SNR-based Reinforcement Learning Rate Adaptation for Time Critical Wi-Fi Networks: Assessment through a Calibrated Simulator

Giovanni Peserico
Autec s.r.l.
and *Dept. of Information Engineering*
University of Padova, Italy
giovanni.peserico@phd.unipd.it

Tommaso Fedullo
Dept. of Management and Engineering
University of Padova, Italy
tommaso.fedullo@phd.unipd.it

Alberto Morato
CMZ Sistemi Elettronici s.r.l.
and *Dept. of Information Engineering*
University of Padova, Italy
alberto.morato.3@phd.unipd.it

Federico Tramarin and Luigi Rovati
Dept. of Engineering "Enzo Ferrari"
University of Modena and Reggio Emilia, Italy
federico.tramarin@unimore.it luigi.rovati@unimore.it

Stefano Vitturi
National Research Council of Italy
CNR–IEIIT
stefano.vitturi@ieiit.cnr.it

***Abstract*—Nowadays, the Internet of Things is spreading in several different research fields, such as factory automation, instrumentation and measurement, and process control, where it is referred to as Industrial Internet of Things. In these scenarios, wireless communication represents a key aspect to guarantee the required pervasive connectivity required. In particular, Wi-Fi networks are revealing ever more attractive also in time- and mission-critical applications, such as distributed measurement systems. Also, the multi–rate support feature of Wi–Fi, which is implemented by rate adaptation (RA) algorithms, demonstrated its effectiveness to improve reliability and timeliness. In this paper, we propose an enhancement of RSIN, which is a RA algorithm specifically conceived for industrial real–time applications.**

The new algorithm starts from the assumption that an SNR measure has been demonstrated to be effective to perform RA, and bases on Reinforcement Learning techniques. In detail, we start from the design of the algorithm and its implementation on the OmNet++ simulator. Then, the simulation model is adequately calibrated exploiting the results of a measurement campaign, to reflect the channel behavior typical of industrial environments. Finally, we present the results of an extensive performance assessment that demonstrate the effectiveness of the proposed technique.

***Index Terms*—Factory Automation, Wi–Fi, Rate Adaptation, Reinforcement Learning**

I. Introduction

The Internet of Things (IoT) is spreading not only in the consumer field, but also in factory automation, process control and distributed measurement systems. As a matter of fact, several research activities are currently in progress to extend the IoT paradigm also in the aforementioned new fields of applications to implement the so called Industrial Internet of Things (IIoT) [1]. In this scenario, IoT-based Distributed Measurement Systems are now revealing attractive to provide real–time and continuous measurements possibly collected over wide areas. [2]. Within IIoT systems, where great flexibility needs to be guaranteed to smartly connected *things*, wireless connectivity plays surely a key role. Since Wi–Fi is one of the most popular and effective wireless systems pervasively available today, considerable research activity has been directed toward the adoption of Wi–Fi systems also in real-time and high-reliability contexts, such as those typical of modern Instrumentation and Measurement (I&M) applications [3], [4].

Nevertheless, the adaptability of the IEEE 802.11 standards to real–time and deterministic communications is still a challenge. A feature of Wi–Fi that demonstrated to be effective in this context is the multi–rate support. This is the possibility of dynamically selecting the transmission rate from a rather wide set of different rates, to cope with the variations of the communication channel status. Intuitively, high data rates can be used when the channel status is good, whereas lower rates (that use more robust modulation schemes) can be selected when the status is bad. Therefore, introducing effective Rate Adaptation (RA) algorithms may help to reduce the number of packet transmission failures and the consequent retransmission attempts, so that to achieve more reliable and timely communication, which represents a main requirement of the industrial communication field [5].

Several RA algorithms have been presented during the years, and even recently the topic attracted significant efforts [6]–[9]. In particular, in [10] the authors proposed a RA technique, specifically conceived for industrial real–time applications, based on either the measurement or the estimate of the SNR. Such a technique, called RSIN, allows to select the set of rates to be adopted for the transmission (and possible retransmission attempts) of a packet able to minimize the residual packet error rate, while ensuring to match a specific deadline. Also, in [11] the possibility of exploiting Reinforcement Learning (RL) techniques to enhance the performance of the RA activity has been assessed, showing to provide encouraging results.

