A Cognitive Mechanism for Rate Adaptation in Wireless Networks

Luciano Chaves[1⋆], Neumar Malheiros[1], Edmundo Madeira[1], Islene Garcia[1], and Dzmitry Kliazovich[2]

[1] Institute of Computing – University of Campinas
Av. Albert Einstein 1271, 13083-970, Campinas, Brazil
luciano@lrc.ic.unicamp.br,
{ncm, edmundo, islene}@ic.unicamp.br
[2] DISI – University of Trento
Via Sommarive 14, I-38050, Trento, Italy
kliazovich@disi.unitn.it

Abstract. Sophisticated wireless interfaces support multiple transmission data rates and the selection of the optimal data rate has a critical impact on the overall network performance. Proper rate adaptation requires dynamically adjusting data rate based on current channel conditions. Despite several rate adaptation algorithms have been proposed in the literature, there are still challenging issues related to this problem. The main limitations of current solutions are concerned with how to estimate channel quality to appropriately adjust the rate. In this context, we propose a Cognitive Rate Adaptation mechanism for wireless networks. This mechanism includes a distributed self-configuration algorithm in which the selection of data rate is based on past experience. The proposed approach can react to changes in channel conditions and converge to the optimal data rate, while allowing a fair channel usage among network nodes. Simulation results obtained underline performance benefits with respect to existing rate adaptation algorithms.

Key words: Wireless Networks, Rate Adaptation, Self-configuration, Self-optimization, Cognitive Algorithms.

1 Introduction

Advanced wireless technologies have emerged as the key building blocks in designing the broadband access network architectures of the future Internet. In particular, the development of sophisticated modulation schemes was a main contribution to provide high performance wireless networks. The modulation process consists in translating a data stream into a suitable form for transmission on the physical medium. Higher data rates are commonly achieved by using modulation schemes that efficiently take advantage of good channel conditions.

⋆ The authors would like to thank FAPESP (process numbers 2006/50512-4 and 2008/07770-8) and IBM Research for supporting this work.

However, these schemes are more sensitive to medium quality degradation and do not perform well for long range transmissions. On the other hand, the use of robust modulation schemes leads to more resilient connections, but it results in lower data rates due to redundancy and control information overhead.

Several modulation schemes were defined in wireless technology standards to deal with unstable channel conditions. For instance, IEEE 802.11 interfaces support multiple modulation schemes, each one commonly referred for its nominal achievable data rate. Although, that standard does not specify how to dynamically select an appropriate modulation scheme (data rate) for current conditions in order to optimize network performance. This challenging issue is called rate adaptation. The absence of a standard solution motivated the proposal of several rate adaptation algorithms.

The rate adaptation process can be divided into two phases: assessing channel conditions, and then selecting the most appropriate rate based on channel quality information. Existing rate adaptation algorithms can be classified as statistics-based or signal-strength-based accordingly to their approach to measure channel conditions [1]. Statistics-based solutions infer channel quality from statistical information about frame transmissions, like frame error rate and number of retransmissions. Signal-strength-based solutions rely on wireless signal measurements as channel condition indicators. In this case, common parameters are the signal-to-noise ratio (SNR) and the received signal strength indicator.

Several rate adaptation mechanisms have been proposed and even some of them are widely used. However, current solutions still face some limitations. Signal-strength-based mechanisms suffer from the lack of a strong correlation between SNR indicator and the delivery probability at a given data rate. In addition, data rate configuration is performed at the sender node, while the signal information is available at the receiver, which leads to communication overhead [2]. These factors limit effectiveness of this approach in practice [3]. Statistics-based mechanisms are affected by the difficulty of finding proper thresholds for the selection of optimal data rate. Also, such mechanisms present long convergence time due to the use of statistical information, which leads to performance degradation on such dynamic environments [4].

In order to tackle these problems, we propose a cognitive approach for rate adaptation in wireless networks. The proposed solution, called COgnitive Rate Adaptation (CORA), is a distributed mechanism which enhances the network element with self-configuration functionality to dynamically adapt the data rate. CORA implements a cognitive algorithm to decide on the optimal rate based on a knowledge information base. It is able to quickly react to changes on channel conditions in order to avoid performance degradation, and can also ensure fair resource sharing among nodes.

