 m [31]. As shown in the following sections, these approximations are sufficient for rate selection. In addition, we use filtering techniques to address any deficiencies resulting from these approximations.

4. Hybrid FTMRate Approach

We propose Hybrid FTMRate which extends our previous contribution, FTMRate [2], by combining it with a classical rate selection approach to mitigate one of FTMRate's drawbacks, namely reliance on LOS operation (Fig. 2). In this section, we first describe general system settings and assumptions, then give an overview of the core of the proposal (i.e., FTMRate operation) and finally describe how Hybrid FTMRate extends this concept.

4.1. System Settings and Assumptions

We consider devices equipped with single IEEE 802.11ax radio interfaces, operating in infrastructure mode, with fixed channel width and guard interval values, performing FTM measurements at a rate of once every 0.5 s (2 Hz). Furthermore, we assume the following:

- static or low-mobility (nomadic) stations,
- an exponential Gaussian error model of FTM measurements based on [31],
- a log-distance path loss model and Nakagami multipath fading,
- perfect RSS to MCS mapping.

These assumptions allow to study (Hybrid) FTMRate performance and we revisit and discuss them in Section 7.

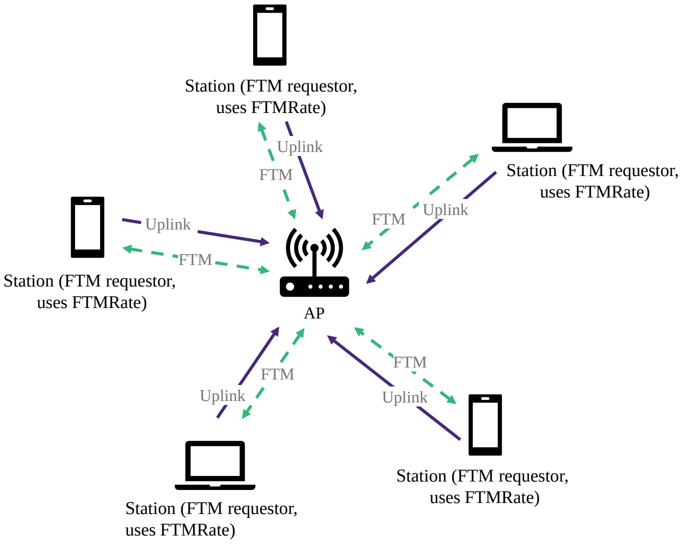


Figure 3: General FTMRate scenario. Stations are FTM initiators, while the AP is the FTM responder. Additionally, stations use FTMRate for selecting their transmission rates, based on the results of FTM ranging.

4.2. FTMRate Overview

In FTMRate, each 802.11 station runs an independent ML agent that selects rates (MCS values). In the construction of this agent, we use probabilistic modeling, which is crucial as FTM measurements are inaccurate and 802.11 is far from being a simple system. Probabilistic modeling means that every unknown is modeled as a random variable with some probability distribution. Certain complicated aspects of the system are also abstracted by statistical learning models. Under some mild assumptions, we can reduce the measurement error by including past measurements in the model. The net result of such an approach is the distribution of the possible rate for each MCS at every point in time. This distribution allows the selection of the optimal MCS and control of the FTM probing rate to reduce overhead.

The general FTMRate scenario is illustrated in Fig. 3. Each station estimates its distance from the AP at time t using the FTM procedure illustrated in Fig. 1. The true distance is denoted as ρ_t and its noisy estimation as ρ_t^{RTT}. This noise is caused by imprecision inherent in the FTM procedure and caused by the wireless chipset, multi-path propagation effects, channel bandwidth, etc. The channel model is assumed to be known and has the form of a conditional distribution of RSS given the distance between the sender and the receiver[1]: $p_{\boldsymbol{\theta}_c}(\gamma|\rho)$, where γ is the RSS and $\boldsymbol{\theta}_c$ is a vector of the channel model (c) parameters. In the following, the subscript denotes parameters of a function, e.g., $p_{\boldsymbol{\theta}_c}(\cdot) := p(\boldsymbol{\theta}_c, \cdot)$. A frame transmission is successful with probability $\xi = s_{\boldsymbol{\theta}_s}(\gamma, \mu)$, where μ is the MCS used for the transmission and $s_{\boldsymbol{\theta}_s}$ is the CDF of the sinh-arcsinh distribution (s). Since distance measurements are noisy,

[1] Channel model estimation and prediction is left for future study.

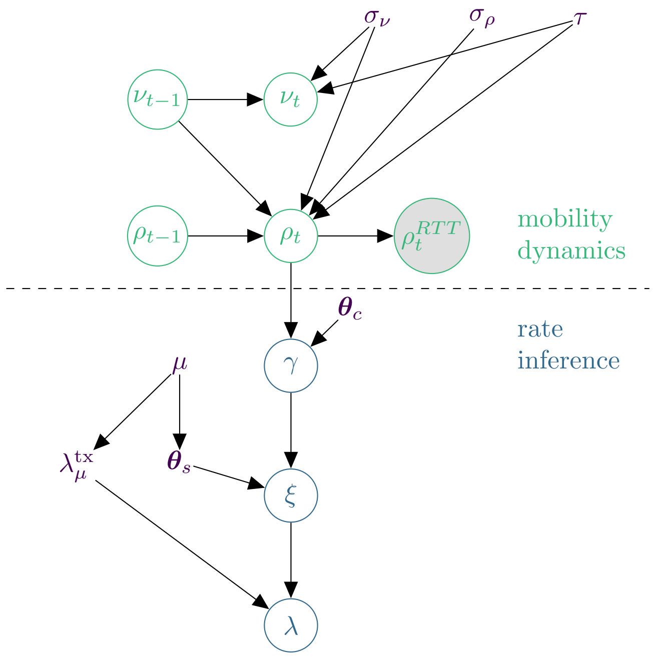


Figure 4: Graphical representation of the proposed model that relates distance observations with a hidden state and the rate achievable at a given MCS. Each arrow represents the statistical dependence between the random variables (circled) and the parameters (not circled). The graph comprises two parts: station mobility dynamics (top) and rate inference (bottom).

a smoothing filter is applied to obtain a more accurate estimate of the actual distance.

The graphical model we use is depicted in Fig. 4: the top part represents mobility dynamics (used to smooth the noisy FTM readings), and the bottom part represents MCS inference. In the following, we describe both parts in detail. The notation is summarized in Table 4.

4.2.1. Mobility Dynamics

Upon each measurement at discrete time t, the agent obtains a noisy reading ρ_t^{RTT} of the true distance between the transmitter and the receiver. We may reduce the distance uncertainty by the sequential filtering of observations. In particular, we assume that the true distance ρ evolves with the local linear trend ν and both obey the following stochastic differential equations [32]:

$$\begin{cases} \mathrm{d}\nu &= \sigma_\nu \mathrm{d}W_1 \\ \mathrm{d}\rho &= \nu \mathrm{d}t + \sigma_\rho \mathrm{d}W_2, \end{cases} \tag{3}$$

where $\mathrm{d}W_1$ and $\mathrm{d}W_2$ are independent Wiener processes and the observations are assumed to be the true distance with noise ϵ_t: $\rho_t^{RTT} = \rho_t + \epsilon_t$, whose distribution could be either centered normal with known variance or exponentially modified Gaussian, also with known parameters [31]. Furthermore, $\sigma_\nu \in \mathbb{R}_+$ is the variation in velocity and $\sigma_\rho \in \mathbb{R}_+$ is the variation in distance.

This approach is inspired by the physics of Brownian motion. We assume that each station has a latent radial velocity ν that is subject to random changes. The variation of these changes is controlled by σ_ν and setting $\sigma_\nu = 0$ gives us the dynamics of [16]. As in physics, the infinitesimal change of distance ρ depends on the current value of ν, however, since this is just an approximation, we capture other effects, which change ρ, as additive noise. The amount of this noise is controlled by the σ_ρ parameter.