In this work, we move from the above outcomes and investigate the design of an innovative RL–based RA policy that is able to account for the channel status by measuring the SNR, such as the aforementioned RSIN. The proposed assessment is carried out via a simulation model implemented on the popular OMNet++ simulator tool. However, in order to make the analysis as much realistic as possible, we first exploited the results of some experimental sessions to calibrate the simulation model and, subsequently, use it to evaluate the performance of the proposed RL–based RA technique.

In detail, the paper is organised as follows. Section II provides some theoretical background, about Reinforcement Learning and the possibility of using it for rate adaptation algorithms. Also, this section introduces some basic concepts about RSIN. Section III describes the calibration of the simulation model. Section IV presents the new RA policy. Section V reports and analyses the obtained simulation results. Finally, VI concludes the paper.

II. Theoretical Foundations and Related Work

A. Reinforcement Learning

The concept of Artificial Intelligence (AI) is nowadays spreading in several and diverse areas, and allows for a system to learn from data how to perform an activity. Within the AI framework, the well-known and widespread concept of Machine Learning (ML) represents a set of techniques adopted to solve a large amount of problems. Specifically, Reinforcement Learning (RL) represents an interesting technique whose uniqueness is the trial and error methodology, where an agent learn how to act on an environment from experience. Figure 1 well describes the RL behavior. More details about RL and its applications in communications and networking are discussed in [12].

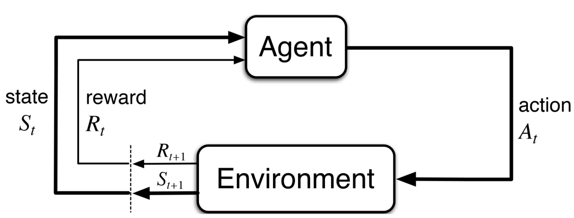

Figure 1. Reinforcement Learning (RL) components

The Agent at the step t in Fig. 1, starting from the state S_t and the reward R_t performs an action A_t on the Environment. In turn, the system moves to the new state S_{t+1} and a Reward R_{t+1} is provided to the agent, based on the goodness of the performed action A_t. These components form a Markov Decision Process (MDP), which is usually defined by the tuple $\{S, A, P, R\}$, respectively, state, action, probability and reward.

In this context, it is fundamental to define the relationship between the state of the system and the best action to perform towards the desired goal, that is, the best policy π consisting in the ordered sequence of pairs ($state_i$, $bestActionForState_i$). Several algorithms have been defined to this purpose based on the evaluation the expected cumulative reward associated with policy π, which can be performed either with a value function $V^{\pi}(s) = E_{\pi}\{R_t|S_t = s\}$, or an action-value function $Q^{\pi}(s, a) = E_{\pi}\{R_t|S_t = s, A_t = a\}$.

B. Reinforcement Learning applied to Rate Adaptation

The aforementioned approach has been adopted in a previous work [11], where we have investigated the design of a Reinforcement Learning-based Rate Selection algorithm (RLRA). In particular, the policy π was designed with the aim of achieving a good trade–off between the reduction of the packet loss (R_{loss}) and the maximization of the current rate. To this purpose, the rewards was defined as

$$\begin{aligned} R_{t+1} = \quad & \beta \cdot (-R_{loss_{t+1}} + R_{loss_t}) \\ & + (1-\beta) \cdot \frac{Rate_{t+1} - Rate_t}{Rate_{t+1} + Rate_t} - 1; \end{aligned} \tag{1}$$

Specifically, Eq. 1 aims at assigning higher Rewards to Actions that makes the rate higher and the loss lower. The performances of this algorithm in terms of the end–to–end delay revealed promising compared to other well-known RA policies (e.g. Robust Rate Adaptation Algorithm). For this reason, we take the RLRA algorithm in [11] as a basis for the design of a more complex RL–based RA scheme that exploits the knowledge of the SNR targeted at time critical Wi-Fi Networks for distributed real-time measurement systems.