The rest of this paper is organized as follows. Section 2 summarizes related work. Section 3 presents the core of the proposed rate adaptation mechanism. Section 4 aims at performance evaluation and comparison with typical rate adaptation mechanisms. Finally, Section 5 concludes the paper outlining directions for future work on the topic.

2 Related Work

One of the first solutions presented for the rate adaptation problem was the Auto Rate Fallback (ARF) algorithm, proposed by Kamerman and Monteban [5]. ARF defines fixed thresholds to increase or decrease outgoing data rate accordingly to the number of successes or failures on consecutive transmission attempts, respectively. It is a simple solution which attempts to use the highest effective data rate at each moment. However, it suffers from instability. Even when ARF reaches the optimal data rate, it keeps trying to increase the rate after a specific number of successful transmissions occurs. An improvement over ARF was proposed by Lacage *et al.* [6]. Their algorithm, called Adaptive Auto Rate Fallback (AARF), makes use of a binary exponential back off mechanism to dynamically adapt increase and decrease thresholds in order to mitigate instability resulting from unnecessary changes on data rate. Both solutions adjust the rate only to neighboring values. Also, they use the frame error rate (FER) to estimate channel quality, but they are not able to identify the cause of frame losses. As a result, they reduce the rate when a collision occurs, which is not necessary [7].

Another relevant solution is the SampleRate algorithm proposed by Bicket [3]. It periodically sends frames at data rates other than the current one to estimate the expected per-frame transmission time for each available rate. Then, it adjusts the rate to the best known one. The limitation of this approach is that it is based on a small number of probing packets, which can be misleading and trigger incorrect rate changes. It also may lead to long convergence time.

Using a complete signal-strength-based solution, the Receiver-Based Auto Rate (RBAR) algorithm [8] gets a feedback on channel quality from the receiver node to determine the optimal rate at the sender. That algorithm requires the use of the Request to Send/Clear to Send (RTS/CTS) mechanism[3] to obtain channel quality information from the destination node. This solution is not affected by collision problems and can perform well even with highly unstable channel conditions. However, the use of SNR information may lead to a problem due to the complexity of measuring it and mapping the measured value onto a specific rate [1, 3]. In addition, this solution requires changes in the 802.11 protocol and introduces the RTS/CTS overhead.

The Collision-Aware Rate Adaptation (CARA) algorithm, proposed by Kim *et al.* [7], uses statistical information instead of SNR, as well it is able to differentiate frame losses. It works similarly to ARF, increasing or decreasing data rate due to consecutive successes or losses in frame transmissions, respectively. Moreover, it can use the RTS/CTS mechanism to identify the cause of frame losses. In the case that collision is the cause of frame loss, the algorithm can prevent unnecessary decrease of date rate. The rate should be decreased only if frame losses result from channel condition degradation. This approach reduces misleading channel quality information due to collisions, but introduces over-

[3] The RTS/CTS mechanism can be used in 802.11 networks to reduce frame collisions when accessing the shared wireless medium.

head and can lead to instability by alternating between the use or not of the RTS/CTS mechanism.

The Cross-layer Rate Adaptation (CLRA), proposed by Khan *et al.* [9], considers application preferences and time constraints when deciding on the best data rate. This algorithm aims at selecting the lowest rate that can effectively meet required traffic demand of running applications. The use of equations can straightforward guide this rate selection. The problem comes when the medium allows the use of a higher rate and this algorithm selects a lower one. This behavior induces the performance anomaly problem discussed in [10]. That is, if there is at least one host with a lower rate, than the throughput of all hosts transmitting at the higher rate is degraded below the level of this lower rate.

Limitations of the aforementioned solutions are related with supporting loss differentiation to avoid unnecessary rate adjustments, avoiding complex or not realistic channel quality metrics, and preventing long convergence time. All these factors were considered in the design of the proposed cognitive rate adaptation mechanism. CORA does not consider frame loss ratio to guide rate selection. Consequently, there is no need for loss differentiation. Also, the proposed algorithm does not depend on signal strength indicators like SNR. As detailed later, CORA is built upon a cognitive algorithm that selects the optimal rate accordingly to evidenced past experience. As a result, it is able to quickly react to changes and adjust the data rate for current network conditions avoiding performance degradation.