Name	Definition
α	level exponential smoothing weight
β	trend exponential smoothing weight
γ	RSS
Δ	difference between default and current transmission power
δ_t	trend transition noise
ϵ_t	distance measurement error
ε_t	level transition noise
$\boldsymbol{\theta}_c$	channel model parameters
$\boldsymbol{\theta}_s$	parameters of *sinh-arcsinh* normal distribution
$p_{\boldsymbol{\theta}_c}(\gamma\|\rho)$	channel model
λ	expected rate
λ_μ^{tx}	transmission rate for the μ-th MCS
μ	current MCS
ν	radial velocity
ν_t	trend (velocity like) at discrete time t
ξ	probability of successful transmission
ρ	true distance
ρ_t^{RTT}	noisy reading of true distance from the AP
ρ_t	distance from the AP at discrete time t
σ_ν	variation of velocity
σ_ρ	variation of distance
τ	time between measurements or the time since the last measurement
$s_{\boldsymbol{\theta}_s}$	CDF of the sinh-arcsinh distribution

Table 4: Notation used.

The particular choice for random processes representing noise is also an approximation, as it allows for negative distances. In practice, this is not a problem, and the analytical solution is a great advantage of such a model; however, an exact model is also proposed and described in the Appendix.

The solution to (3) is known as the Ornstein–Uhlenbeck process, a continuous-time stochastic process. In practice, the process is observed (sampled) at discrete times $t = t_{2,i+1}$, which yields the following discrete linear dynamical system:

$$\begin{cases} \nu_{t+1} &= \nu_t + \delta_t, \\ \rho_{t+1} &= \rho_t + \nu_t \tau + \varepsilon_t, \\ \rho_t^{RTT} &= \rho_t + \epsilon_t, \end{cases} \tag{4}$$

where τ is either the time between measurements $\tau = (t_{2,i+2} - t_{2,i+1})$ or the time since the last measurement, and both process (transition) noises, δ_t and ε_t, have a joint multivariate normal distribution:

$$\begin{bmatrix} \delta_t \\ \varepsilon_t \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_\nu^2 \tau & \frac{\sigma_\nu^2 \tau^2}{2} \\ \frac{\sigma_\nu^2 \tau^2}{2} & \tau \left(\frac{\sigma_\nu^2 \tau^2}{3} + \sigma_\rho^2 \right) \end{bmatrix} \right). \tag{5}$$

If FTM were costless and instantaneous, an agent could trigger it upon each frame transmission. However, since it introduces overhead, we must keep the measurement rate as low as possible to protect link capacity. The key to achieving high throughput is a discretized continuous-time model that can be used at a random time while being updated at discrete times. This necessitates an inference procedure, which we outline in the next section.

4.2.2. Ranging Inference

The internal state (ρ, ν) of (3) can be inferred from observations at discrete time points, while a continuous model extrapolates dynamics of frame transmission times that occur between measurements. We observe that the uncertainty increases with time and that a new measurement makes the estimate more certain. The trend component ν increases the inertia of the model, thus allowing longer times between measurements if the object moves without sudden acceleration. Inference can be done using any of the popular methods, such as the Kalman filter (KF) or particle filter (PF). The selected method is dictated by the assumption about the distribution of the measurement error. For a normal distribution, we have a linear Gaussian state space model and the Kalman filter is an analytical solution for posterior inference [33]. Any other distribution requires a more general or approximate method, and in this case, we use a particle filter [34][2]. Note that it is important to correctly estimate distance uncertainty, as the subsequent transformations are highly non-linear. A Taylor series expansion (not shown here) confirms that the variance of distance transforms into a bias in the expected rate. Thus, by measuring and reducing distance uncertainty, we obtain more accurate estimates of the expected rates.

Having said that, we also use double exponential smoothing (ES) with linear trend [35] as the third method of inference, serving as a baseline. ES[3], being a Holt linear model [35], estimates ρ as $\rho_t = l_{t-1} + \tau s_{t-1}$, where the state (decomposed into trend s and level l) is estimated with simple filtration [35]:

$$\begin{cases} l_{t+1} = & \alpha \rho_t^{RTT} + (1-\alpha)(l_t + v_t), \\ s_{t+1} = & \beta(l_{t+1} - l_t) + (1-\beta) s_t, \end{cases} \tag{6}$$

where $\alpha, \beta \in (0, 1)$ are the model parameters. This approach is simple and fast; however, it requires a constant sampling rate.

Having introduced several inference methods, we now discuss the advantages and disadvantages of using them. *Exponential smoothing* is extremely simple and lightweight, but has several strict requirements: a constant sampling rate (non-uniform sampling is not uniquely defined), the probabilistic interpretation has a single source of error (which correlates measurement and movement noises without any physical reason), and finally, the method adds α and β as two additional hyperparameters to configure. *Kalman filter* is more sophisticated, but still rather lightweight. KF returns the result as a distribution that allows, e.g., adaptive measurements, and has an analytical solution for posterior inference [33]. Its underlying assumptions include a linear Gaussian model of the system and a Gaussian distribution of error. Additionally, KF requires *a priori* knowledge of sensor noise. *Particle filter* can model any distribution of error (e.g., exponential Gaussian) and also returns the result as a distribution. However, the distribution is approximated by discrete samples (particles). This makes PF computationally heavy (depending on the number of particles): more particles result in a better approximation, yet a higher computational demand. Having said that, this method is accelerator friendly, and we observed improved performance on graphics processing units.

	Exponential smoothing	**Kalman filter**	**Particle filter**
Low computational complexity	+	+	−
Easily tunable	−	−	+
Probabilistic interpretation	+/−	+	+
Analytical solution	+/−	+	−
Extensible	−	−	+

Table 5: Comparison of the inference variants

In summary, we study FTMRate in three inference variants: Kalman filter, particle filter, and exponential smoothing. Such inference allows us to estimate the state of the system at the measurement time which then allows for continuous extrapolation and evaluation at a random time. All these methods remain conceptually simple and should be possible to implement even on embedded devices with low computing power. Table 5 compares the key features of the approaches we use.

4.2.3. Rate Selection

In the following, we explain in detail how rates are selected in our scheme. Formally, the problem we are dealing with is a Markov decision process because the dynamics of (3) form a Markov process. However, since the state transition is independent of the action taken by the agent (i.e., the chosen MCS), in our proposal, the MCS is selected according to

$$\mu^*(\rho) = \arg\max_{\mu} \mathsf{E}_{\gamma \sim p_{\boldsymbol{\theta}_c}(\gamma|\rho)} \lambda_\mu^{\text{tx}} s_{\boldsymbol{\theta}_s}(\gamma, \mu), \tag{7}$$

where λ_μ^{tx} is the transmission rate for the μ-th MCS. This selection process includes multiple steps detailed below.

First, we measure and smooth the distance. Then we map the distance ρ_t to RSS γ_t using the following log-distance channel model:

$$\gamma(\rho) = \gamma_0 - (L_0 + 10E\log_{10}(\rho)) + \Delta, \tag{8}$$

where E is the path loss exponent, γ_0 – the reference RSS, L_0 – a reference path loss measured at a distance of 1 m,

[2] This is also required for 2D dynamics, as stated in Section 7.

[3] Note that in the special case of random walk plus noise this is equivalent to a Kalman filter [36].

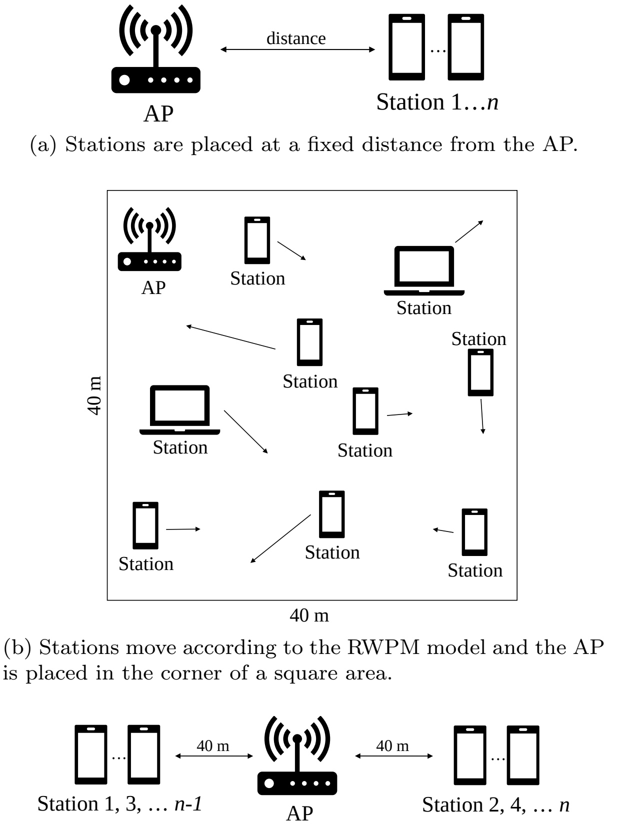
and Δ – the difference between the default and currently used transmission power. Therefore, the parameters of the channel model used are $\boldsymbol{\theta}_c = (\gamma_0, L_0, E) \in \mathbb{R}^2 \times \mathbb{R}_+$. The Δ variable allows to instantly adjust to transmission power changes. Next, we compute the probability of successful transmission ξ modeled with a CDF of the *sinh-arcsinh* normal distribution [37] whose four parameters $\boldsymbol{\theta}_s \in \mathbb{R}^2 \times {\mathbb{R}_+}^2$ are estimated from simulations. Then, we scale the rate at each MCS (λ_μ^{tx}) by the successful transmission probability ξ to obtain the expected rate. Finally, we choose the MCS value that maximizes the expected value of the data rate distribution.