C. Rate Selection for Industrial Networks

Rate Selection for Industrial Networks (RSIN) is a Rate Adaptation algorithm for IEEE 802.11 networks, proposed in [10] and specifically targeted to real-time and generally time-critical applications. RSIN builds upon the direct measurement of the SNR, and a complex optimization problem to find, for each frame to transmit, the sequence of transmission rates (one for each possible transmission attempt) ensuring the highest transmission reliability combined with the highest possible speeds, in order to deliver the frame within a specified deadline. Specifically, considering a wireless frame whose payload is l Bytes long, the measured SNR s as perceived by the intended receiver, and the associated deadline D, the objective of the RSIN optimization problem is to find *i)* the optimum number of retransmissions N_{opt}, and *ii)* the Retransmission Chain (RC), i.e. $r^{(i)} \quad \forall \quad i \in \{1, \ldots, N_{opt}\}$, that minimize the residual Packet Error Probability P_r, respecting the deadline constraint:

$$\begin{cases} \min\limits_{N \le N_{max}, r^{(i)} \in R} P_r(l, s \in S, N, r^{(1)} \ldots . r^{(N)}) \\ \max\limits_{N \le N_{max}, r^{(i)} \in R} t_{trans}(l, s \in S, N, r^{(1)} \ldots . r^{(N)}) \le D \end{cases} \tag{2}$$

With the formulation of Eq. (2), it appears evident that RSIN can be hence taken as an ideal benchmark for the assessment of the RL–based algorithm we propose in this work.

Nevertheless, RSIN may require a rather high computation load, as described in [13]. Indeed, a general implementation of the RSIN algorithm requires either to solve the problem in Eq. (2) for each transmitted frame, or to create and keep periodically updates a Look Up Table (LUT), in either cases to allow dealing with different packet payloads, deadlines and a dynamic channel behaviour. A simpler RSIN implementation may be realized by solving the problem in Eq. (2) *a priori* during the initialization stage and storing the obtained LUT, in this case limiting the algorithm dynamic.

III. Calibration of the Simulation Model

A. OMNeT++ Simulation Model

OMNeT++ is a C++ discrete event simulator, widely adopted to simulate communication networks and model the surrounding electromagnetic environment. Exploiting this tool we modeled a simple Wi-Fi network where two stations move at different speeds around an obstacle, as represented in Fig. 2.

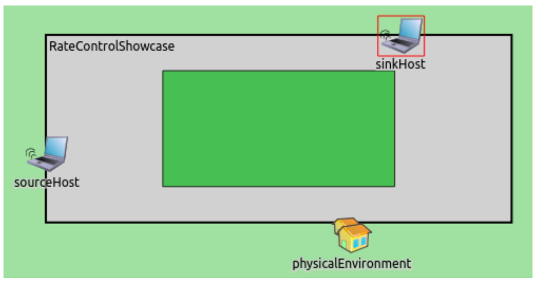

Figure 2. The simulated network

This setup allows the analyze of the behavior of the implemented RA algorithms introducing dynamic channel loss conditions, for instance simulating an harsh industrial environment with the presence of moving personnel or obstacles.

Without loss of generality, the *source* and the *sink* are set up to communicate through a basic IEEE 802.11g network, that hence allows transmission rate from 6 to 54 Mbit/s over a single OFDM stream. The choice of this (widespread but somehow old) version of the standard allows to maintain the assessment simple and effective, at the same time retaining the goodness of the obtained results, that do not depend on the specific Wi-Fi version. Indeed, the extension of the RA policies considered in this work to more recent IEEE 802.11 amendments is actually straightforward.

B. Calibration of the OMNeT++ simulation model

The behavior of the communication channel is a key aspects for a simulation model of wireless systems. This is particularly evident for industrial environments that are often characterized by the presence of (possibly moving) people and machinery. This requires the proper modeling of the electromagnetic environment, that reflect on the dynamic of the path loss and, consequently, on the bit error and frame loss R_{loss}. These models considers the SNR perceived at the receiver as the main input together with the type of modulation and coding scheme used by the devices.

The aim of this study is to provide an accurate assessment of different RA algorithms through simulations, that may be subsequently considered meaningful and representative of real environments. Hence, it becomes imperative to set up a precise calibration of the models implemented in the simulator with respect to experimental data about the channel behavior collected from the field.