3 Proposed Solution

In this work, we propose a cognitive mechanism for data rate adaptation in wireless networks. The proposed mechanism is built upon CogProt, a framework for cognitive configuration and optimization of communication protocols. In next subsection, we briefly describe the CogProt framework, and then we present a detailed description of the cognitive rate adaptation mechanism and its setup.

3.1 The CogProt Framework

The CogProt framework was developed considering the concept of Cognitive Networks, which has emerged as a promising paradigm to deal with performance degradation resulting from changing network conditions. Such paradigm relies on cognitive algorithms to provide dynamic reconfiguration of communication systems, through learning and reasoning, in order to optimize system-wide performance [11]. CogProt aims at implementing this concept by periodically reconfiguring protocol parameters based on recent past experience. As a result, the framework enables network nodes to dynamically adjust their configuration to avoid performance degradation. In order to do this, CogProt introduces a cross-layer cognitive plane as illustrated in Fig. 1. This plane is responsible for optimizing protocol stack parameters at different layers. Such optimization process is performed by several quality feedback loops, one for each parameter of interest.

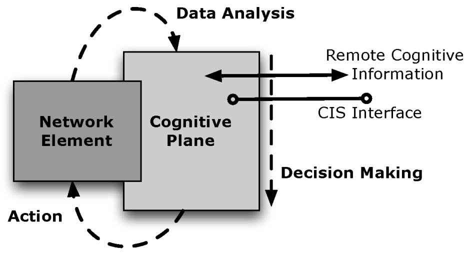


Fig. 1. Overview of the CogProt framework.

The quality feedback loop consists in monitoring the performance of a parameter P and then enforcing reconfiguration actions to converge P to optimal operational point. This optimization process consists of three phases: data analysis, decision making, and action. During the *data analysis* phase, the cognitive plane collects performance information about the current value of the parameter P, according to a specific quality metric. Moreover, a local knowledge base is built from performance information collected at each iteration of the feedback loop in order to assist the *decision making* phase. During this second phase, the cognitive plane selects the value of P that provides the best performance, and assigns it to the mean of a random number generator that follows a normal distribution. This selection relies not only on the local cognitive information base, but also on (remote) cognitive information shared among network nodes and configuration policies for the network segment obtained from a local Cognitive Information Service (CIS). The *action phase* consists of assigning a new value to P. This value is obtained from the random number generator in the operation range of parameter $P \in [P_{min}, P_{max}]$.

This optimization process is repeated at each interval I, and continuously adjusts the mean of the normal distribution to the value of P that provides the best performance according to the performance information base. As a result, the mean of the normal distribution converges to the optimal value for P. Consequently, most of the randomly chosen values for P are optimal under current network conditions. Meanwhile, the mechanism will eventually test neighboring values (close to the mean), allowing adjustment of the parameter in face of changes on network conditions.

3.2 Cognitive Rate Adaptation

Since we have already described the CogProt framework, let us now revisit the rate adaptation problem. The proposed Cognitive Rate Adaptation mechanism (CORA) is an instance of the CogProt framework, designed to optimize the MAC layer performance by dynamically adjusting the transmission data rate. In this case, the parameter of interest is the MAC data rate (R) and the associated quality metric is the MAC throughput. CORA implements a single quality feedback loop into the MAC layer, as illustrated in Fig. 2. Each phase of this self-optimization process is explained as following.

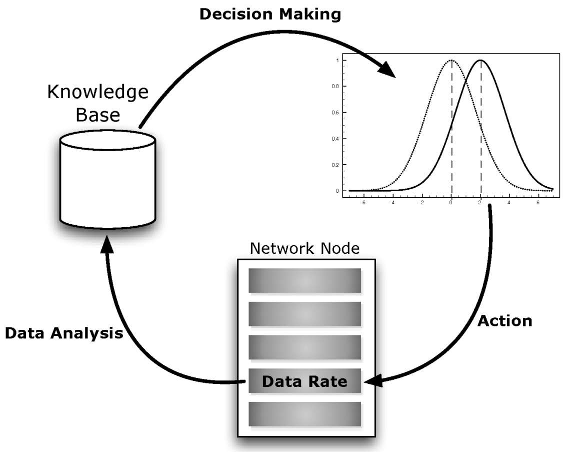


Fig. 2. Overview of the cognitive rate adaptation mechanism.

