TKN Telecommunication Networks Group

Technical University Berlin

Telecommunication Networks Group

DARA: Estimating the Behavior of Data Rate Adaptation Algorithms in WLAN Hotspots

Sven Wiethölter, Andreas Ruttor*, Uwe Bergemann,
Manfred Opper*, and Adam Wolisz

* Artificial Intelligence Group (KI), TU Berlin, Germany

wiethoel@tkn.tu-berlin.de

Berlin, January 2013

TKN Technical Report TKN-13-001

TKN authors have conducted this work within
the CELTIC project MEVICO and have been supported by
the German Federal Ministry of Education and Research (BMBF 01BU1014).

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Notice

This technical report is the extended version of our paper that will appear in the proceedings of IEEE Infocom 2013 (April 14$^{\text{th}}$-19$^{\text{th}}$). The report contains more details regarding the approach, the parameter selection, and the "proof-of-concept" evaluation. In addition, we present one specific application example of our scheme.

January 3$^{\text{rd}}$, 2013 The Authors.

Abstract

Data rate adaptation (RA) schemes are the key means by which WLAN adapters adjust their operation to the variable quality of wireless channels. Adapters choose the most appropriate Modulation and Coding Scheme (MCS) for each packet transmission. The IEEE 802.11 standard does not specify any RA preferences allowing for competition in performance among vendors, thus numerous proprietary solutions coexist. While the RA schemes implemented in individual user terminals are unknown to the AP of a hotspot, it is well known that the way how individual stations adapt their rates strongly influences the performance of the whole WLAN cell. Consequently, this paper presents DARA, a novel approach to estimate the fundamental features of the RA schemes implemented in the individual stations. Upon very short observations of the station operation, we estimate the behavior of RA schemes. We present the Machine Learning approach used by DARA and demonstrate its efficiency using both simulated WLAN configurations as well as measurements from a WLAN setup in an on-campus office environment. While the knowledge about RAs being implemented in individual stations might be conceptually used for different traffic management purposes, we demonstrate one concrete, selected example: the selection of VoIP flows to be served in the hotspot.

Contents

1 Introduction 3

2 Related Work 5

3 DARA Principle 6

3.1 Basic Functionality 6

3.2 Motivation for the Selected Parameters 7

3.3 Considerations for Practical Usage 7

4 ML Model used by DARA 8

5 Scenarios for DARA's Evaluation 10

5.1 Selected Rate Adaptation Schemes 10

5.2 Setup for the simulation-based Training 11

5.2.1 Simulation Model 11

5.2.2 Traffic Model and QoS Constraints 11

5.2.3 Scenarios 12

5.2.4 Selection of Training Nodes 12

5.3 Setup for the testbed-based Training 12

5.3.1 Testbed Environment 12

5.3.2 Collection of Data 13

5.3.3 Placement of WLAN Nodes 13

5.3.4 Traffic Model 14

5.4 Training the ML Model 14

6 Evaluation of DARA's Accuracy 15

6.1 Accuracy Metric 15

6.2 Simulation Results 15

6.3 Results from the WLAN Testbed 16

6.4 Conclusions from the Estimation Results 16

7 Applying DARA for Technology Selection 21

7.1 System under Consideration 21

7.2 Schemes for the Selection of WLAN Flows 22

7.2.1 RSSI-based Decisions 22

7.2.2 DARA-based Selection (DARA-S) 22
7.3 Performance Evaluation 22
7.3.1 Mixes of RA Schemes 23
7.3.2 Comparison of the Selection Schemes 23
7.3.3 Accuracy Analysis of DARA-S 24

8 Conclusions 26

Chapter 1

Introduction

The popularity of Wireless LANs has been continuously increasing over the last years. Both, laptops and handhelds ranging from smart phones to tablets are nowadays equipped with WLAN network interface cards (NICs). As the mobile traffic further grows, offloading it to WLANs becomes more and more attractive [1]. In their recent paper, K. Lee et al. [2] argue that today's WLANs could take over 65percent of the total mobile traffic leading also to energy savings up to 55 percent as a result of WLAN's small transmission durations.

Similar to other wireless technologies, a key issue for successful WLAN transmissions is the Signal to Interference plus Noise Ratio (SINR). In reality, SINR fluctuates as a result of fading and interference, therefore modern wireless technologies usually incorporate schemes trying to adapt the transmission parameters to the SINR conditions. The IEEE 802.11 standard [3] specifies distinct sets of MCSs for each Physical Layer (PHY) specification (i.e., 802.11a/b/g/n). Determined by the applied PHY configuration, the transmitter is free to select a Modulation and Coding Scheme (MCS) on a per-packet basis for each transmission from its set of MCSs. The policy for the selection of MCSs resulting in different bit rates is usually referred to as the *Rate Adaptation (RA) Scheme*.

While specifying different MCSs, the IEEE 802.11 standard itself does not give strategies and mechanisms for the RA. As a result, individual WLAN NICs of multiple vendors may apply different RA algorithms. Vendors compete in usage of proprietary solutions aimed at achieving a leading performance. Thus, an Access Point (AP) in a WLAN hotspot serving stations (STAs) with NICs from different vendors is confronted with a heterogeneity of their behavior—without having any knowledge of the applied algorithms on the STA side.

For various purposes, it may be highly beneficial to have knowledge about the RA behavior of all the WLAN cards involved. Even a minor number of STAs transmitting at low-rates may harm the overall capacity of the WLAN cell. Medepalli et al. [4] showed that a single low rate 6 Mbps user is approximately equivalent to three 54 Mbps VoIP users in their setting. However, there exist no available mechanisms for the AP to inquire the applied RA scheme and its behavior from a WLAN NIC today. Further, it is even questionable whether such an inquiry would be practically feasible as hundreds of different devices from various vendors may make a standardization as well as an implementation of proper semantics and syntax describing individual RA schemes quite complicated.

This work presents a novel solution, denoted as Data Rate Estimation (DARA), which allows the AP to estimate the rate selection behavior of a STA by observing it on short time scales in the order of just some beacon intervals and making an "educated guess" afterwards. DARA does not assume knowledge of the internal operation of the RA scheme. To our best knowledge, this is the first solution offering such possibility.

The paper is structured as follows. After the related work in Chapter 2, Ch. 3 discusses the principle of DARA, before we present the Machine Learning (ML) model in Ch. 4. Ch. 5 describes the scenarios for DARA's evaluation leading to the results in Ch. 6. Ch. 7 presents the results of our application example in which we utilize DARA for a selection of flows to be served by a WLAN hotspot. We demonstrate that using the information about RA instead of the usual RSSI allows us to admit more flows while assuring the same QoS level of the individual flows. Ch. 8 concludes our work.

Chapter 2

Related Work

Comparisons of different, practical RA algorithms in the literature usually rely on schemes being implemented in the WLAN Linux driver Madwifi [5] which comes with the algorithms Minstrel, Onoe, SampleRate, and AMRR. Yin et al. [6] have shown the differences between the operational points of these algorithms by varying attenuation as well as interference durations in a wired setup. Ancillotti et al. [7] showed in their indoor measurements that the observed network throughput greatly varies due to differences in the behavior of AMRR, SampleRate, and Onoe in congested situations.