To this aim, an important observation is that some specific models have already been defined for the industrial environment we are considering, as for instance the channel model “F” proposed by the Task Group n (TGn) during the development of the IEEE 802.11n standard, or even the one introduced in [14]: unfortunately they are rather complex and not currently implemented in OMNeT++. Thus, we resorted to analyze the already available models. The most suitable one, that already showed to be suitable in describing loss performance of Wi-Fi OFDM link is the NIST Error Model [15]. It is characterized by a complex yet thorough set of parameters allowing to fine tune the system to model the intended industrial environment.

For the latter crucial calibration phase, we focused on the relationship between packet error rate (PER) perceived at the Data-Link Layer, that hence depends on the outcomes of the NIST error model, with respect to the SNR at the receiver. To this aim, we referred to the experimental setup proposed in [16], where this PER-SNR relationship has been determined experimentally. Reproducing these measurements data as a reference, we hence fine tuned the main parameters of the OMNet++ NIST error model, in a typical calibration procedure. The results are reported in Fig. 3, where the PER–SNR curves obtained with OMNeT++ are compared with the experimental data.

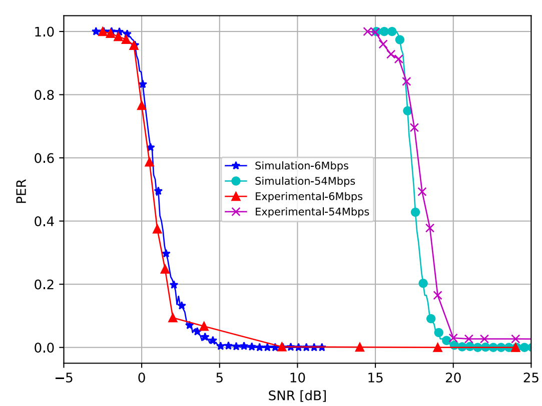

Figure 3. Experimental and simulated PER–SNR curves after calibration.

As a last observation, we point out that the NIST error model is based on different functions depending on the specific modulation adopted for the transmission, i.e. BPSK, QPSK, 16–QAM and 64–QAM. To ensure readability, in Fig. 3 only two curves have been presented, representative of the first and last modulation schemes, respectively. However, the calibration phase has been conducted for all the modulation schemes, with analogous results. For this reason, it is possible to conclude that this particular calibration of the OMNeT++ NIST error model is rather accurate, allowing to run meaningful simulations providing packet loss performance representative of a real environment.

IV. A NEW RL–BASED RA POLICY BASED ON SNR KNOWLEDGE

The RLRA algorithm addressed in Section II-B has been hence properly modified to take into account the channel behavior by means of the perceived SNR level. This value is used to properly modify States, Actions, and the Rewards given to the agent by the Markov Decision Process.

Specifically, States S_i are defined considering i) the SNR, ii) the chosen Rate and iii) the frame loss rate R_{loss}. The SNR is divided into 6 different regions SNR_L_i, whose width is 5 dB, while the range of R_{loss} is divided in 10 regions $loss\_L_i$, as described in Fig. 4. Rates r_i are indexed, for simplicity, using the Modulation and Coding Scheme notation (MCS_i) from 0 to 7, corresponding to 6 Mbps and 54 Mbps, respectively.

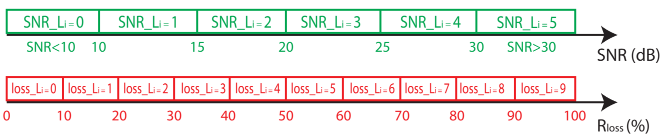

Figure 4. SNR and R_{loss} discretization.

Those three terms are then suitably combined, giving rise to 480 different states S_i. Each state have been then univocally indexed by means of Eq. (3).

$$S_i = loss\_L_i + MCS_i * 10 + SNR\_L_i * 80 \tag{3}$$

Rewards function has been adapted as in Eq. (4) to take into account the SNR level in the RA scheme, in order to provide a better reward to actions that increase the rate when the SNR is high (channel in a good condition), and vice-versa.

$$\begin{aligned} R_{t+1} = \quad & \beta \cdot (-R_{loss_{t+1}} + R_{loss_t}) \\ & + (1-\beta) \cdot \frac{Rate_{t+1} - Rate_t}{Rate_{t+1} + Rate_t} \cdot \frac{SNR_t}{40} - 1; \end{aligned} \tag{4}$$

As far as Actions are concerned, two different algorithms have been developed.