Data Analysis: Performance information on each value $R_i \in [R_{min}, R_{max}]$ is maintained in the local knowledge base. Let R_c be the current value for data rate. In this phase, the mechanism measures the performance P_c obtained from using R_c in terms of MAC throughput. Then the information is averaged with an exponentially weighted moving average (EWMA) as follows:

$$P_i = (1 - w) * P_i + (w) * P_c \tag{1}$$

where P_i is the stored performance information for R_i, w is the weight assigned to immediate performance for R_c and $c = i$. That is, the measured performance for the current data rate is used to calculate the average performance for this data rate. The local knowledge base reflects the performance history for each possible value. The weight w can be used to control the relevance of recent performance information. High w values increase the effect of immediate performance information on the average performance, which enable the algorithm to quickly react in face of changes on channel conditions. Small w values lead to a more conservative behavior which results in system stability because, in this way, irrelevant transient states do not affect the convergence to the optimal operational point.

Decision Making: In this phase the mechanism decides on the optimal date rate. The algorithm looks the knowledge base up for the data rate R_i that provides the best performance. The corresponding data rate is assigned to the mean of the normal distribution. CORA algorithm does not make use of remote cognitive information or configuration policies from the CIS. The decision make process is decentralized since it relies only on information available at the network element itself, and does not require any communication with other nodes.

Action: A new random $R_i \in [R_{min}, R_{max}]$ is generated according to the normal distribution, and assigned to the MAC data rate. The standard deviation

(STD) for the normal distribution defines the aggressiveness of the mechanism. The lower the STD, the more conservative is the behavior of the algorithm in trying new values for the data rate. Therefore, this parameter directly affects the convergence time and system stability.

This quality feedback loop is performed by each element at the end of a sample interval I. Several works discuss this timing consideration for channel quality measurement [1, 3, 6]. All of them argue that, to properly perform rate adaptation, the correlation between channels errors must be at least of the order of the sampling interval.

It is important to clarify that CORA mechanism is executed by each network element in an independent way. Besides, each element has its own local knowledge base. These features characterize a completely decentralized system, without the need of a coordinator element. Furthermore, it is feasible that elements enabled with CORA work together with elements that run/perform another rate adaptation algorithm (or none). This is possible because CORA demands neither information exchange among network nodes nor changes on the MAC protocol.

4 Performance Evaluation

In order to evaluate performance, the *Network Simulator (ns-2)*[4] was extended with CORA functionality. The performance of the proposed solution was compared against typical rate adaptation mechanisms, namely, Auto Rate Fallback (ARF) and Receiver-Based Auto Rate (RBAR). Wireless network used in experiments is configured according to IEEE 802.11g standard, using the Free Space propagation model and operating with the Destination-Sequenced Distance Vector (DSDV) routing protocol. Control frames are sent at the basic date rate (6 Mbps) and the frame size is 1500 bytes. The simulation time is 1200 seconds. The results are the average from at least 10 iterations, using a 95% confidence interval.

There are 8 available modulation schemes in IEEE 802.11g corresponding to the following nominal data rates: 6, 9, 12, 18, 24, 36, 48 and 54 Mbps. We have mapped an integer index $i \in [0, 7]$ to each available data rate $R_i \in [R_{min}, R_{max}]$. In addition, a "performance table" is implemented to store the MAC throughput information for each R_i. At start-up, the "performance table" is empty and the data rate is set to 6 Mbps as a conservative approach. At the end of each interval I, the algorithm stores performance information for the current data rate, selects the index i that corresponds to the rate R_i with the highest MAC-layer performance and then sets the mean of the normal distribution to i.