Relatively few work is available regarding the estimation of RA schemes. These solutions usually require a priori knowledge about the internal operation of the candidate RA scheme. Recently, Kim et al. [8] presented an analytical Markov chain model for WLAN including an estimation for the rate selection of the selected scheme Minstrel. Their estimation bases on the frame loss rates and models the internal behavior of the considered RA scheme. Further, Mirza et al. [9] presented a scheme that allows for a classification of RA algorithms on the basis of their observed MAC parameters, i.e., a *fingerprinting* of RA schemes. Their approach bases on the specific, internal rate change pattern for each scheme. As the authors target a precise classification of each algorithm reactively on the basis of relatively long observations, i.e., large packet traces, their scheme is not suited for a fast estimation regarding the behavior of the algorithm observed within the WLAN cell.

Chapter 3

DARA Principle

The goal of our Data Rate Estimation scheme “DARA” is to identify the rate selection behavior of STAs after a short period of communication denoted as Δt_{trial} which is just in the order of some beacon intervals, only. Note that this estimation does not assume any preliminary knowledge about the actual rate adaptation scheme applied on each STA.

3.1 Basic Functionality

DARA uses the following abstraction of the RA schemes: $f : x \mapsto y$, where y is the probability mass function (pmf) of the IEEE 802.11 PHY data rates for a large time span $\Delta t_{\text{long-term}}$ being in the order of several tens of seconds. Here, y gives us the probability for each data rate to be used for successful transmissions. The input vector $x = (x_1, x_2)$ consists of two components both calculated over a small Δt_{trial} instead, which is just in the order of some beacon intervals. The first, x_1, contains the mean values as well as the standard deviations of RSSI values, the number of transmission attempts per data frame, and the “busy airtime” [3] seen at AP. Further, x_2 is a small sample of y over the short Δt_{trial} aiming at potential internal effects of RA schemes which are not easily obtainable from off-the-shelf WLAN equipment.

Here, the mapping function $f(x)$ is completely unknown. Thus we first need to derive a function $\tilde{f}(x)$ by a training process. This is realized by observing the instances x and y over the time span Δt_{train} consisting of multiple, subsequent Δt_{trial}. While we compute one set of input parameters x for each short Δt_{trial}, y includes the rate selection probabilities over the whole observation period Δt_{train}. We choose $\Delta t_{\text{train}} \gg \Delta t_{\text{trial}}$ and $\Delta t_{\text{train}} < \Delta t_{\text{long-term}}$, whereby we select Δt_{train} such that it already gives us approximately a ”long-term” distribution of the successfully applied data rates.

We learn the mappings of different observed x and y pairs by means of the ML approach of Gaussian processes in detailed described in Ch. 4. Once the model has learnt the mappings, we use it for estimations, i.e., we calculate the estimate y_* for a new input vector x_* obtained over a single period Δt_{trial}.

Note that different RA schemes may lead to different mapping functions. We build classes of WLAN devices that are highly probable to apply a similar adaptation scheme. This allows us to have an abstraction for each class of WLAN NICs. The classes of devices are identified itself by their triple ⟨*vendor ID, NIC model, firmware version*⟩ as it is highly probable that the same RA scheme is applied in similar NICs.

3.2 Motivation for the Selected Parameters

Note that for our observations, we stick to simple parameters being available and easily accessible on STA or AP side [10]: RSSI, successful data rates, transmission attempts, and "busy airtime". The latter is defined as the percentage of time that the wireless medium was sensed busy seen by AP [3] . Acharya et al. [11] have shown that this measure is suitable to reflect the load situation and to relate it with resulting packet losses in a WLAN cell, thus allowing us to identify the operational point of the network from the load perspective.

3.3 Considerations for Practical Usage

Let us now discuss some practical aspects for the measurements described above. We assume that AP and STAs support the IEEE 802.11k amendment using the 802.11k statistic request [3], such that the AP can trigger a STA to start measurements specified in this request. Specifically, the AP requests the STA to maintain a list of its transmission attempts which includes the fraction of successful and non-successful transmissions (i.e., without the reception of an ACK). In parallel, the AP keeps track of successfully received data frames and logs for each of them the data rate as well as the RSSI values. After Δt_{trial}, STA reports its data with the 802.11k STA statistics response [3].

For the description of the device type, AP additionally requests IEEE 802.11v diagnostic information [3] from the STA to obtain device information. Among others, the 802.11v diagnostic information includes the required triple of manufacturer ID, the NIC model, as well as the firmware version.

Once AP has obtained all data over Δt_{trial}, it passes them to a computing entity (CE) which conducts the calculations for both the training as well as the estimation. For the latter, CE checks whether it has been trained already with data from a NIC with the triple ⟨*vendor ID, NIC model, firmware version*⟩. If so, it loads the triple-specific parameterization for the ML model and the estimation for y_* on the basis of the input parameters x_* starts.

When a device with an unknown triple, i.e., NIC class, appears in the WLAN hotspot, no predictions are possible as the model is yet untrained. Further, if a device belongs to a known class of NICs but the actual estimation for x_* delivers a high variance, i.e., the model itself has been pretty unsure about the quality of its output, we take device's behavior for the re-training of the existing model. For both cases, the AP collects further measurements of these STAs over the longer time spans Δt_{train}. Afterwards, CE conducts the (re)training for the concerned NICs.

Chapter 4

ML Model used by DARA

We assume that the rate adaptation algorithm is usually unknown and consequently the form of our mapping function $f(x)$ is not available. And even in those cases where the algorithm is open source, that does not help much: the relation between x and y is the result of a complex dynamics and could be only calculated by simulating it stepwise.

Therefore we do not try to model this function explicitly, but use a non-parametric approach to learn the complete mapping from x to y. For that purpose, we assume that $f(x)$ is a sample from a *Gaussian Process* (GP) [12], which is a distribution over functions with the expectation values

$$E[f(x_i)] = 0 \quad \text{and} \quad E[f(x_i)f(x_j)] = k(x_i, x_j), \tag{4.1}$$

where x_i and x_j denote arbitrary input vectors. Any finite set of function values $\{f(x_1), f(x_2), \dots\}$ comes from a multinomial Gaussian distribution fully defined by the first and second moments given in (4.1). This mapping is extended to multidimensional outputs, like our data rate distribution, by adding an input value which selects the desired output component.

The kernel $k(x_i, x_j) = \mathrm{Cov}(f(x_i), f(x_j))$ describes variance and autocorrelation of the function $f(x)$, which are easier to specify than a parametric form. For our model we choose the squared exponential kernel with *Automatic Relevance Determination* (ARD):

$$k_1(x_i, x_j) = \theta_0 \exp\left[-\frac{1}{2}\sum_{m=1}^{D} \eta_k (x_{i,m} - x_{j,m})^2\right].$$

Here $\theta_0, \eta_1, \dots, \eta_D$ are *hyperparameters*, which can be adapted to the training data set $B = \{x_i, y_i | i = 1...N\}$. High values of the weight η_m indicate that the m-th element of x is relevant, while its influence on y is small for low values [13]. We also take into account that the model might not fit perfectly and assume Gaussian observation noise with variance θ_2:

$$k_2(x_i, x_j) = \theta_2\, \delta_{i,j}.$$

Linear effects are modelled explicitly by a linear kernel

$$k_3(x_i, x_j) = \theta_3 \sum_{m=1}^{D} x_{i,m}\, x_{j,m}$$

with hyperparameter θ_3. Finally, the full kernel used in the ML-model is given by the sum

$$k(x_i, x_j) = \sum_{n=1}^{3} k_n(x_i, x_j).$$

As we do not know the correct values of the hyperparameters $\theta = (\theta_0, \eta_1, \dots, \eta_D, \theta_2, \theta_3)$ before, these are adapted to B by minimizing the negative marginal likelihood

$$-\log p(y|x, \theta) = \frac{1}{2} y^\top K_B^{-1} y + \frac{1}{2} \log |K_B| + \frac{n}{2} \log 2\pi, \tag{4.2}$$

where K_B is the covariance matrix between the y contained in B given by $(K_B)_{i,j} = k(x_i, x_j)$ [13]. After this minimization, performing GP regression only requires linear algebra [13]. The estimate y_* is given by

$$y_* = E[f(x_*)] = k_*^\top K_B^{-1} y,$$

where x_* denotes the test input and k_* is the vector of covariances between x_* and B. We also get the variance

$$\sigma_*^2 = \mathrm{Var}(f(x_*)) = k(x_*, x_*) - k_*^\top K_B^{-1} k_*$$

of the function at the test point, which is an indicator for the uncertainty of the prediction. Especially this is an advantage compared to non-Bayesian methods, where confidence intervals are more difficult to determine. Additionally, this non-parametric approach has an automatic tradeoff between model complexity and fit of the training data.