1) RLRA-SNR: This first and simpler algorithm defines 8 different actions A_t cornensponding to the specific rate chosen at the instant t for next step, that is $A_t = 0$ (which corresponds to $r_{t+1} = 6$ Mbps) to $A_t = 7$ ($r_{t+1} = 54$ Mbps). RLRA-SNR, hence, does not differentiate between the first transmission attempt and the possible retransmissions. Indeed, each packet retransmission is associated to a new chosen Action.

2) RLRA-SNR-RC: The second algorithm, conversely, aims at providing a comprehensive prediction of the whole frame transmission process including also eventual retransmissions due to bad channel conditions. Specifically, RLRA-SNR-RC determines two different rates to be used for the packet transmission, namely r_t and r_{t+1} with $r_{t+1} \le r_t$, and performs n_t and n_{t+1} retransmission attempts with the first and the second rate, respectively. The algorithm further defines a maximum number of transmission attempts n_{max}, performing the remaining $n_{max} - n_t - n_{t+1}$ retransmissions at the minimum available rate. All the possible Retransmission Chains (RC) resulting from the aforementioned parameters can be clearly computed in advance to avoid any computation load at runtime.

V. ASSESSMENT OF THE PROPOSED ALGORITHM

A. Implementation details

The two proposed RA algorithms, namely RLRA-SNR and RLRA-SNR-RC, have been implemented within the IEEE 802.11 Data-Link Layer model of OMNet++.

As described above, RLRA-SNR-RC depends on a set of parameters, whose choice impacts on its behavior. Firstly, with the 8 available trasmission speeds of IEEE 802.11g and the constraint $r_{t+1} \le r_t$ we clearly obtain 28 different Actions.

A RC is then associated to each action. To this purpose, in our simulation assessment we fixed $n_{max} = 8$, and imposed $n_t = 3$ for r_t at the higher rates ($MCS = 6{,}7$), and $n_{t+1} = 2$ for r_{t+1} at the lower $MCS = 1{,}2{,}3{,}4{,}5$. With the proposed choice of parameters the algorithm tends to trade-off between the throughput maximization and the reliability. Although the analysis of different settings, for instance more conservative choices to maximize reliability, is definitely an important topic,

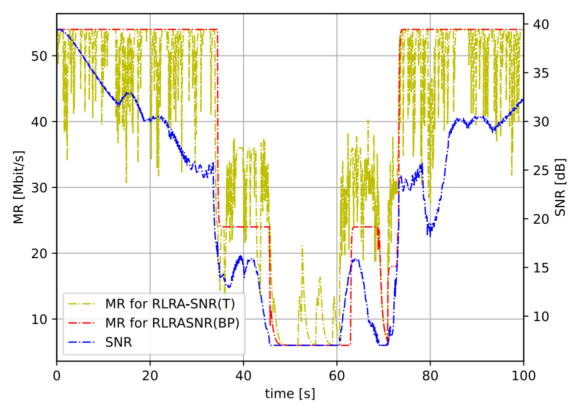

Figure 5. SNR vs Chosen Rate for RLRA–SNR, $\beta = 0.45$.

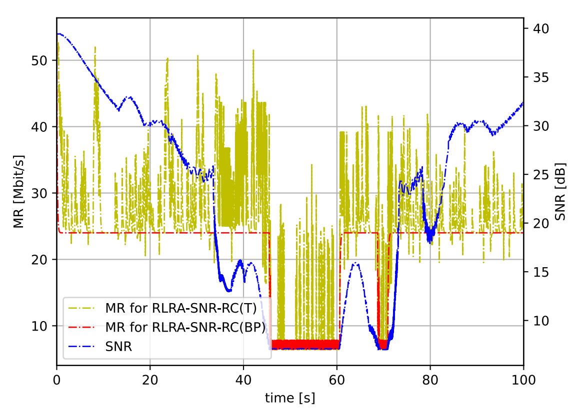

Figure 6. SNR vs Chosen Rate for RLRA–SNR–RC, $\beta = 0.45$.

for reason of space we limit here to present the results obtained with the aforementioned values.