As discussed in the previous section, there are three control parameters for the cognitive mechanism: I, w, and STD. We performed simulations using different combinations with these parameters, in order to select the values that provide best results. We simulated sampling intervals (I) ranging from 10 ms

[4] Available at http://www.isi.edu/nsnam/ns

up to 8 s. In all scenarios, values close to 100 ms provided the best performance results. We evaluated w values ranging from 0 up to 0.9. The value 0.9 provided the best performance in terms of convergence time because it assure a higher weight to immediate past performance information. As a result, CORA is able to quickly reacts to changes on channel quality. In addition, we evaluated STD values from 0.2 up to 1.5 and the value 0.3 provided the best performance. Lower STD values allowed the algorithm to choose the optimal rate with a higher frequency (in this case, near 90%) which resulted in better performance.

4.1 Simulation Results

Scenario 1: Single Flow. In this scenario, we consider two nodes: a stationary node S and a mobile node M, as illustrated in Fig. 3. There is a single flow from S to M, with a constant bit rate (CBR) traffic source of 20 Mbps over UDP protocol. Node M starts moving away from S with constant speed of 0.25 m/s, until it reaches a distance of 300 m between them.

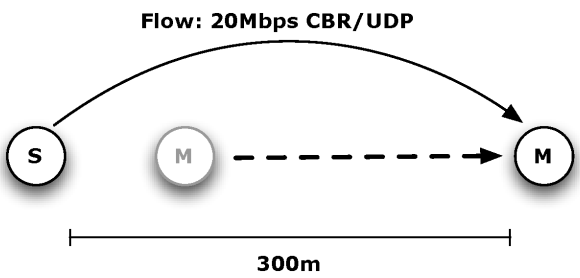


Fig. 3. Simulated topology for Scenario 1.

First of all, we evaluate the performance of each available modulation scheme. Figure 4 shows throughput as a function of the distance between nodes. Each modulation scheme is referred for its nominal rate (from Mod6 to Mod54). The presented throughput accounts for protocol overhead and frame losses. Lower data rates have greater transmission ranges. As the range increases, the signal attenuates due to fading and path loss. At different distances, there is a data rate that provides the best throughput. In this case, it is necessary to select the appropriate available data rate to maintain the highest possible throughput.

Then, we compared the throughput achieved by the three rate adaptation algorithms: CORA, ARF, and RBAR. The results are presented in Fig. 5. We can see that CORA performs better in converging to the best rate at each distance from the sender. ARF periodically tries to send frames at the next higher rate, inducing unnecessary adaptations (and consequently, losses). The RBAR performance is penalized by the use of RTS/CTS mechanism in a scenario free of collisions.

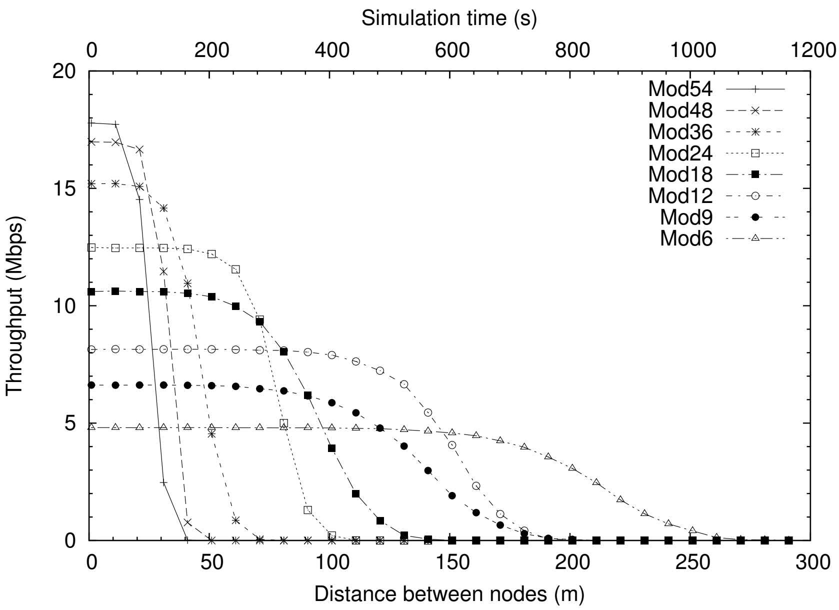


Fig. 4. Throughput over distance for 802.11g fixed data rates.