Chapter 5

Scenarios for DARA's Evaluation

For a first "proof of concept" of DARA, we consider two real RA schemes—AARF/AMRR and Minstrel. For both, we are interested in the obtainable accuracy for estimations if we train DARA with data of random durations Δt_{train}. For this work, we study estimates for VoIP traffic as its strong QoS requirements make it very interesting for a support of timely decisions, e.g., for handovers or WLAN reconfigurations. Large-scale measurements showed a significant fraction of calls having a duration above 25 s [14]. To account also for larger call lengths, we select calls for the training such that Δt_{train} uniformly distributes between 20 and 40 s. For $\Delta t_{\mathrm{trial}}$, we use ten consecutive 100 ms beacon intervals observing the short-term RA behavior for up to 50 VoIP packets.

Methodologically, we first consider simulations by ns-2 with detailed SINR, RA, and fading models [15]. Secondly, we further conduct a campaign of measurements in our WLAN testbed located in our office environment on campus, since interference as a result of high hotspot densities in today's WLAN channels is not adequately covered by the simulations.

This Chapter gives a short overview over the selected RA schemes, elaborates on the simulation and testbed setup in which we obtain the data for training and estimation, and describes the conducted training process with its tool chain.

5.1 Selected Rate Adaptation Schemes

Both in simulations as well as measurements, we apply AARF/AMRR and Minstrel. Minstrel is known to be one of the most advanced schemes [6, 9], while we consider also the popular family of AARF/AMRR schemes, which has been shown to struggle with congested environments [7].

The Adaptive ARF (AARF) scheme is an extension of the *Automatic Rate Fallback (ARF)* algorithm [16] which increases the data rate after ten successful transmissions and reduces its rate if two transmission attempts fail in a row. As this static adaptation has been shown to be susceptible to fluctuating wireless channels, AARF dynamically adapts the thresholds for up- and downgrading the data rates in addition to the ARF operation. Adaptive Multi Rate Retry (AMRR) is a specific flavor of AARF enabling to deal with the specific timing behavior of the Madwifi driver. While in our simulations, we stick to the original definition of AARF [16], we apply AMRR in our measurements as it comes originally with Madwifi [5].

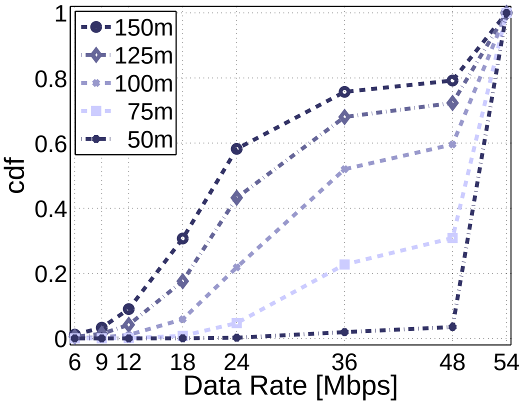


Figure 5.1: Distribution of data rates with "perfect" rate selection

The second scheme, Minstrel, conducts a certain fraction of transmissions for testing suitable, alternative data rates. From this, the algorithm estimates the expected throughput for each, and finally selects the rates with the highest expected throughputs. Periodically, Minstrel evaluates the estimation of the expected throughputs each 100 ms; further details about Minstrel are given at [17].

5.2 Setup for the simulation-based Training

In our simulation, we train the ML model at an operational point of the network at which no further VoIP flows can be accommodated in the WLAN cell (QoS constraints are given below); this results into about 40 VoIP nodes transmitting concurrently in the hotspot. Thereby, the first randomly selected half of nodes applies AARF, while the rest operates Minstrel.

5.2.1 Simulation Model

The WLAN hotspot is represented by an IEEE 802.11g AP that is 11e-capable by providing EDCA functionality. All WLAN devices apply 802.11g Extended Rate Physicals (ERPs) with OFDM modulation—from 6 up to 54 Mbps. The 802.11e/g parameters are chosen according to [3]. To take into account that radio signals are not only affected due to path loss but also due to multipath propagation, we use a Log-Distance Path Loss, a Ricean Fading and a SINR model. Models' details of the applied network simulator ns-2 are described in [15]. Additionally, we extended the WLAN model by the two RA schemes AARF and Minstrel.

5.2.2 Traffic Model and QoS Constraints

For the generation of the VoIP traffic, we use the same model and parameterization as in [15], i.e., an exponential ON/OFF model generating packets according to the ITU-T codec G.711 (160 Byte audio packets each 20 ms during ON periods). Similar to [15], we define the QoS limit for VoIP in terms of losses, consisting of lost and late packets. If five or more percent of the VoIP packets are lost, the quality is assumed to be lousy.

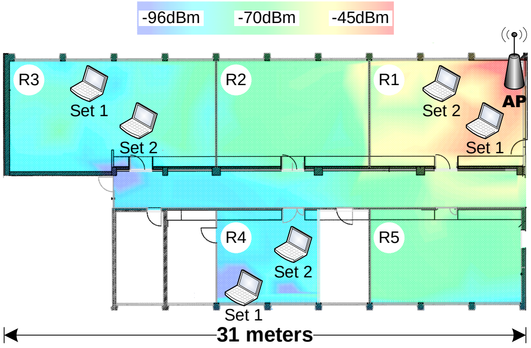


Figure 5.2: Testbed environment

5.2.3 Scenarios

We investigate five scenarios in which we vary the size of the quadratic area with edge lengths of 50, 75, 100, 125, and 150 meters; the AP resides at a corner of this area; the STAs are equally distributed. Larger edge lengths thereby lead to greater sizes of the areas such that the data rates applied for 802.11 transmissions reduces. Fig. 5.1 shows the cumulative distribution function (cdf) of data rates for each edge length, if the rate selection is done on ideal "perfect" knowledge about the rate to be applied.

5.2.4 Selection of Training Nodes

As we want to cover high- as well as low-rate STAs, we train the model on the scenario where the "worst rate" distributions appear, i.e., the large scenario with 150 m edge length. For the training of our model, we assign all nodes to bins with radii around the AP of multiples of 30 meters. With 150 m, this results into 5 bins. Then we randomly select one NICs of each type (either AARF or Minstrel) from each bin as a training node.

5.3 Setup for the testbed-based Training

In addition to the simulations, we verify our approach in a real testbed which additionally includes interference from other WLAN hotspots in overlapping channels.

For the training with testbed data, we follow a two stage methodology as pure VoIP traffic leads to a rather low-loaded WLAN with our available hardware. Firstly, we train our ML model with data observed in pure VoIP scenarios. Secondly, we consider measurements from scenarios with VoIP together with background loads from TCP downloads, e.g., resulting from HTTP downloads or TCP video streaming solutions. In all measurements, all VoIP nodes together apply the same RA scheme—either AMRR or Minstrel.