To provide an immediate picture of the two algorithms behavior with respect to different channel conditions, we can observe Fig. 5 and Fig. 6, relevant to RLRA-SNR and RLRA-SNR-RC, respectively. The figures shows the pattern of the chosen transmission rates for both the initial training phase (indicated with T) and after the best policy (BP) has been defined. For representation purposes, given the high number of samples, we applied a moving average filtering on data, namely Mean Rate (MR). The former shows that, as expected, higher rates are selected when the channel is in a good state (high SNR values) and vice-versa. The latter Fig. 6 conversely highlights how the appropriate structure of the RLRA-SNR-RC algorithm yields to globally lower rates allowing to achieve an increased reliability in the communication. As a final observation, the curves relevant to the training phase are characterized by a rather unsteady behavior, since all the possible actions and states need to be tested.

B. Simulations Outcomes

We set up a simulation where a total of 100.000 packets are exchanged between the two stations in Fig. 2, and each packet carries a payload of 50 Bytes, a typical length for time-critical real-time measurement systems. The two nodes moves within the environment: in the first period of time they are in plain line of sight, with a good channel status, whereas in the second part, an obstacle starts hindering the line of sight path, increasing the path loss and worsening the SNR. The performance index considered in this study is the end-to-end delay, a typical indicator for time-critical or real-time networks. The assessment of the two proposed algorithms is carried out as a comparison of their performance with those obtained by other known RA startegies. In particular, in this preliminar analysis we will consider:

1) RSIN (Section II-C);
2) RL–based RA algorithm (RLRA, Section II-B);
3) RL–based RA algorithm with SNR (RLRA-SNR, Section IV-1);
4) RL–based RA algorithm with SNR and Retransmission Chains (RLRA-SNR-RC, Section IV-2).

The outcomes obtained from a first set of simulations have been reported in terms of experimental cumulative distribution function (ECDF) of the end-to-end delay in Fig. 7. A more in-depth analysis of the outcomes in terms of number of received packets, R_{loss} and end-to-end delay statistics can be found in Table I.

The analysis of Fig. 7 allows to observe that, as already expected from the previous Fig. 6, RLRA-SNR-RC has a more conservative behavior and tends to chose lower rates with respect to both RSIN and RLRA-SNR, hence providing slightly higher end-to-end delays. Moreover, this algorithm also presents a steeper curve, indicating that it is able to settle at the suitable rate faster than the other algorithms.

Table I provides some more insights. Importantly, all the algorithms adopting a measurement of the SNR are able, as expected, to perform better than RLRA, which instead learns from the past transmission history. This applies both in terms of average end–to–end delay and standard deviation, and allows to conclude that the accurate knowledge of the channel status enables more appropriate decisions on the transmission rate yielding to an improved determinism. Another significant result is that both RLRA–SNR and RLRA–SNR–RC performs better than RSIN, indicating their ability to find a better trade off between the reliability (i.e.minimization of the loss) and the use of high transmission rates. Moreover, RLRA–SNR and RLRA–SNR–RC provides rather similar performance, with RLRA–SNR showing a slightly lower average and standard deviation than RLRA–SNR–RC, mostly thanks to the higher adopted rates.

Table I
SIMULATION RESULTS

Algorithm	Received packets	R_{loss} (%)	Delay (ms) Mean	Delay (ms) Std. Dev.
RSIN	82858	17,142	8,407	93,511
RLRA	78835	21,165	15,327	98,162
RLRA–SNR	83494	16,506	4,106	87,272
RLRA–SNR–RC	83426	16,574	4,338	90,944
RLRA–SNR (T)	80868	19,132	7,536	90,528
RLRA–SNR–RC (T)	82472	17,528	8,887	93,340

Finally, the last two rows of Table I reports the network performance during the training phase of both the proposed algorithms. As expected, the performance during this phase are generally worse than those obtained using the Best Policy π. Clearly, this turns out to indicate that the training activity of our RL algorithms, necessary to the define the final best policy, has been effective.