Each of these steps could be considered a trainable parametric model, so it introduces an error. Therefore, we represent the intermediate values as random variables in the graphical model (Fig. 4). Nevertheless, we observe that the errors are small and can be omitted in the first approximation. Since all transformations are bijective, so is their composition, as shown in [2]. Thus, the approximation mentioned above allows us to directly map the distribution $p(\rho_t|\rho_{t<})$ to the rate distribution at each MCS. The main benefit is to map the distance uncertainty ρ to the uncertainty of the selected rate. Since this transformation is highly non-linear, the variance of distance measurements shifts the expected rate. This may cause a suboptimal MCS to be selected when operating close to the crossing point of two curves. The expectation in (7) is computed after non-linearities as an average of multiple samples because, to the best of our knowledge, this non-linear transform has no known analytical result for expectation.

4.3. Hybrid FTMRate Overview

To address the challenge of obstacles such as walls that can impact FTM-based rate selection, we propose Hybrid FTMRate (Fig. 2). We introduce a latent binary variable z_t which has a value $z_t = 1$ if we are operating in a LOS environment and FTMRate gives the correct rates. Otherwise, for the NLOS case, we have $z_t = 0$, FTMRate is biased and a classical agent (e.g., probing-based) is expected to perform better.

The introduction of the latent variable brings a hierarchy to the problem, because now the probability of a successful transition is a mixture of distributions conditioned on z_t. Since z is hidden, we cannot marginalize it, and thus the probability distribution must be learned. As this is a binary variable, an MAB agent can be used for this task. In the proposed hybrid approach, the MAB agent chooses the rate selection algorithm, either FTMRate or a classical one. Once the algorithm is selected, it chooses the rate and the information about the transmission outcome (reward) follows up through all agents up to the MAB agent, providing online learning. With the MAB's ability to explore, the agent can quickly adapt to the environment. As MABs are efficient and in the proposed solution there are only two arms, the computational overhead of this method is minimal.



(a) Stations are placed at a fixed distance from the AP.

(b) Stations move according to the RWPM model and the AP is placed in the corner of a square area.

(c) Stations are placed in hidden clusters, 40 m from the AP.

Figure 5: Topologies of the (a) equal distance, (b) mobile stations, and (c) hidden stations scenarios.

5. Performance Evaluation

We conduct a performance evaluation of first FTMRate and then Hybrid FTMRate. For the simulation environment, we use the ns-3.36.1[4] network simulator. We compare our proposal in its three inference variants (i.e., exponential smoothing, Kalman filter, and particle filter) with the following baseline rate selection protocols:

- Minstrel – the default rate selection method in Linux systems, also available in ns-3,
- Thompson sampling (TS) – an ML-based rate selection method available in ns-3 [16],
- oracle – a hypothetical rate selection method that uses perfect location knowledge to calculate RSS using (8) and selects MCS based on this knowledge. It does not consider multi-path fading effects, so it is an approximation of the upper bound.

The FTMRate algorithm is implemented in Python, using the JAX library [38]. It is connected to ns-3 using the ns3-ai interface [39]. We provide the complete code used in this study with detailed instructions as open source[5].

[4] https://www.nsnam.org/

[5] https://github.com/ml4wifi-devs/ftmrate

Parameter	**Value**
Band	5 GHz
PHY/MAC	IEEE 802.11ax
Channel width	20 MHz
Spatial streams	1, SISO
Guard interval	3200 ns
Frame aggregation	A-MPDU aggregation
Loss model	Log-distance w/ Nakagami fading
Path loss exponent E	3
Reference RSS γ_0	$\approx 110\,\text{dB}$
Reference path loss L_0	$\approx 46.6\,\text{dB}$
Per-station traffic load	Uplink, 125 Mb/s UDP
Packet size	1500 B

Table 6: General simulation parameter settings

We consider both static and mobile scenarios with full-buffer stations transmitting to a single AP (which under our network configuration settings amounts to 125 Mb/s per-station offered load). We also assume no outside interference, i.e., no overlapping Wi-Fi networks. The general simulation settings are given in Table 6, while the scenario-specific settings are given in Table 7. To decrease simulation time, we use only 20 MHz channels (in the absence of overlapping networks, the results are qualitatively identical to using wider channel widths) and single spatial stream (SISO) transmissions. In each figure, we provide 99% confidence intervals for the mean of the approximate throughput, represented as bands around the data points, unless stated otherwise.

With respect to [2], we repeat only the basic scenario (Section 5.1) to illustrate how FTMRate's performance differs from classical rate selection algorithms. Then, we analyze the impact of dynamic power settings (Section 5.2), hidden stations (Section 5.3), and signaling overhead (Section 5.3). Next, we analyze the performance of Hybrid FTMRate (Section 5.5). Meanwhile, we refer the reader to [2] for results related to mobility where we show that (a) the selected rate of performing ranging measurements (once every 0.5 s) is enough for FTMRate to provide satisfactory performance in low-mobility scenarios and (b) FTMRate outperforms two baseline algorithms in scenarios with random station mobility.

5.1. Equal Distance

We begin with a scenario in which the stations are immobile and placed at a given distance from the AP, identical for each station (Fig. 5a). We analyze two distances $\rho = 1\,\text{m}$ and $\rho = 20\,\text{m}$. The former setting involves stations located close to the AP. This leads to excellent channel conditions, but collisions can still happen. To address this, stations should always opt for the highest MCS and avoid confusing collisions with weak channel conditions. In the latter setting, both collisions and channel errors caused by weak signals happen, although there are no hidden stations involved. At this distance, according to (7), an MCS value of 7 is optimal.

The results shown in Fig. 6 confirm the well-known effect that the aggregated network throughput decreases with the number of transmitting stations. Additionally, both Minstrel and TS suffer from the impact of collisions on the probability of successful transmissions. This is because they cannot discriminate between frame loss due to weak signal (poor channel conditions) and a collision. Therefore, they treat every packet loss equally and reduce the MCS, which is, however, incorrect in case of collisions, since lower MCS will not decrease the contention level. Our approach is resilient to this problem because it does not consider whether transmissions are successful or not, but only considers FTM measurements. Therefore, FTMRate, in all its inference variants, can reach the approximate upper bound (the oracle).

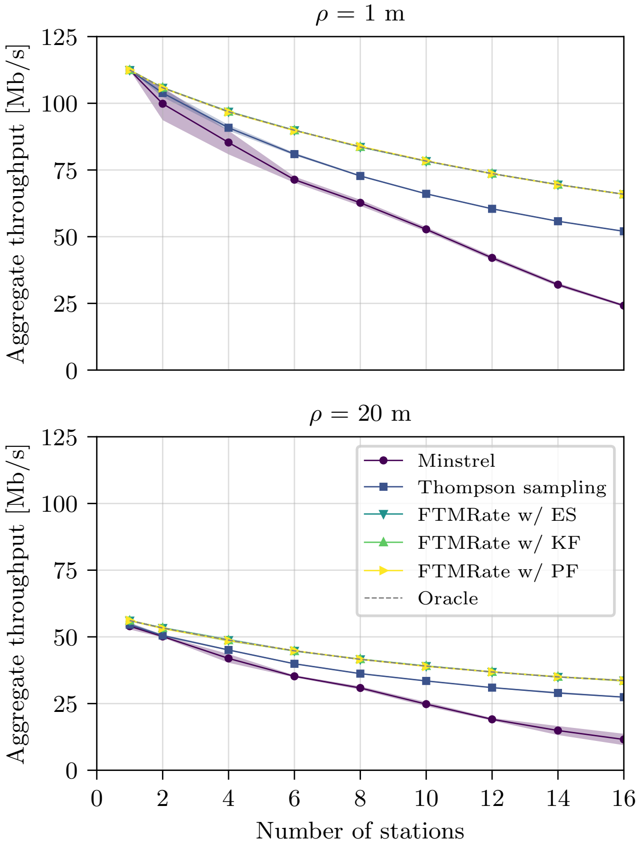


Figure 6: Aggregate network throughput in the equal distance scenario (Fig. 5a): $\rho = 1\,\text{m}$ (top) and $\rho = 20\,\text{m}$ (bottom). Note that all three FTMRate curves overlap.