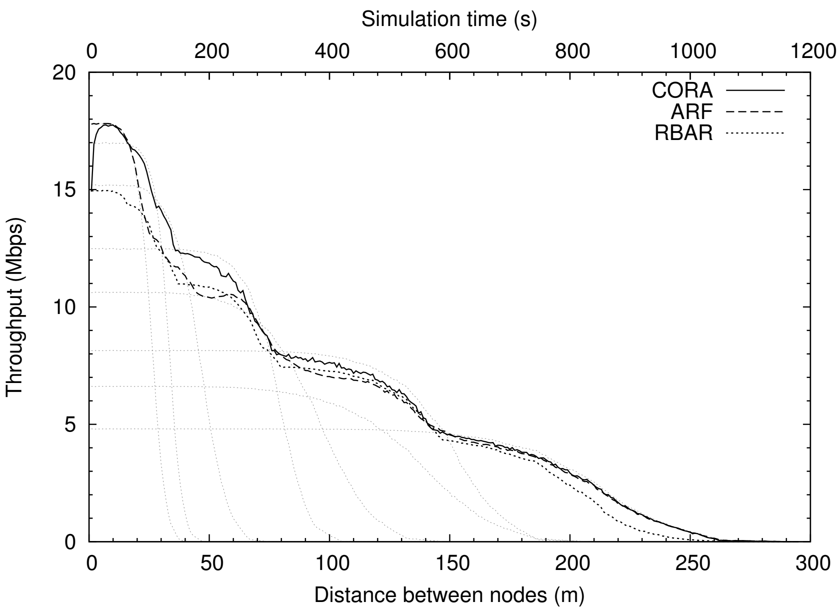


Fig. 5. Throughput over distance for each rate adaptation algorithm.

The average throughput for each algorithm and each fixed data rate is presented in Fig. 6. There is also a “maximum” value that represents the theoretical best achievable performance when selecting always the best rate. CORA reaches up to 96% of this maximum value, outperforming ARF by 4.5% and RBAR by 11.2%.

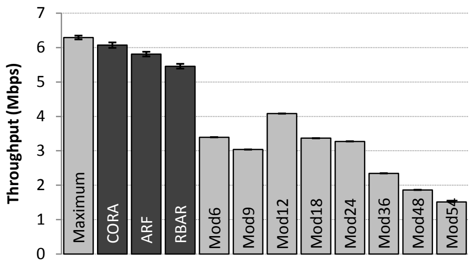
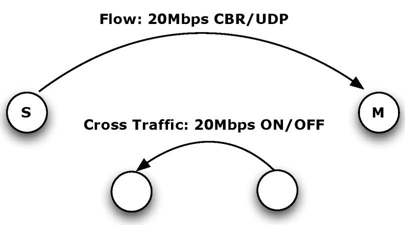
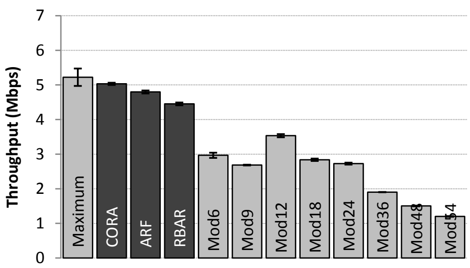


Fig. 6. Average throughput for each algorithm and fixed data rates in Scenario 1.

Scenario 2: with Cross-Traffic. This scenario extends the first one by adding two nodes generating cross-traffic, which leads to collision and interference. Cross-traffic nodes are stationary, fixed at 10 meters between each other. There is a single flow between them with an intermittent CBR traffic source of 20 Mbps. Figure 7(a) illustrates this topology. Also in this scenario, CORA provides better performance than ARF and RBAR, as presented in Fig. 7(b). Collisions leaded to unnecessary rate decreases by ARF. The use of RTS/CTS mechanism in RBAR contributed to its poor performance. CORA reaches up to 96.5% of the maximum achievable value, outperforming ARF by 4.9% and RBAR by 13.1%. The fact that the cross-traffic source starts and stops repeatedly confirms the ability of CORA to quickly react to changing channel conditions.