Another interesting outcome from this simulation study is relevant to the average throughput that the different algorithms allow, which is represented in Fig. 8. While both RSIN and the proposed RL-based algorithms are all able to face effectively the channel impairments due to the obstacle, it can

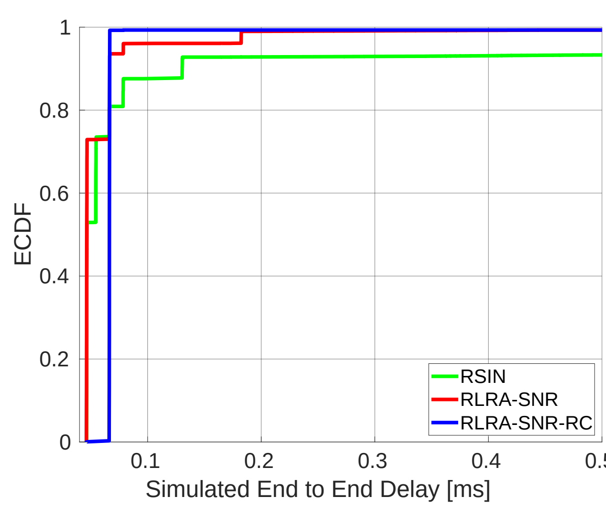

Figure 7. Experimental Cumulative Distribution Function for the E2E Delay.

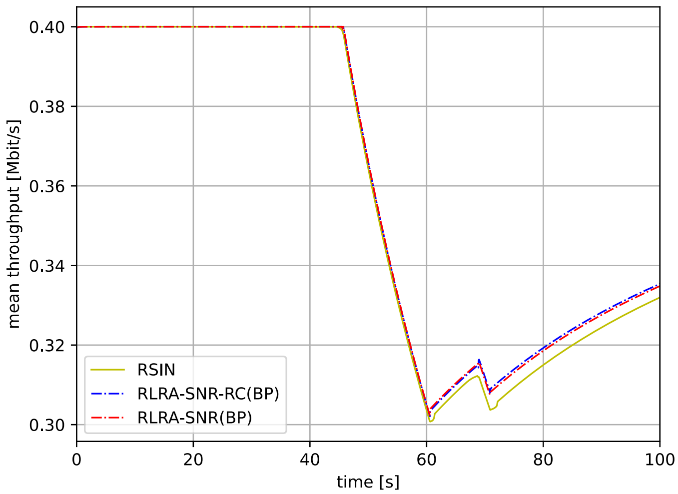

Figure 8. Comparison of the Mean Throughput of different RA techniques.

be observed that RLRA–SNR and RLRA–SNR–RC perform slightly better in terms of throughput. This means that, given the small payloads used in the considered scenario, the RL-based algorithms experience a lower number of packet losses.

Finally, this preliminary study allowed to draw some further conclusions. On the one hand, the computational effort, at run time, required by the proposed algorithms is low compared to that of RSIN, since the latter need to periodically run an optimization problem and update the LUT while the RL–based algorithms directly use the Best Policy. On the other hand, the proposed algorithms are intrinsically able to handle payload variations, whereas RSIN, in those cases, need both an updated PER–SNR map for the new payload length and to run the optimization problem anew.

VI. Conclusions and Future Directions of Research

In this article, we presented a novel rate adaptation strategy for Wi–Fi based on a reinforcement learning approach, and targeted to the need of real-time measurements in industrial environments. This RA algorithm is based on a reinforcement learning approach, where the agent learns how to perform an effective RA through channel sensing, by means of SNR measures. Two different versions of the algorithm have been proposed, that differ for the type of actions taken at each specific state, namely RLRA–SNR and RLRA–SNR–RC. The latter, in particular, is based on the definition of a Retransmission Chain, hence specifically managing the retransmissions.

The performance assessment has been carried out by means of simulations, exploiting the widespread OMNeT++ simulator. Nevertheless, to provide meaningful outcomes, representative of realistic situations, we have proposed a calibration phase for the simulation models, exploiting experimental data where the channel behavior has been measured.