5.2. Dynamic Power Settings

In previous scenarios, we assumed fixed transmission power settings. However, FTMRate also performs well when the transmission power changes. To illustrate this, we analyze a scenario where a single station transmits frames to the AP while the transmission power randomly alternates between two levels, differing by $\Delta = 5\,\text{dB}$ or $\Delta = 15\,\text{dB}$. The power switches follow an exponential distribution with a mean period of constant power $1/\lambda = 4\,\text{s}$ or $1/\lambda = 8\,\text{s}$. We selected the above parameters only for illustrative purposes: the power levels reflect a clear

Scenario	**Equal distance**	**Dynamic power**	**Mobile stations (RWPM)**	**Hidden stations**
Stations	{1, 2, 4, ..., 16}	1	10	{2, 4, 6, ..., 16}
Distance from AP	Fixed distance	Fixed distance	Random walk in a square area (40 m × 40 m)	Fixed distance
Velocity	0 m/s	0 m/s	0 − 1.4 m/s with 0 − 20 s pause	0 m/s
Start position	{1, 20} m	5 m	Random	40 m
Simulation time	No. of stations $\times 10 + 50$ s	55 s	1000 s	No. of stations $\times 10 + 50$ s
FTM signaling	Out-of-band or in-band	Out-of-band	In-band	Out-of-band

Table 7: Simulation parameter settings for each scenario

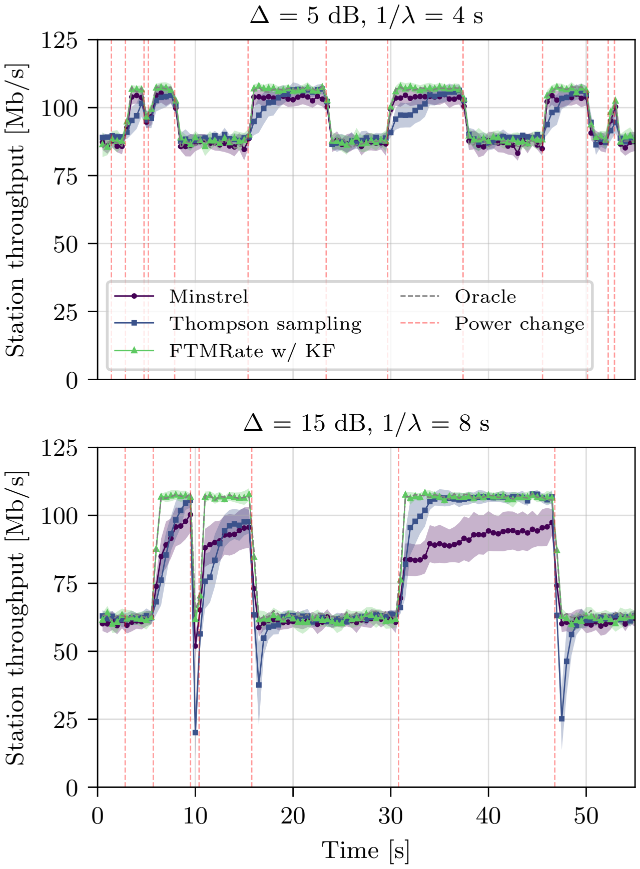


Figure 7: Throughput in the dynamic power scenario with a single station transmitting to the AP. The power level difference $\Delta = 5\,\text{dB}$, mean constant power period $1/\lambda = 4\,\text{s}$ (top) and $\Delta = 15\,\text{dB}$, $1/\lambda = 8\,\text{s}$ (bottom). Only one FTMRate variant is presented as all three exhibit similar performance.

data rate change of several MCS values and λ to show the convergence properties of the algorithms. Indeed, this scenario is inspired by the study in [16]. The obtained results (Fig. 7) clearly indicate that our method adapts rapidly, outperforming both baseline rate selection algorithms, across various power level differences and switching frequencies. This is due to the indirect estimation of the RSS value in (8), which allows to immediately change the transmission power and thus adjust MCS accordingly.

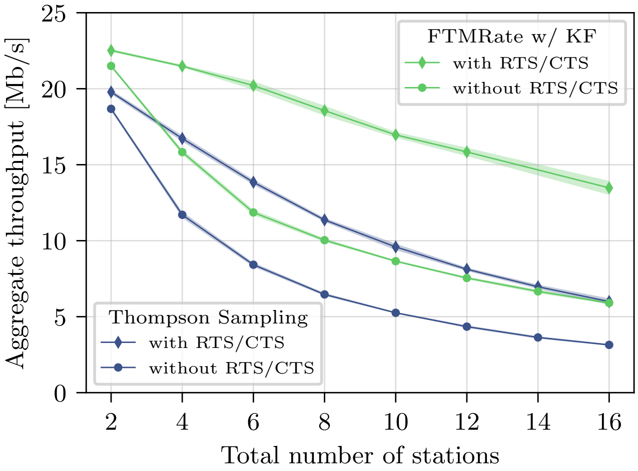


Figure 8: Aggregate network throughput in the hidden station scenario with stations grouped in hidden clusters (Fig. 5c), with $\rho = 40\,\text{m}$.

5.3. Hidden Stations

Since FTMRate is not affected by collisions, it should perform well in scenarios with hidden stations. We set up station clusters on opposite sides of an AP at a distance of $\rho = 40\,\text{m}$ (Fig. 5c). In this analysis, we compare performance for both RTS/CTS enabled and disabled, but restrict the rate managers to TS and FTMRate with KF. The former serves as the baseline due to its previous good performance, while all FTMRate's variants perform similarly so we choose the one which balances complexity with extensibility (Table 5).

The results in Fig. 8 clearly show that FTMRate outperforms TS. Even with RTS/CTS disabled, FTMRate's performance matches TS with RTS/CTS. This is a direct result of FTMRate's design (transmission rates are not downgraded on account of collisions).

5.4. Signaling Overhead (In-Band FTM)

In this section we measure the signaling overhead of both the proposed FTMRate algorithm and the FTM procedure itself, by comparing their operation with the oracle as well as with other rate selection algorithms.

We conduct tests with in-band measurements, i.e., when both ranging and data communication are done over the

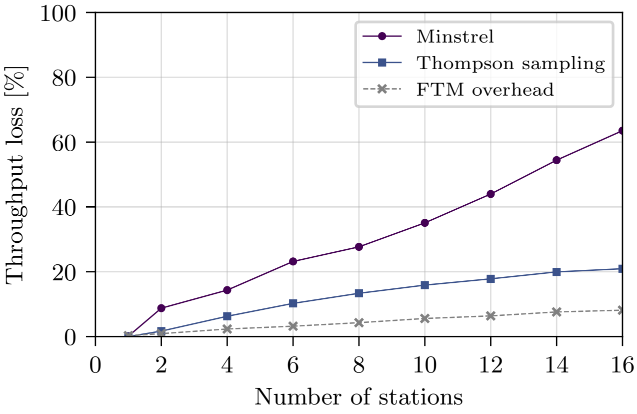


Figure 9: Evaluation of FTM overhead in the equal distance scenario (for $\rho = 1\,\mathrm{m}$): throughput loss caused by switching to in-band FTM measurements or by switching to another rate selection manager.

same channel. This evaluation is carried out in two scenarios: *equal distance* (for the FTM procedure evaluation) and *mobile stations* (for the FTMRate algorithm evaluation), with settings as in Table 7. Both scenarios have a large number of stations where the impact of overhead is most significant. For this in-band analysis, we use FTM-ns3 [40] (which extends ns-3.35 with the FTM procedure and its signaling overhead) and not the Python-based out-of-band FTM implementation used in the previous subsections. We use the default configuration of FTM for the 20 MHz channel. FTM measurements are performed, independently for each station, every 0.5 seconds and if the measurement fails, it is immediately repeated until it is successful.