5.3.1 Testbed Environment

The measurements are conducted in three different rooms on the first floor (Fig. 5.2) of our building located on campus. The university itself operates a large campus-wide WiFi network to offer wireless access. As a result, our measurement environment has to deal with strong interferences from different floors as well as from the neighbor buildings. We operate our testbed on WLAN channel 1; during the measurements we have seen between 15 and 20 other WLAN hotspots within the channels 1 to 5.

In the testbed, we apply four WLAN STAs and one AP, all of them running on laptops of the type IBM R50/R51 or Lenovo R500 (CPUs of 1.5 to 2.53 GHz and 768 MB to 2 GB RAM). They include CardBus WLAN NICs with Atheros chipsets (Netgear WAG511 v1/v2 and WPN511) all driven by the Madwifi version including the most recent Hardware Abstraction Layer (HAL) [18]. In our settings this helps us avoiding Madwifi's well-known HAL stability problems [19] over large time scales from a couple of hours up to several days. For all nodes, transmit power is fixed to 10 dBm, power save as well as RTS/CTS is turned off, and 802.11g ERP-OFDM with 6 to 54 Mbps is applied. Further we use 802.11e EDCA with the VoIP (VO) QoS class and the Background (BK) class for TCP (with Madwifi's standard settings for 802.11e parameters [3] of BK: AIFSN = 7, $\mathrm{CW}_{\mathrm{min}} = 4$, $\mathrm{CW}_{\mathrm{max}} = 10$, and VO: $\mathrm{AIFSN}_{\mathrm{AP}} = 1$, $\mathrm{AIFSN}_{\mathrm{STA}} = 2$, $\mathrm{CW}_{\mathrm{min}} = 2$, $\mathrm{CW}_{\mathrm{max}} = 3$). All notebooks are operated with Debian 6.0 and the vanilla kernel 2.6.32.46 without experimental modules. We wired the WLAN devices by an 1 Gbps Ethernet backplane over which we transport measured data to an SNMP manager located on a separate server. Further, the devices are synchronized with PTP (Precision Time Protocol) for a precise, internal time stamping of measurement data.

5.3.2 Collection of Data

On the sender side, we built a wrapper around the RA module tracing all in- and outputs, i.e., the rate selection, the number of retransmissions, information about the success of the transmission, the sequence number, and the timestamp for each data packet. On the receiver side, we log all events in the Madwifi driver that arrive at the routine handling successfully received data packets. There, we trace the applied data rate, the packet size, the RSSI value, the sequence number, and the according timestamp. Further at AP, we trace the occupied airtime of WLAN traffic in up- and downlink direction per beacon interval of 100 ms by periodically reading out the related registers [11].

5.3.3 Placement of WLAN Nodes

Fig. 5.2 shows a snapshot of the RSSI values from our hotspot obtained with the software "Netspot" [20] in our office environment. Within the labeled rooms *R1* to *R5*, we measured the RSSI-values from our hotspot at least at 20 different positions per room. Although this is an exemplary snap-shot, it helped us understanding the propagation of the radio signals in our testbed, where we placed our AP at the upper right corner of room 1. Repeating these snap-shots on different days showed that rooms 3 and 4 offer the most interesting positions as they show strong fluctuations in RSSI within small areas. As a result, we conduct the following placements of WLAN STAs: our "bad" STA is put in room 4 in the low RSSI region, the "medium" STA resides in room 3, while the "good" STA is in room 1 together with the AP. Further, we place our TCP background-load STA in room 1 close to the good STA.

Each WLAN node in room 1, 3, and 4 is placed at two different positions (set 1 and 2) with a distance of about 3.5 m to each other. To separate the training of the ML model from the actual estimation, we use the first set of positions for the training and data from the second set for estimations, only.

5.3.4 Traffic Model

We emulate VoIP traffic by generating packets with a 160 Byte audio payload each 20 ms. The same QoS limits as in Sec. 5.2.2 apply. Further, we generate background (BG) loads with TCP Downloads (TCP-BG). During all measurements, our TCP-BG STA runs a download at different goodputs from a server within our wired infrastructure. The duration of a complete download always exceeds the measurement duration in every setting. To study different operational points of the WLAN hotspot, we vary the goodput of the TCP-BG STA by throttling the download goodput on server side. We train AMRR and Minstrel both with pure VoIP and different TCP-BGs. For AMRR, TCP-BG goodput is set to 0.5 and 1.5 MB/s, while Minstrel training is conducted with 0.5 and 1.75 MB/s. The high goodput values lead to operational points of the hotspot such that medium and bad STA start to perceive QoS degradations in some Δt_{train}.

5.4 Training the ML Model

As described in Ch. 4, the hyperparameters of the ML model need to be adapted, i.e., optimized, for the collected training data x and y. We feed x and y into the gpml-framework [21] which calculates the negative log likelihood from Eq. 4.2 for the selected kernel and its hyperparameters. We obtain optimal hyperparameters separately for simulation and measurement data by minimizing the negative log likelihood with Matlab's optimizer "fminunc" [22] which applies a subspace trust-region method on the basis of the interior-reflective Newton method. To ensure that we have actually found a (local) minimum in the negative log likelihood, we consider the gradient (being sufficiently close to zero) and the eigenvalues of the Hessian matrix (being all greater than zero).

Chapter 6

Evaluation of DARA's Accuracy

To judge the accuracy of DARA's estimators obtained in Ch. 5 for AARF/AMRR and Minstrel, we now conduct estimations for new input parameters x_*. For this, we shift our terminals to different positions such that we obtain x_* for which DARA was not trained.

6.1 Accuracy Metric

At each node selected for an estimation, we conduct 32 estimations y_* and compare the estimated data rate distributions with the real values y. As a metric for the accuracy, we consider the mean squared error for each data rate (MSE$_{rate}$) [13]

$$MSE_{\mathrm{rate}}(y_*, y) = \frac{1}{N} \sum_{i=1}^{N=32} (y_{*_i} - y_i)^2.$$

Confidence intervals with a 95 percent level have been computed and stay within $+/-0.06$ around the mean if not stated otherwise; thus we do not plot them for better visibility.

6.2 Simulation Results

For each of the rate adaptation schemes, we select 10 test nodes with radii of multiples of 15 meters around AP.

The results are shown in Fig. 6.1 and 6.2 for AARF and Minstrel. The values are plotted over the data rates and for the different positions of the selected test nodes. Both plots show the mean measured and the mean estimated values as well as MSE over the 32 repetitions. From these results, one can see that the AARF estimates perform on average generally well, but impose certain errors at high rates for near nodes and some smaller errors at medium rates for middle positioned nodes. Considering Minstrel estimates, the similarity between the curves of the measured values (Fig. 6.2(a)) and the estimated values (Fig. 6.2(b)) is also dominant. However, additional prediction errors appear for the furthest nodes transmitting at low rates. Note that these errors are mostly conservative in the sense that the predictions include lower rates than contained in the actual, measured values.

6.3 Results from the WLAN Testbed

In our testbed, we conduct all measurements for the estimation from the second positions (set 2 in Fig. 5.2) and with TCP-BG loads of 0.25, 0.75, 1.25, and 2.0, MB/s (denoted as low TCP1 and 2, medium TCP, and high TCP). Further, the data for estimations have been obtained a couple of days after the training measurements (Ch. 5). Typical for highly utilized WLAN channels, our environment also shows strong, time-varying interferences from other WLAN APs.