(a) Simulated topology



(b) Average throughput

Fig. 7. Simulated topology and average throughput of each algorithm in Scenario 2.

Scenario 3: Multiple Destinations. CORA has improved support for rate adaptation involving multiple destinations. In order to evaluate that, we consider in this scenario an infra-structured wireless network with a single Access Point (AP) sending data to several stationary client nodes at different distances, as shown in Fig. 8. Throughput performance was measured considering two different topologies. The first one consists of two client nodes: one located close enough to the AP such that the highest available data rate is the optimal one, while the other client is far enough such that the lowest data rate is the best one. The second topology includes seven client nodes located at increasing distances from the AP. In this case, the AP must converge to different optimal data rates when sending frames to each destination node. There are simultaneous flows from the AP to each client node using FTP (over TCP) and CBR (over UDP) traffic sources. In this scenario, CORA outperforms ARF and RBAR for up to 22%, as we can see in Fig. 9.

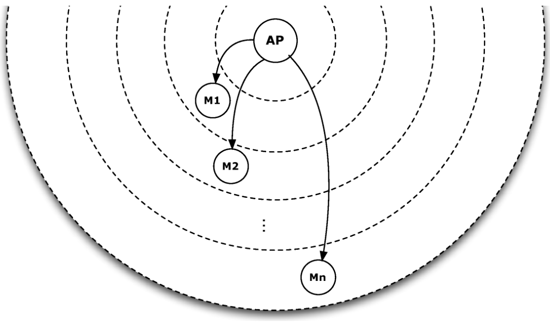
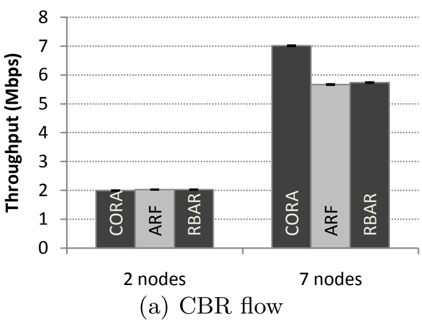
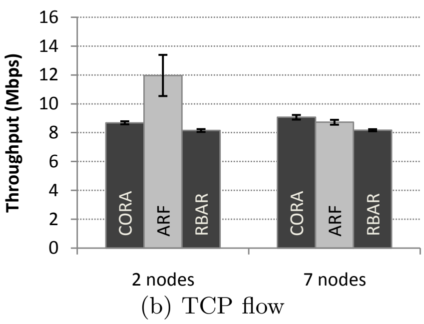


Fig. 8. Simulated topology for Scenario 3.



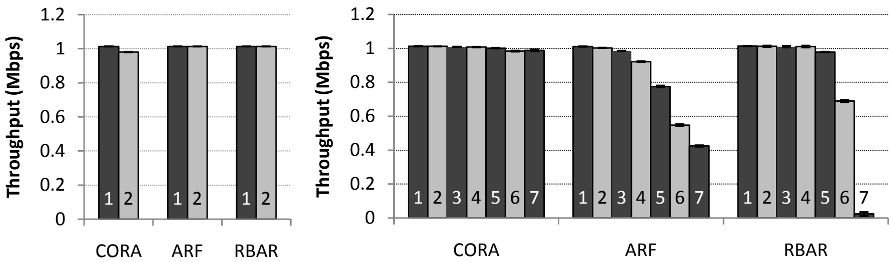
(a) CBR flow



(b) TCP flow

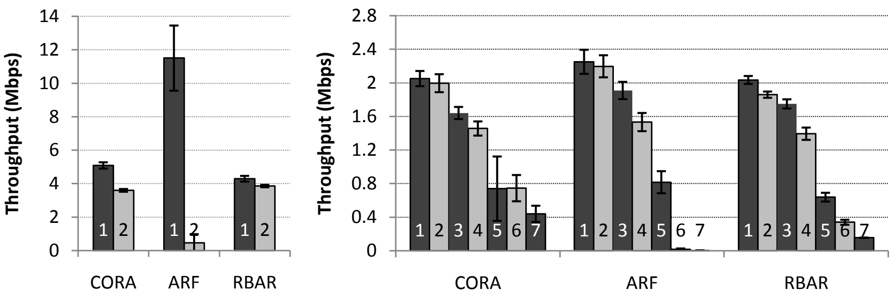
Fig. 9. Average network throughput in Scenario 3.