We evaluate FTM overhead for the equal distance scenario (Fig. 9) by calculating the *throughput loss*, defined as the difference in achieved throughput between the oracle manager and an oracle manager that performs FTM measurements and selects MCS knowing the exact distance. For Minstrel and TS, the throughput loss is defined as the difference between the throughput achieved with the oracle and when switching to the respective manager. The FTM overhead is the smallest in comparison to other algorithms, which suggests that this advantage can be exploited by the FTMRate algorithm.

In the mobile stations scenario we evaluate FTMRate overhead. The obtained results (Fig. 10) are pnly marginally inferior to those obtained through out-of-band measurements [2]. However, FTMRate still outperforms Minstrel and is no worse than TS.

5.5. Hybrid FTMRate Performance

We implement a Hybrid FTMRate agent in which a TS agent is responsible for selecting the appropriate manager (as the MAB in Fig. 2). This top-level TS agent has two arms, corresponding to the selection of FTMRate and TS rate selection managers, respectively. Depending on its decision, either FTMRate or TS rate selection manager is responsible for selecting MCS for data transmission.

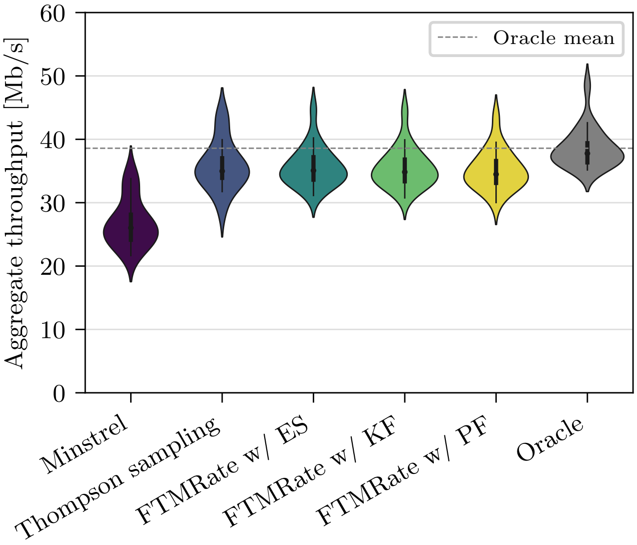


Figure 10: Aggregate network throughput with mobile stations deployed randomly around the AP (Fig. 5b) when FTMRate operates in-band.

Fig. 11 presents a case in which the station moves away from the AP with a speed of 0.5 m/s and every 5 m encounters a point where the signal strength drops by 3 dB (which represents a wall). In this scenario, the performance of the basic FTMRate suffers from NLOS, which causes the success probability estimate to be inflated. The hybrid approach enables intelligent detection of this case and switching to MCS selection through a classical manager. We can expect that Hybrid FTMRate will be no better than TS and indeed observe that this is the case: Hybrid FTMRate performs equal to or only slightly worse than TS, ensuring high performance throughout the run.

The second scenario, presented in Fig. 12, is analogous to the case discussed in Section 5.1 for $\rho = 1\,\mathrm{m}$. The results confirm that even with an additional MAB agent, which introduces a small overhead for exploration and continuous testing of the condition of both agents (FTMRate and TS), the performance of hybrid FTMRate is superior to other probing-based solutions.

The presented cases demonstrate that hybrid FTMRate is not only efficient in dense scenarios with multiple collisions, but can also be a versatile data rate manager.

6. Experimental Validation

Having performed extensive simulation studies, we now conduct an experimental validation which focuses on the performance of FTMRate. We assess whether FTMRate can select rates correctly in real-world conditions. Below, we first describe the testbed and necessary calibration steps. Then, we discuss results for two experimental scenarios: static and mobile.

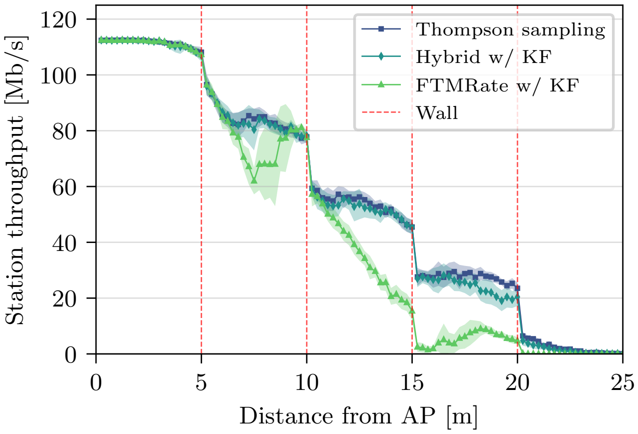


Figure 11: Station throughput in a scenario with a moving station, walls, and a hybrid FTMRate agent.

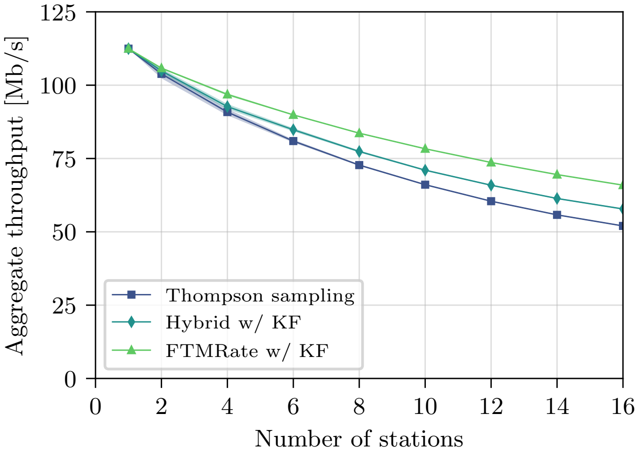


Figure 12: Aggregate network throughput in the equal distance scenario ($\rho = 1\,\text{m}$) with a hybrid FTMRate agent.

6.1. Testbed

We set up a testbed composed of an access point and a single client, both with Intel Joule 570x compute modules and Intel Dual Band Wireless-AC 8260 Wi-Fi adapters. They run Ubuntu 16.04 servers with default kernels, `iwlwifi` backports, and FTM enabled [41]. Additionally, the AP runs `hostapd`. Table 8 provides the parameter settings used in the experiments.

We use `iwl-mvm-rs` as the baseline rate selection algorithm, which is the rate selection algorithm of the Intel Linux wireless driver (`iwlwifi`) [42]. For FTMRate, we select the KF variant because it balances complexity and extensibility.

The measurements were taken in a large sports hall ($17 \times 54 \times 10.5$ m), with little interference from neighboring networks. The station used in our testbed is presented in Fig. 13. Attenuators, visible in the figure, are used to limit the maximum transmission range of the station. This allows us to observe more transmission rate changes over the same distance compared to a setting without attenuators.

Parameter	Value
Antenna gain	2 dBi
Attenuator loss (total)	35 dB
Band	2.4 GHz
PHY/MAC	IEEE 802.11n
Channel width	20 MHz
Spatial streams	1 (SISO), 2 (MIMO)
Guard interval	800 ns
Frame aggregation	A-MPDU aggregation
Path loss exponent E	≈ 1.31
Reference RSS	$\approx 67.82\,\text{dBm}$
Per-station traffic load	Uplink, $\approx$ 11.4 packets per second, UDP
Packet size	1000 B

Table 8: General experiment parameter settings

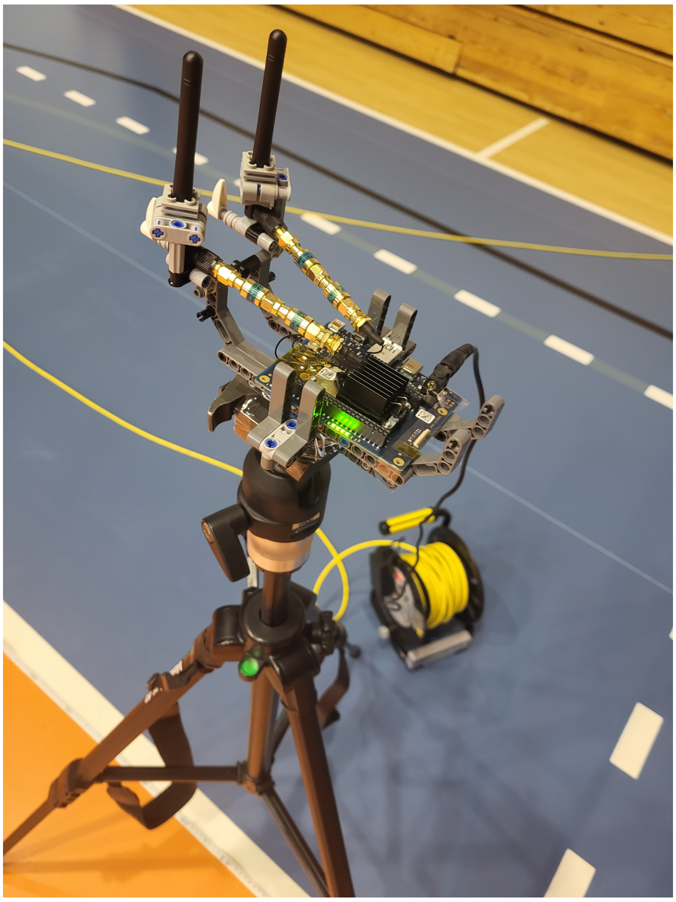


Figure 13: An Intel Joule 570x board serving as a station in our testbed. The attenuators used to limit the range are clearly visible. A similar board (without attenuators) served as the AP.