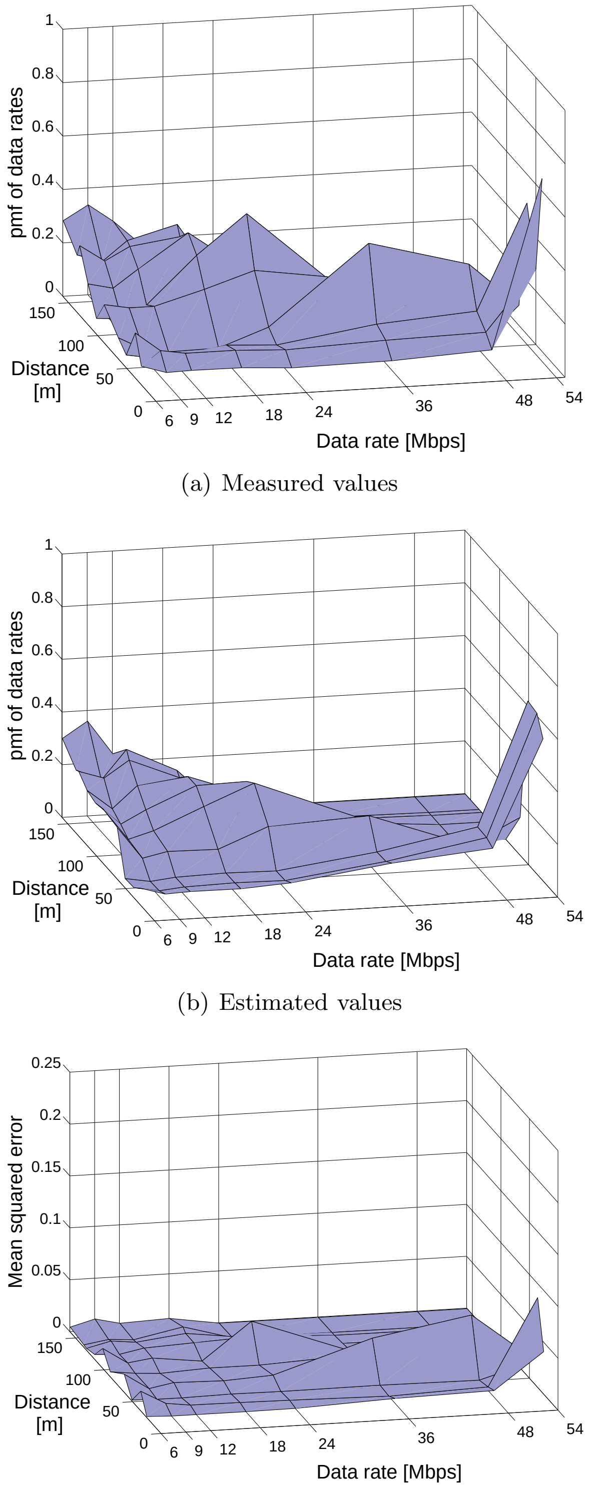
Results for the estimation in our testbed are shown in Fig. 6.3 and 6.4 for AMRR and Minstrel. Similar to the simulation results, the estimations for AMRR show small errors for the medium positioned node 3 (in the figures, each node number refers to the room in which it is placed on its “set 2” position). Further, there appear some errors for node 1 at medium and high TCP-BG scenarios. At these points, confidence intervals for node 1 are within MSE + / − 0.04 to 0.11.

Contrary, Minstrel estimations in Fig. 6.4(c) show more and stronger MSEs than AMRR estimates. The smaller MSEs for low RSSI regions were identified in simulations already. However, in the testbed, large MSEs were observed for data rates above 18 Mbps, appearing pairwise mostly at 24 and 36 Mbps especially for the medium node 3. These pairwise errors result from a “wrong” guess of the ML model regarding the exact data rate: Instead of correctly estimating a large fraction of transmissions at a certain rate, the model predicts this at the neighbor data rate—leading to large MSEs for both.

6.4 Conclusions from the Estimation Results

From the simulation and testbed results, we conclude that obtaining the exact pmf of the data rates is error prone for the sophisticated RA scheme Minstrel. Especially, in the testbed setup dealing with strong interference from other hotspots, the selected ML model is not completely capable of covering Minstrel’s internal, statistical behavior for the high rates of 24 Mbps and above and thus “guesses” a certain fraction of data rates at the direct neighbor rate. However, note that DARA does not have any knowledge about the internal behavior of an RA scheme and is only estimating on the basis of simple accessible PHY/MAC parameters described in Ch. 3.2.

Thus, the results reveal that—although the exact pmf cannot obtained correctly in all cases—one is still able to determine a “trend” for a considered STA, i.e., whether it operates in the low-rate regions or in the higher-rate areas—being just dependent on the selected RA scheme. Even such a trend may help in today’s network in various applications.