Simulation results are encouraging, since on the one hand the RL-based approach revealed effective for RA purposes and, on the other hand, the exploitation of SNR measurements ensures to better adapt to channel conditions, resulting in lower end-to-end delay average and standard deviation.

Given the very encouraging results, the current preliminary work opens up to several future analysis. In particular, the RL–based algorithms should be implemented and assessed in a wider and more complex network setup to tests the algorithms capabilities in critical working conditions. Moreover, an extensive experimental campaign needs to be deployed, to validate the proposed solutions on a real experimental setup, and to better estimate their computational overhead.

References

[1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, and M. Gidlund, “Industrial internet of things: Challenges, opportunities, and directions,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4724–4734, Nov 2018.

[2] B. Ooi and S. Shirmohammadi, “The potential of iot for instrumentation and measurement,” *IEEE Instrumentation Measurement Magazine*, vol. 23, no. 3, pp. 21–26, 2020.

[3] S. R. Pokhrel and S. Singh, “Compound-tcp performance for industry 4.0 wifi: A cognitive federated learning approach,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[4] L. Skrzypczak, D. Grimaldi, and R. Rak, “Analysis of the different wireless transmission technologies in distributed measurement systems,” in *Proc. IDAACS*, Rende, Italy, 2009, pp. 673–678.

[5] M. Luvisotto, Z. Pang, and D. Dzung, “Ultra high performance wireless control for critical applications: Challenges and directions,” *IEEE Transactions on Industrial Informatics*, vol. 13, no. 3, pp. 1448–1459, 2016.

[6] T. Huang, S. Li, and S. Gao, “Raca: A joint rate and channel adaptation scheme for dense 802.11n networks,” *Procedia Computer Science*, vol. 111, pp. 183 – 189, 2017, the 8th International Conference on Advances in Information Technology. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050917311997

[7] F. A. Setia, T. Y. Arif, and R. Munadi, “Collision-aware rate adaptation algorithm for high-throughput ieee 802.11 n wlans,” in *2018 6th International Conference on Information and Communication Technology (ICoICT)*. IEEE, 2018, pp. 12–17.

[8] F. Wu, W. Yang, J. Ren, F. Lyu, P. Yang, Y. Zhang, and X. Shen, “Ndn-mmra: Multi-stage multicast rate adaptation in named data networking wlan,” *IEEE Transactions on Multimedia*, 2020.

[9] S. Khastoo, T. Brecht, and 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*, 2020, pp. 161–170.

[10] M. Luvisotto, F. Tramarin, and S. Vitturi, “A learning algorithm for rate selection in real–time wireless lans,” *Computer Networks*, vol. 126, 07 2017.

[11] G. Peserico, T. Fedullo, A. Morato, S. Vitturi, and F. Tramarin, “Rate adaptation by reinforcement learning for wi-fi industrial networks,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, vol. 1, 2020, pp. 1139–1142.

[12] N. C. Luong, D. T. Hoang, S. Gong, D. Niyato, P. Wang, Y. Liang, and D. I. Kim, “Applications of deep reinforcement learning in communications and networking: A survey,” *IEEE Communications Surveys Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[13] T. Fedullo, F. Tramarin, and S. Vitturi, “The impact of rate adaptation algorithms on wi-fi-based factory automation systems.” *Sensors*, vol. 20, no. 5195, 2020.

[14] E. Tanghe, W. Joseph, L. Verloock, L. Martens, H. Capoen, K. V. Herwegen, and W. Vantomme, “The industrial indoor channel: Large–scale and temporal fading at 900, 2400 and 5200 MHz,” *IEEE Transactions on Wireless Communications*, vol. 7, no. 7, July 2008.

[15] L. E. Miller, “Validation of 802.11a/uwb coexistence simulation,” NIST, NIST Pubs. [Online]. Available: https://doi.org/10.6028/NIST.WCTG.10-17-2003

[16] F. Tramarin, S. Vitturi, M. Luvisotto, and A. Zanella, “On the use of ieee 802.11n for industrial communications,” *IEEE Transactions on Industrial Informatics*, vol. 12, no. 5, pp. 1877–1886, 2016.