6.2. FTMRate Calibration

Calibration is necessary before using FTMRate in a new channel environment. The channel model, given in (8), which consists of the path loss exponent E, the reference RSS γ_0 and the reference path loss L_0 parameters is fitted to the data (obtained by sending frames from the station and measuring RSS at the AP, cf. Fig. 14). This task reduces to exponential curve fitting, where γ_0, L_0, and Δ are treated as a single shift parameter which is estimated in addition to the exponent parameter E.

The second step is necessary to estimate the transmission success probability for each MCS value. This probability function is modeled by the CDF of a *sinh-arcsinh* normal distribution which consists of four parameters, well described in the literature [37]. For each MCS, these parameters can be obtained by first measuring the ratio of successful frame transmissions (for each measured distance) and then by fitting the CDF curve. It may help to restrict the model by substituting the *sinh-arcsinh* CDF with a simpler normal CDF.

The Kalman filter method necessitates the estimation of an additional parameter, the sensor noise, represented by the standard deviation of a normal distribution that models measurement errors. This parameter can be predetermined *a priori*; however, we determine it by identifying the normal distribution that minimizes the Kullback–Leibler divergence which in this case reduces to the mean of the squared differences between the corrected FTM distances and uncorrected ones.

6.3. Static Scenario

We first evaluate the performance of FTMRate in a basic static configuration, where a stationary station is positioned at a distance of 2 m from the AP (Fig. 5a). This test allows us to assess the convergence time of the algorithm and validate the operation of our solution in a scenario with few collisions. The data is gathered from 10 independent repetitions, each lasting 25 s, and the approximate throughput is based on arrival times and the data rate of received packets.

The results presented in Fig. 15 indicate that FTMRate immediately selects a high MCS value, in contrast to the default manager, which takes approximately 5 s to find the optimal value. This scenario highlights the advantage of a closed-loop solution over a probing-based approach, which is the close-to-zero time convergence to the appropriate MCS value.

6.4. Dynamic Scenario

We also evaluate the performance of FTMRate in a dynamic scenario, in which a team member held a tripod with a station mounted on top (Fig. 13) and walked away from a stationary AP at an approximately constant speed for 10 s, traveling from 2 m to 12 m away from the AP. The beginning of the movement was triggered by an audio signal to synchronize with the start of the measurements. We repeated the experiment 15 times.

The measured data rate is per-frame and is estimated as the theoretical value based on the frames received by the AP (Fig. 16). Due to hardware deficiencies, the system has limitations which preclude a full saturation throughput experiment. We present these per-frame data rate results as an initial performance indicator. The results indicate that FTMRate outperforms `iwl-mvm-rs`, especially at the beginning of the experiment, when the station is still close to the AP. In the latter part of the experiment, FTMRate selects a single (optimal) value of MCS, hence the lack of deviation. The baseline manager, which uses a probing-based approach, fails to select the highest MCS values, whereas FTMRate, thanks to its closed-loop solution, selects them properly.

7. Challenges

In this section, we discuss two key challenges of the core FTMRate proposal: NLOS operation and the linear dynamics of the station.

7.1. Alternative Approaches for NLOS

We have shown that NLOS conditions are detrimental to FTMRate performance. Our proposal, Hybrid FTMRate, can alleviate these issues, although not perfectly.

Operation in NLOS conditions may alternatively be addressed by knowledge transfer from the FTMRate agent to a classic agent. For example, TS maintains a simple model of the success probability for each MCS in the form of a collection of beta distributions. To this end, we also maintain the success probability distributions in the form ξ, so it is easy to add prior knowledge from FTMRate to the TS agent. The parameters of the beta distribution can be set to minimize the divergence to ξ. In particular, the Kulbak-Leibler divergence can be chosen and minimized as follows. First, sample from ξ, and fit the beta distribution to the samples. This procedure is a single-shot bootstrap of TS and allows knowledge transfer between agents. Although promising, this may still pose some issues, as the TS agent cannot exploit the dynamics of the system.

Other approaches can also be considered such as data fusion (i.e., relying not only on range measurements) [43] or detection of LOS paths before the actual distance measurement [44].

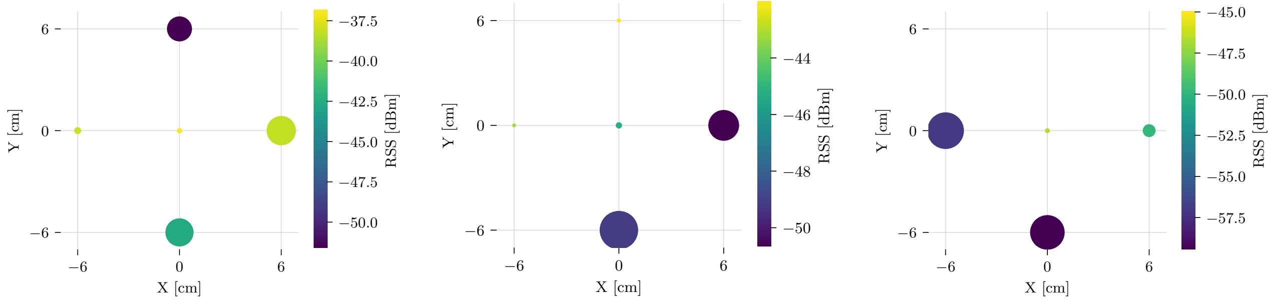
7.2. Exact Dynamics in 2D

The dynamics of (3) is an approximation, neglecting an additional degree of freedom in the circular position for a given distance. In this section, we propose an exact probabilistic model of the dynamics.

The extension is quite simple and follows (3), with one exception. The dynamic is duplicated for two orthogonal planar dimensions and the measurement gives a noisy distance. Mathematically, such a model is described as follows:

$$\begin{cases} \mathrm{d}\boldsymbol{\nu} &= \boldsymbol{\sigma}_\nu \mathrm{d}\boldsymbol{W}_1 \\ \mathrm{d}\boldsymbol{x} &= \boldsymbol{\nu}\mathrm{d}t + \boldsymbol{\sigma}_\rho \mathrm{d}\boldsymbol{W}_2, \end{cases} \tag{9}$$

where $\boldsymbol{\sigma}_\nu$ and $\boldsymbol{\sigma}_\rho$ are diagonal matrices and $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^2$ are two dimensional Wiener processes. The aforementioned measurement is $\rho_t^{RTT} = |\boldsymbol{x}_t| + \epsilon_t$, where $|\boldsymbol{x}_t| = \sqrt{x_{t,1}^2 + x_{t,2}^2}$.



(a) $d = 2\,\mathrm{m}$

(b) $d = 4\,\mathrm{m}$

(c) $d = 8\,\mathrm{m}$

Figure 14: Examples of RSS values measured at the AP during the experiments. The five points indicate station placement during transmission: the central point is at the designated distance from the AP while the four other points are offset by approximately half a wavelength (6 cm). The area of the circles denotes the variance of the results. The high dispersion of results is caused by multipath propagation.