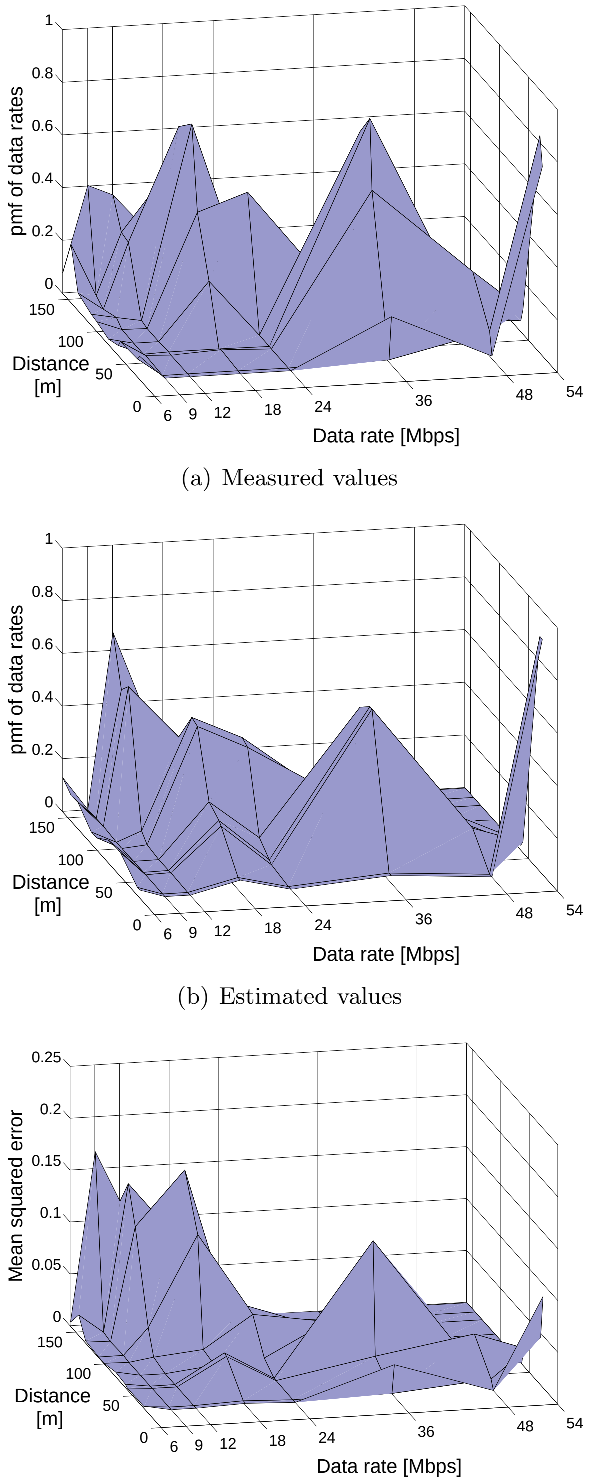


(a) Measured values

(b) Estimated values

(c) Mean squared error

Figure 6.1: AARF: simulation results

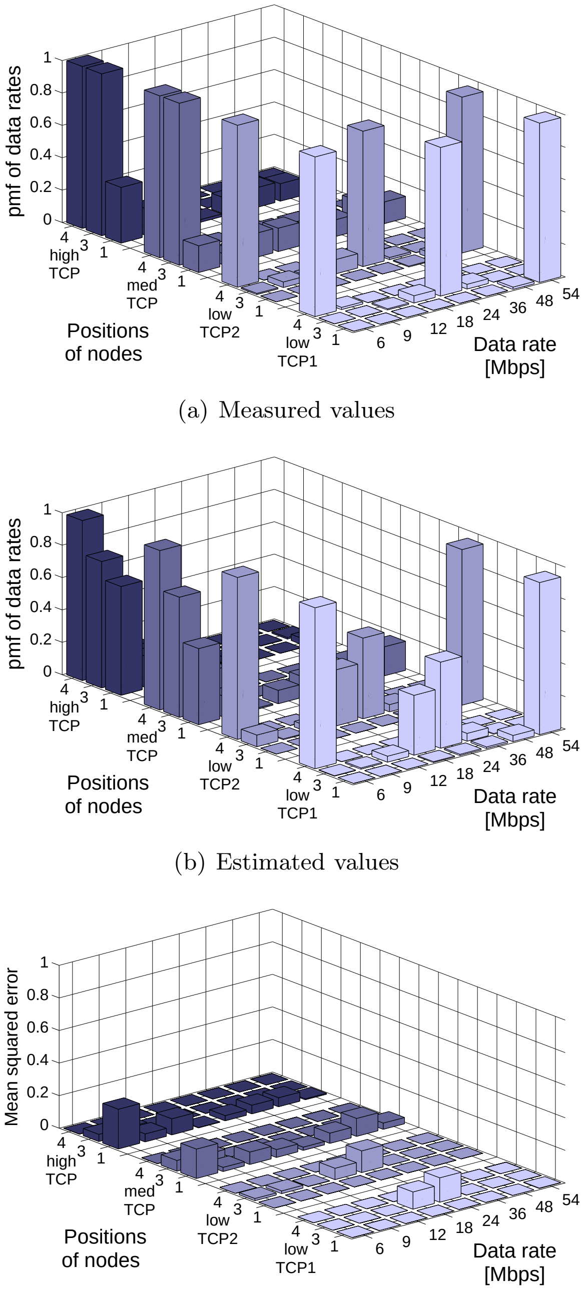


(a) Measured values

(b) Estimated values

(c) Mean squared error

Figure 6.2: Minstrel: simulation results

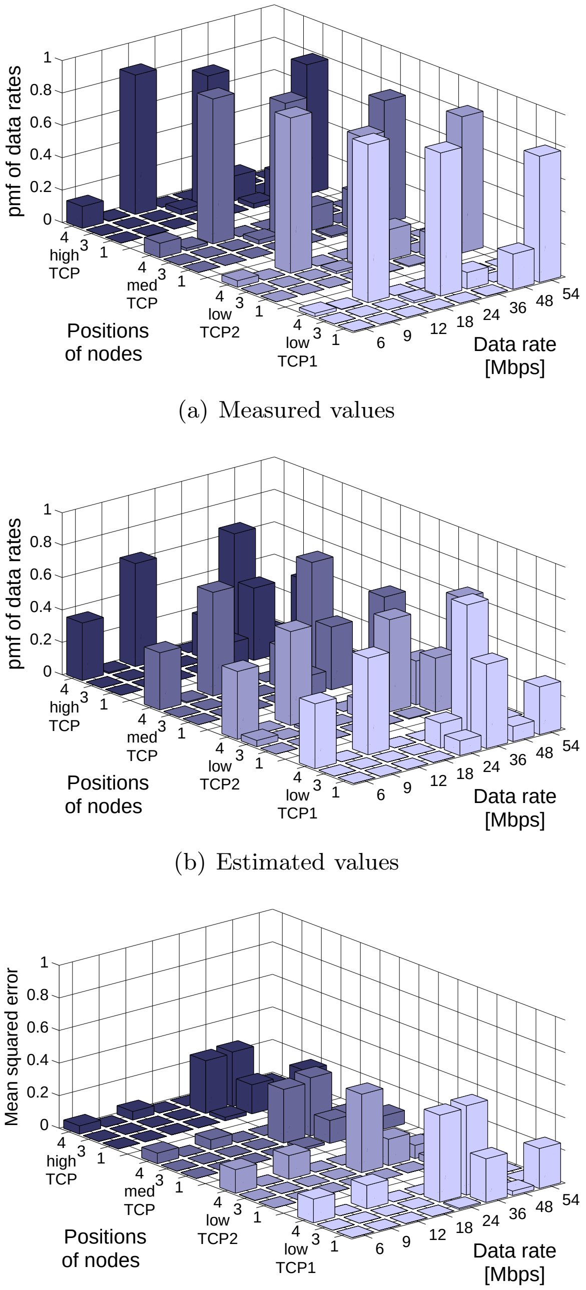


(a) Measured values

(b) Estimated values

(c) Mean squared error

Figure 6.3: AMRR: measurement results



(a) Measured values

(b) Estimated values

(c) Mean squared error

Figure 6.4: Minstrel: measurement results

Chapter 7

Applying DARA for Technology Selection

Offloading traffic from cellular technology (CT) to WLAN is an attractive solution for dealing with the rapidly increasing mobile traffic [1, 2]. Further, WLAN hotspots offer services such as VoIP much cheaper to end-users than mobile operators. In line with this, “Smart Offload” extensions envision to offload all services (voice, SMS, MMS, etc.) of mobile operators to WLAN hotspots [23]. Such a movement of traffic streams happens not necessarily one way. In order to accommodate the maximum number of streams at WLAN hotspots, some streams may be pushed back from WLAN into CT—allowing more “suitable” offloads further freeing the capacity in CT.

We consider a hotspot scenario located in a large area, e.g., within a shopping mall, where the number of active VoIP users by far exceeds the capacity of the WLAN cell. With the VoIP users being in the coverage of both the WLAN hotspot as well as CT, we target to unload CT at most. For this, we aim at maximizing the total number of served VoIP users in WLAN, while assuring that their QoS levels do not undergo their limits given in Ch. 5.2.2. For this maximization, we target a selection of VoIP STAs for WLANs which conduct their transmissions at high data rates, while “low-rate” STAs are pushed to CT.

In our performance evaluation with the simulation settings described in Ch. 5.2, we study the achievable improvements regarding the maximum VoIP capacity of a WLAN hotspot if one bases the selection of VoIP streams on the rate adaptation behavior of a STA. For this, we compare DARA with the common RSSI-based decisions.

7.1 System under Consideration

We consider a cellular technology enhanced by a WLAN hotspot. The WLAN cell is completely within the coverage area of CT. All terminals have a NIC for WLAN and CT. We assume that they can perform vertical handovers being conducted unnoticeable for the end users. All NICs are IEEE 802.21-capable [24] such that the networks can trigger a handover for users’ devices. WLAN AP and STAs support IEEE 802.11k/v with STA statistics and diagnostic reports [3].

We assume that highly-mobile end users are served by CT, while all stationary users within CT may be subject for consideration as handover candidates if they are within the coverage of the WLAN hotspot.

For DARA, our Computational Entity “CE” conducts the required calculations in negligible time in the order of some hundred milliseconds. CE can be run either on a separate device that is co-located to the AP or on a centralized server being responsible for a group of APs, which is the common scenario in enterprise WLAN networks today.

7.2 Schemes for the Selection of WLAN Flows

We apply the following basic algorithm with which we decide for a selection of VoIP users for the WLAN hotspot. We evaluate periodically each three seconds whether the QoS values of the served stations (STAs) in WLAN have been violated. In case of harmed QoS limits, we select randomly one of the STAs with bad QoS and trigger a handover to CT by means of IEEE 802.21. Contrary, if no QoS violation has occurred, we accommodate an additional user in the WLAN cell by selecting him in a random-uniform fashion from all STAs being within WLAN coverage as well as being connected to CT. If not done before, the new user associates with the WLAN cell. Upon the start of his uplink transmissions, we observe his behavior over Δt_{trial} and determine whether to keep him in WLAN afterwards. If we do not select him for WLAN access, we trigger a handover to CT with IEEE 802.21.

We study two different schemes to determine whether to handover or to keep a user in WLAN—RSSI-based decisions as well as selections on the basis of DARA (DARA-S).

7.2.1 RSSI-based Decisions

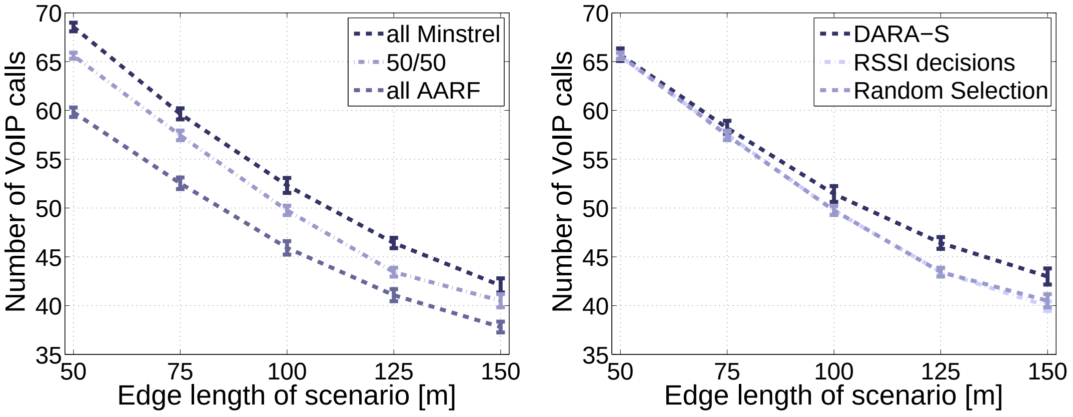
It is a common approach to base the network selection on the received signal strength of a considered STA [25]. For this, we collect the RSSI-measurements of STA’s uplink data transmissions at AP over Δt_{trial}. Afterwards, we average the measured values and compare it with a threshold. This threshold has been selected such that STAs should be able to transmit the high rate part of their frames (on average) at least with 12 Mbps.

7.2.2 DARA-based Selection (DARA-S)

Here, we identify low rate STAs by means of DARA’s estimation applying the hyperparameters from Ch. 5.2. To decide for the permanent selection of an arriving STA, we consider the probabilities of the low rates from the estimated histogram. Similar to the RSSI-based decisions, we target to avoid 6 and 9 Mbps transmissions. We trigger a handover, if the sum of probabilities for the *set of low rates*, i.e, 6 and 9 Mbps, is above 20 percent. We selected this threshold to allow RA schemes also a “testing” of very low data rates for their internal statistics.

7.3 Performance Evaluation

In order to identify the gains of DARA-S, we finally conduct a campaign of simulations in which we study the maximum number of VoIP clients in the WLAN cell.



(a) Mixes of RA schemes

(b) DARA-S and RSSI

Figure 7.1: VoIP capacity in different scenarios

There, we follow a two-stage methodology. Firstly, we study the impact of the different RA schemes. For this, we analyze the VoIP capacity if no sophisticated selection scheme for arriving VoIP terminals is applied, i.e., STAs are selected in a random fashion. There, we study three different mixes of rate adaptation algorithms: in the first two mixes, AP and all STAs together either apply AARF or Minstrel (denoted as *all AARF* and *all Minstrel*). In the third one, the AP uses Minstrel while the (randomly selected) first half of STAs applies AARF and the rest Minstrel (denoted as *50/50 mix*).Then, in the second stage, we conduct our selection schemes on the 50/50 mix and compare it with the random selections. Lastly, we study the accuracy of DARA-S's classifications.

7.3.1 Mixes of RA Schemes

Fig. 7.1(a) shows the results for the mixes of STAs with different RA schemes. "All Minstrel" outperforms all others, while the "all AARF" case leads to the smallest number of VoIPs in all scenarios. The "50/50 mix" already leads to a significant improvement over "all AARF", which shows how different RA schemes influence the capacity of a cell, thus motivating sophisticated selection schemes taking the issue of the rate selection into account.

7.3.2 Comparison of the Selection Schemes

Fig. 7.1(b) shows the results with DARA-S and RSSI decisions for the 50/50 mix. Both are similar for the high-rate scenarios of 50 and 75 meters as these do not have to deal with low rates as result of the channel quality (cf. rate distribution of ideal adaptation in Fig. 5.1). With increasing scenario sizes, the probability of low-rate STAs increases. There, DARA-S outperforms RSSI decisions greatly: The number of VoIP STAs increases (3.4, 6.8, and 7.6 percent for 100 to 150 m scenarios). Note that the ideal rate distribution of the 150 m scenario (Fig. 7.1(b)) is still far away from being a pure low-rate case as rates of 12 Mbps and below are just applied in ten percent of the transmissions. Thus, this scenario represents the case of links with medium, fluctuating quality as can be found in typical in-house environments.

In these scenarios, DARA-S is also not significantly different from the high *all Minstrel* curve in Fig. 7.1(a), i.e., DARA-S leads to a similar performance as the pure Minstrel mix with the simple random decisions. Further, we plotted the random selection curve of the *50/50 mix* in Fig. 7.1(b). In all scenarios, the *50/50 mix* with random selection and RSSI decisions are not significantly different. In other words, applying RSSI does not lead to any improvements over the very simple random decisions, in which a STA is triggered to handover to CT once its minimum QoS level is violated.

7.3.3 Accuracy Analysis of DARA-S

To evaluate the performance of our classifier recognizing low-rate STAs, we conduct an accuracy analysis [26] for DARA-S by observing the classification decisions plus the rate selection behavior afterwards for each node in each scenario. As a result, we can judge about the correctness of each decision by true positives (tp) and true negatives (tn), i.e., the classification was correct and STA's network access is granted (tp) or denied (tn) correctly. Contrary false positives (fp) and false negatives (fn) imply a wrong classification, such that the STA is spuriously kept in the network (fp) or forced to conduct a handover (fn). For a single representation of these four cases, we have selected Matthews Correlation Coefficient (MCC) as it is known to provide a balanced evaluation compared to an analysis considering just simple true or false positive rates [26]. MCC is defined as

$$\mathrm{MCC} = \frac{\mathrm{tp}\,\mathrm{tn} - \mathrm{fp}\,\mathrm{fn}}{\sqrt{(\mathrm{tp}+\mathrm{fn})(\mathrm{tp}+\mathrm{fp})(\mathrm{tn}+\mathrm{fp})(\mathrm{tn}+\mathrm{fn})}}. \tag{7.1}$$

MCC is the correlation coefficient between DARA-S's classifications and the observed rate selections of the STAs. A MCC of -1 represents a total misalignment, $+1$ indicates a complete agreement, and 0 implies a purely random performance.