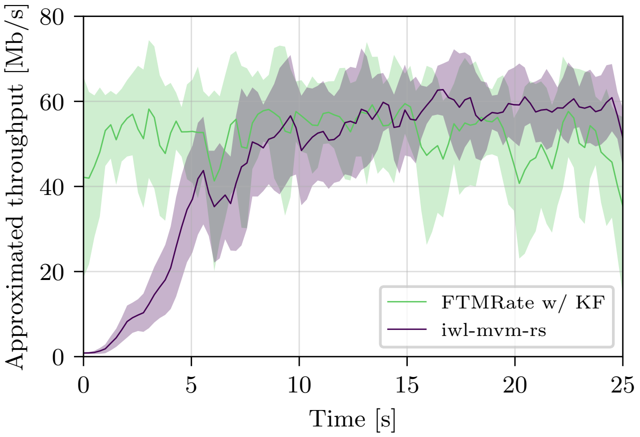


Figure 15: Approximate throughput in a static scenario using real hardware.

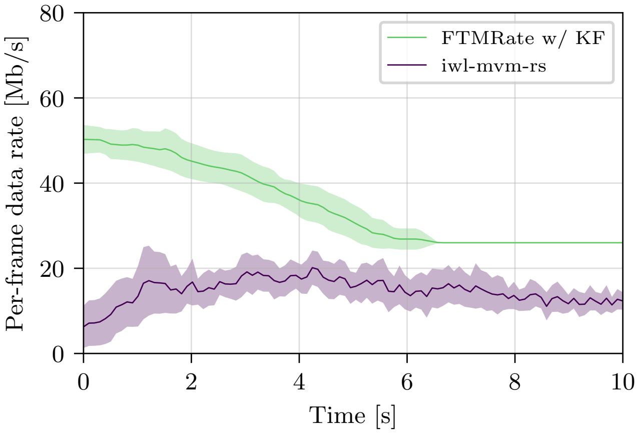


Figure 16: The average per-frame data rates observed at the receiver in a dynamic scenario (station moving away from the AP) using real hardware.

Equation (9) contains two independent dynamics (3) for each dimension. After discretization, a transition noise distribution is derived for both dimensions exactly as in (4).

The main reason we opt for approximate dynamics in throughout the paper, rather than the above-mentioned 2D dynamics, is that (9) is nonlinear, or can be made linear at the cost of losing normal distributions, thus efficient Kalman filtering cannot be used here. Still, for completeness, we propose here an exact model that can be used with a particle filter.

8. Discussion and Impact

We see from all the presented results that in LOS scenarios FTMRate has two main features which provide an advantage over existing solutions. First, it is collision-immune, which is beneficial when collision rates are high, e.g., on account of hidden stations or simply increased station density. The benefits of FTMRate are visible even for a low number of stations (Fig. 6). The additional use of RTS/CTS only increases the gain in hidden station scenarios (Fig. 8). We are also confident that these gains can be found in cross-technology coexistence scenarios, where sporadic transmissions in unlicensed bands by Bluetooth, ZigBee, or other devices may result in collisions but not lead, in the case of FTMRate, to a rate decrease.

Second, FTMRate has a lower convergence time compared to the dominant probing-based approach. Since most communication is bursty and sporadic in nature, user applications can benefit from the low convergence time. Additionally, network scenarios with a high turnover rate can also benefit. Consider a massive Internet of Things deployment, where sensors send traffic infrequently, with not enough frames for a probing-based method to converge. Another scenario is a subway train moving between hotspots located at stations – as the train arrives, all its passengers want to transmit simultaneously (using a probing-based approach would waste resources).

However, FTMRate also has some drawbacks. First, it requires signaling overhead, but we have shown that this does not offset the gains, even when done in-band (Fig. 9). The multi-band operation of future Wi-Fi devices will facilitate moving signaling (including FTM measurements) to other bands. Second, FTMRate requires adjustment to NLOS scenarios. We propose Hybrid FTMRate in Section 4.3 which reverts to classic algorithms in NLOS scenarios and we outline other approaches in Section 7. The final drawback of FTMRate is that it requires training. The FTM procedure itself requires establishing an offset, but this can be done automatically [44]. Meanwhile, the channel model parameters can either be estimated based on knowledge of the environment type or estimated online (which we leave as future work). Finally, the accuracy of FTMRate can be influenced by the delay (or loss) of FTM frames, which can be countered by optimizing the FTM burst size (to perform repeated measurements), as well as sending FTM frames with the most robust MCS and with the highest priority.

9. Conclusions

FTMRate is a new data rate selection algorithm for IEEE 802.11 networks. Its novelty lies in the use of FTM-based distance measurements and the application of statistical learning to these measurements to (a) estimate the distance from the AP, (b) estimate channel quality (RSS), and (c) map RSS to MCS (and thus the transmission rate). Through simulations, we have shown that our collision-immune design leads to the following desirable features:

(a) FTMRate performs better in dense scenarios in comparison to standard rate selection approaches (Section 5.1).

(b) Changing transmission power levels does not interrupt the performance of FTMRate (Section 5.2).

(c) FTMRate provides superior performance in hidden station scenarios (Section 5.3).

(d) FTMRate's high throughput results are not offset by in-band FTM signaling and remain no worse than that of probing-based managers which unnecessarily decrease rates under high contention (Section 5.4).

(e) Hybrid FTMRate can be part of a versatile data rate manager for NLOS and mixed LOS/NLOS scenarios (Section 5.5).

Finally, our experimental analysis confirms that FTMRate works in commercial-off-the-shelf devices, is compatible with MIMO, exhibits rapid convergence times (Section 6.3), and is also capable of selecting the correct data rates in dynamic scenarios (Section 6.4).

We note that by observing any random variable on the *rate inference* path in Fig. 4, we can transform the problem into channel inference. Therefore, as future work, we consider investigating methods to estimate channel parameters as well as optimize measurement frequency and study cases where interference from far-away transmissions should lead to lowering the MCS. Furthermore, we want to design a dedicated retransmission chain for FTMRate and find scenarios where our rapidly converging, probing-free approach can be particularly beneficial, such as in millimeter wave (mmWave) bands, which are inherently LOS.

Acknowledgments

This research was funded by the National Science Centre, Poland (DEC-2020/39/I/ST7/01457) and by the German Research Foundation (DFG DR 639/28-1). We gratefully acknowledge Polish high-performance computing infrastructure PLGrid (HPC Centers: ACK Cyfronet AGH) for providing computer facilities and support within computational grant no. PLG/2022/015838. For the purpose of Open Access, the authors have applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission. The authors thank Piotr Biel and the staff of the Faculty of Physical Education and Sport of the AGH University of Krakow for their support in conducting the experimental validation.

References

[1] S. Szott, K. Kosek-Szott, P. Gawłowicz, J. Torres Gómez, B. Bellalta, A. Zubow, F. Dressler, Wi-Fi Meets ML: A Survey on Improving IEEE 802.11 Performance with Machine Learning, IEEE Communications Surveys & Tutorials 24 (2022) 1843–1893. doi:10.1109/COMST.2022.3179242.

[2] W. Ciezobka, M. Wojnar, K. Kosek-Szott, S. Szott, K. Rusek, FTMRate: Collision-Immune Distance-based Data Rate Selection for IEEE 802.11 Networks, in: 24th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2023), IEEE, Boston, MA, 2023. doi:10.1109/wowmom57956.2023.00039.

[3] IEEE, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Std 802.11-2016, IEEE, 2016. doi:10.1109/IEEESTD.2016.7786995.

[4] C. Wang, Dynamic ARF for throughput improvement in 802.11 WLAN via a machine-learning approach, Elsevier Journal of Network and Computer Applications 36 (2013) 667–676. doi:10.1016/j.jnca.2012.12.025.

[5] O. Puñal, H. Zhang, J. Gross, RFRA: Random Forests Rate Adaptation for Vehicular Networks, in: A World of Wireless, Mobile and Multimedia Networks (WoWMoM), IEEE, 2013, pp. 1–10. doi:10.1109/WoWMoM.2013.6583398.

[6] E. Kurniawan, P. H. Tan, S. Sun, Y. Wang, Machine learning-based channel-type identification for IEEE 802.11ac link adaptation, in: 24th Asia-Pacific Conference on Communications (APCC), IEEE, 2018.

[7] C.-Y. Li, S.-C. Chen, C.-T. Kuo, C.-H. Chiu, Practical Machine Learning-Based Rate Adaptation Solution for Wi-Fi NICs: IEEE 802.11ac as a Case Study, IEEE Transactions on Vehicular Technology 69 (2020) 10264–10277. doi:10.1109/TVT.2020.3004471.

[8] S. Khastoo, T. Brecht, A. Abedi, Neura: Using neural networks to improve wifi rate adaptation, in: Proceedings of the 23rd International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Association for Computing Machinery, New York, NY, USA, 2020, p. 161–170. doi:10.1145/3416010.3423217.

[9] IEEE, IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, Std 802.11-2020, IEEE, 2020.

[10] MADWIFI, Minstrel documentation, 2005. URL: https://sourceforge.net/p/madwifi/.

[11] R. Grünblatt, I. Guérin-Lassous, O. Simonin, Study of the intel wifi rate adaptation algorithm, in: CoRes 2019-Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication, 2019, pp. 1–4.

[12] T. Joshi, D. Ahuja, D. Singh, D. P. Agrawal, SARA: Stochastic Automata Rate Adaptation for IEEE 802.11 Networks, IEEE Transactions on Parallel and Distributed Systems 19 (2008) 1579–1590. doi:10.1109/TPDS.2007.70814.

[13] R. Karmakar, S. Chattopadhyay, S. Chakraborty, Dynamic Link Adaptation in IEEE 802.11ac: A Distributed Learning Based Approach, in: IEEE 41st Conference on Local Computer Networks (LCN), IEEE, 2016, pp. 87–94. doi:10.1109/LCN.2016.20.

[14] R. Combes, J. Ok, A. Proutiere, D. Yun, Y. Yi, Optimal rate sampling in 802.11 systems: Theory, design, and implementation, IEEE Transactions on Mobile Computing 18 (2018) 1145–1158.

[15] H. Gupta, A. Eryilmaz, R. Srikant, Low-complexity, low-regret link rate selection in rapidly-varying wireless channels, in: IEEE Conference on Computer Communications, IEEE, 2018, pp. 540–548.

[16] A. Krotov, A. Kiryanov, E. Khorov, Rate Control With Spatial Reuse for Wi-Fi 6 Dense Deployments, IEEE Access 8 (2020) 168898–168909. doi:10.1109/ACCESS.2020.3023552.

[17] B. Radunovic, A. Proutiere, D. Gunawardena, P. Key, Dynamic channel, rate selection and scheduling for white spaces, in: Proceedings of the Seventh COnference on emerging Networking EXperiments and Technologies, 2011, pp. 1–12.

[18] R. Karmakar, S. Chattopadhyay, S. Chakraborty, IEEE 802.11ac Link Adaptation Under Mobility, in: 2017 IEEE 42nd Conference on Local Computer Networks (LCN), 2017, pp. 392–400. doi:10.1109/LCN.2017.90.

[19] R. Karmakar, S. Chattopadhyay, S. Chakraborty, A deep probabilistic control machinery for auto-configuration of wifi link parameters, IEEE Transactions on Wireless Communications 19 (2020) 8330–8340. doi:10.1109/TWC.2020.3021597.

[20] S. Cho, Reinforcement learning for rate adaptation in csma/ca wireless networks, in: J. J. Park, S. J. Fong, Y. Pan, Y. Sung (Eds.), Advances in Computer Science and Ubiquitous Computing, Springer Singapore, Singapore, 2021, pp. 175–181.

[21] V. Saxena, H. Tullberg, J. Jaldén, Reinforcement learning for efficient and tuning-free link adaptation, IEEE Transactions on Wireless Communications 21 (2021) 768–780.

[22] S.-C. Chen, C.-Y. Li, C.-H. Chiu, An Experience Driven Design for IEEE 802.11ac Rate Adaptation based on Reinforcement Learning, in: 40th IEEE International Conference on Computer Communications (INFOCOM 2021), IEEE, Virtual Conference, 2021. doi:10.1109/infocom42981.2021.9488876.

[23] K. Yano, K. Suzuki, B. S. Ojetunde, K. Yamamoto, A Study on Update Frequency of Q-Learning-based Transmission Datarate Adaptation using Redundant Check Information for IEEE 802.11ax Wireless LAN, in: ICAIIC, IEEE, 2022, pp. 345–350.

[24] A. Kamerman, L. Monteban, WaveLAN®-II: a high-performance wireless LAN for the unlicensed band, Bell Labs technical journal 2 (1997) 118–133.

[25] S. H. Wong, H. Yang, S. Lu, V. Bharghavan, Robust rate adaptation for 802.11 wireless networks, in: Proceedings of the 12th annual international conference on Mobile computing and networking, 2006, pp. 146–157.

[26] J. Chen, J. Ma, Y. He, G. Wu, Deployment-friendly link adaptation in wireless local-area network based on on-line reinforcement learning, IEEE Communications Letters (2023).

[27] G. Pocovi, A. A. Esswie, K. I. Pedersen, Channel quality feedback enhancements for accurate urllc link adaptation in 5g systems, in: 2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring), IEEE, 2020, pp. 1–6.

[28] W. R. Thompson, On the likelihood that one unknown probability exceeds another in view of the evidence of two samples, Biometrika 25 (1933) 285–294.

[29] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, Playing Atari with Deep Reinforcement Learning, cs.LG 1312.5602, arXiv, 2013. doi:10.48550/arXiv.1312.5602.

[30] M. Bullmann, T. Fetzer, F. Ebner, M. Ebner, F. Deinzer, M. Grzegorzek, Comparison of 2.4 GHz WiFi FTM- and RSSI-Based Indoor Positioning Methods in Realistic Scenarios, Sensors 20 (2020) 4515. doi:10.3390/s20164515.

[31] A. Zubow, C. Laskos, F. Dressler, FTM-ns3: WiFi Fine Time Measurements for NS3, in: 17th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS 2022), IEEE, Virtual Conference, 2022, pp. 1–7. doi:10.23919/WONS54113.2022.9764460.

[32] I. Karatzas, I. Shreve, S. Shreve, S. Shreve, Brownian Motion and Stochastic Calculus, Graduate Texts in Mathematics (113) (Book 113), Springer New York, 1991.

[33] K. Murphy, Machine Learning: A Probabilistic Perspective, Adaptive Computation and Machine Learning series, MIT Press, 2012.

[34] N. Chopin, O. Papaspiliopoulos, An Introduction to Sequential Monte Carlo, Springer Series in Statistics, Springer International Publishing, 2020.

[35] R. Hyndman, A. Koehler, J. Ord, R. Snyder, Forecasting with Exponential Smoothing: The State Space Approach, Springer Series in Statistics, Springer Berlin Heidelberg, 2008.

[36] A. C. Harvey, Forecasting, Structural Time Series Models and the Kalman Filter, Cambridge University Press, 1990. doi:10.1017/CBO9781107049994.

[37] C. Jones, A. Pewsey, The sinh-arcsinh normal distribution, Significance 16 (2019) 6–7.

[38] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs, 2018. URL: http://github.com/google/jax.

[39] H. Yin, P. Liu, K. Liu, L. Cao, L. Zhang, Y. Gao, X. Hei, Ns3-ai: Fostering artificial intelligence algorithms for networking research, in: Proceedings of the 2020 Workshop on Ns-3, ACM, 2020, p. 57–64.

[40] A. Zubow, C. Laskos, F. Dressler, Towards the Simulation of WiFi Fine Time Measurements in NS3 Network Simulator, Elsevier Computer Communications 210 (2023) 35–44. doi:10.1016/j.comcom.2023.07.028.

[41] Enable 802.11mc FTM on Intel 8260, https://github.com/HappyZ/iw_intel8260_localization/wiki/Enable-802.11mc-FTM-on-Intel-8260, ???? Accessed: 2023-05-25.

[42] Intel Linux Wireless Driver, https://github.com/spotify/linux/blob/master/drivers/net/wireless/iwlwifi/iwl-agn-rs.h, ???? Accessed: 2023-05-25.

[43] K. Jiokeng, G. Jakllari, A. Tchana, A.-L. Beylot, When ftm discovered music: Accurate wifi-based ranging in the presence of multipath, in: IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1857–1866.

[44] S. Aggarwal, R. K. Sheshadri, K. Sundaresan, D. Koutsonikolas, Is wifi 802.11mc fine time measurement ready for prime-time localization?, in: 28th ACM International Conference on Mobile Computing and Networking (MobiCom 2022), 16th ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and Characterization (WINTECH 2022), ACM, Sydney, Australia, 2022, pp. 1–8. doi:10.1145/3556564.3558234.