We calculated MCC for different durations Δt_{obs} ranging from 10 to 600 s over which we observed STA's rate selection process after the actual classification (Fig. 7.2). In the high-rate scenarios of 50 and 75 meters, MCC is not defined as these cases frequently lack true and false negatives [26]. For the other scenarios with lower rates—and as a result, also true and false negatives appear—MCC increases with longer observation periods. For Δt_{obs} of more than 50 s, MCC is far above 0.3 in all scenarios demonstrating the classification accuracy of DARA-S even for rather short observation ranges of about a minute. Note that these results lie far above simple randomized decisions (with MCCs around zero). The longer we observe after an estimation, the higher becomes the accuracy until it reaches its maximum.

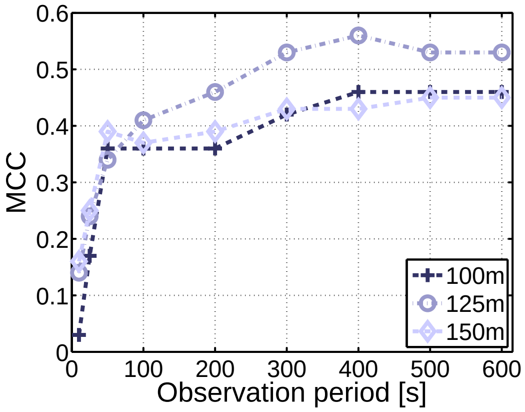


Figure 7.2: Matthews Correlation Coefficient (MCC) for different Δt_{obs}

Chapter 8

Conclusions

This paper presents our novel data rate estimation scheme “DARA” for WLAN networks. The core mechanism of DARA estimates the rate selection of a WLAN end-user device just by observing its behavior on short time scales—without having any knowledge about the applied rate adaptation algorithm. We study DARA’s estimation accuracy for the selected RA schemes AARF/AMRR and Minstrel with data from simulations as well as from the WLAN testbed operated in our office environment. Our results show high accuracies for AARF/AMRR and certain estimation errors regarding the precise shape of the rate distribution with Minstrel. Nevertheless, the results indicate that DARA is indeed able to identify “trends” of the rate selection for different RA schemes, i.e., whether a STA operates in the low or high-rate region. Finally, we utilize these trends in an application example in which we use DARA’s estimates as a basis for selecting suitable VoIP traffic in WLAN hotspots. Our performance evaluation between DARA and RSSI-based decisions shows significant gains in settings where a certain amount of end-user devices are struggling with fluctuating channels. Under these circumstances we gain up to about 8 percent with DARA.

Our future work will consider aspects for incorporating DARA into WLAN enterprise architectures. We aim at a reduction of signaling overheads, procedures for fast online re-training, and further sophisticated kernel functions stronger reflecting interference issues from real-life networks thus allowing for the estimation of exact rate histograms.

Bibliography

[1] S. Wiethölter, M. Emmelmann, R. Andersson, and A. Wolisz, "Performance evaluation of selection schemes for offloading traffic to IEEE 802.11 hotspots," in *Proc. of ICC*, June 2012, pp. 5423–5428.

[2] K. Lee, I. Rhee, J. Lee, S. Chong, and Y. Yi, "Mobile data offloading: how much can WiFi deliver?" in *Proc. of Co-NEXT*. ACM, 2010.

[3] IEEE, "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Std 802.11-2012, 2012.

[4] K. Medepalli, P. Gopalakrishnan, D. Famolari, and T. Kodama, "Voice capacity of IEEE 802.11b and 802.11a wireless LANs in the presence of channel errors and different user data rates," in *Proc. of VTC-Fall*, vol. 6. IEEE, sept 2004, pp. 4543–4547.

[5] The Madwifi Project Team, "The MadWifi Project." [Online]. Available: http://madwifi-project.org/

[6] W. Yin, K. Bialkowski, J. Indulska, and P. Hu, "Evaluations of MadWifi MAC layer rate control mechanisms," in *Proc. of IWQoS*, june 2010.

[7] E. Ancillotti, R. Bruno, and M. Conti, "Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks," in *Proc. of PE-WASUN*. ACM, 2008, pp. 7–14.

[8] I. Kim and Y.-T. Kim, "Realistic modeling of IEEE 802.11 WLAN considering rate adaptation and multi-rate retry," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1496–1504, Dec. 2011.

[9] M. Mirza, P. Barford, X. Zhu, S. Banerjee, and M. Blodgett, "Fingerprinting 802.11 rate adaption algorithms," in *Proc. of INFOCOM*. IEEE, April 2011, pp. 1161–1169.

[10] D. Dujovne, T. Turletti, and F. Filali, "A Taxonomy of IEEE 802.11 Wireless Parameters and Open Source Measurement Tools," *Communications Surveys Tutorials, IEEE*, vol. 12, no. 2, pp. 249–262, 2010.

[11] P. A. K. Acharya, A. Sharma, E. Belding, K. C. Almeroth, and K. Papagiannaki, "Rate Adaptation in Congested Wireless Networks through Real-Time Measurements," *IEEE Transactions on Mobile Computing*, vol. 9, no. 11, pp. 1535–1550, Nov 2010.

[12] C. M. Bishop, *Pattern Recognition and Machine Learning.* Springer, 2006.

[13] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning.* MIT Press, 2006.

[14] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, "Primary user behavior in cellular networks and implications for dynamic spectrum access," *Communications Magazine, IEEE*, vol. 47, no. 3, pp. 88–95, March 2009.

[15] S. Wiethölter and A. Wolisz, "Selecting Vertical Handover Candidates in WLAN Hotspots," TKN, TU Berlin, Tech. Rep. TKN-08-009, 2008.

[16] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 rate adaptation: a practical approach," in *Proc. of MSWiM 2004.* ACM, 2004.

[17] Madwifi Project Team, *Minstrel.* [Online]. Available: http://madwifi-project.org/svn/madwifi/trunk/ath_rate/minstrel/minstrel.txt

[18] ——, *Snapshot archive for madwifi-hal-testing: MadWifi HAL testing, Release 4126*, March 2010. [Online]. Available: http://snapshots.madwifi-project.org/madwifi-hal-testing/

[19] ——, *The Infamous Stuck Beacon Problem.* [Online]. Available: http://madwifi-project.org/wiki/StuckBeacon

[20] NetSpot Team, *Netspot 1.3: Free Wireless WiFi Site Survey Software for MAC OS X.* [Online]. Available: http://www.netspotapp.com

[21] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *J. Mach. Learn. Res.*, vol. 11, pp. 3011–3015, dec 2010.

[22] MATLAB, *version 7.13.0.564 (R2011b).* MathWork Inc., August 2011.

[23] Kineto Wireless, "Smart Offload for Smartphones," Whitepaper, 2010.

[24] D. Corujo, C. Guimaraes, B. Santos, and R. L. Aguiar, "Using an open-source IEEE 802.21 implementation for network-based localized mobility management," *Communications Magazine, IEEE*, vol. 49, no. 9, pp. 114–123, sept 2011.

[25] X. Yan, Y. A. Sekercioglu, and S. Narayanan, "A survey of vertical handover decision algorithms in Fourth Generation heterogeneous wireless networks," *Computer Networks*, vol. 54, no. 11, pp. 1848–1863, 2010.

[26] P. Baldi, S. Brunak, Y. Chauvin, C. A. F. Andersen, and H. Nielsen, "Assessing the accuracy of prediction algorithms for classification: an overview," *Bioinformatics*, vol. 16, no. 5, pp. 412–424, 2000.