Figure 9(a) shows the average network throughput when using CBR flows of 1 Mbps for both topologies, with 2 and 7 client nodes, respectively. A two-client topology requires low bandwidth capacity, and all three algorithms are able to delivery frames with practically no losses. However, when the number of nodes increases, CORA shows the best performance. In the same way, Fig. 9(b) shows the average throughput considering FTP flows in both topologies. In this case, ARF provides the best performance for the 2-node topology but at the cost of unfair resource sharing as we can see in Fig. 10 and Fig. 11.



(a) 2 nodes (b) 7 nodes

Fig. 10. Average flow throughput for CBR flows in topologies with 2 and 7 nodes.



(a) 2 nodes (b) 7 nodes

Fig. 11. Average flow throughput for FTP flows in topologies with 2 and 7 nodes.

Figure 10 shows the average flow throughput in both topologies considering CBR traffic. In this scenario, node 1 is the closest to the AP. With increased number of nodes, we can see that CORA is able to provide system-wide fairness. ARF and RBAR penalize nodes far away from the AP. The performance degradation for distant nodes may be gradual, as observed for ARF, or drastic, like in the case of RBAR for nodes 6 and 7.

In the case of TCP traffic, the problem of unfair resource sharing is even more critical because the high frame error rate for distant nodes caused drastic reductions on the TCP congestion window. Figure 11 presents the average flow throughput for TCP flows in both topologies. As showed before, ARF performs better for TCP traffic in the 2-node topology, but it lead to unfair bandwidth sharing as we can see in Fig. 11(a). In the case of the 7-node topology, CORA provides the best performance which is demonstrated in Fig. 11(b). CORA is the solution that provides the higher throughput for distant nodes, improving overall performance by 3.9% and 10.9% if compared with ARF and RBAR, respectively.

5 Conclusions and Future Work

In this paper, we have proposed a cognitive rate adaptation mechanism (CORA) for wireless networks. Such mechanism is decentralized and can be implemented even on devices with limited resources due to its low processing requirement. Moreover, it does not require any changes of IEEE 802.11 MAC protocol. It is also incrementally deployable, since CORA-enabled nodes can communicate with those nodes that do not implement the mechanism.

Simulation results demonstrate that CORA is able to dynamically adjust data rate always matching current network conditions. It outperforms commonly rate adaptation algorithms such as ARF and RBAR by up to 22% in specific situations. CORA gets close to the theoretical best achievable performance, and quickly reacts to changing channel conditions. The initialization is very short, and the solution quickly converges to the optimal rate. The proposed solution also is able to optimize rate adaptation even for multiple simultaneous transmissions. CORA improves fairness in bandwidth sharing, allowing communication between distant network nodes.

Future work will focus on a detailed study of how to proper setup CORA control parameters. Moreover, we will adapt the existing implementation to support multi-hop wireless networks. Additional probability distribution functions will be investigated as candidates to replace normal distribution, which may allow improving the efficiency of the cognitive optimization algorithm.

References

1. Ancillotti, E., Bruno, R., Conti, M.: Experimentation and Performance Evaluation of Rate Adaptation Algorithms in Wireless Mesh Networks. In: PE-WASUN '08: Proc. of the 5th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, New York, USA, ACM Press (Oct 2008) 7–14
2. Haratcherev, I., Langendoen, K., Lagendijk, R., Sips, H.: Hybrid Rate Control for IEEE 802.11. In: MobiWac '04: Proc. of the 2nd International Workshop on Mobility Management and Wireless Access Protocols, New York, USA, ACM Press (Oct 2004) 10–18
3. Bicket, J.: Bit-rate Selection in Wireless Networks. Master's thesis, Massachusetts Institute of Technology (MIT), Department of Electrical Engineering and Computer Science (Feb 2005)
4. Xia, Q., Hamdi, M.: Smart Sender: A Practical Rate Adaptation Algorithm for Multirate IEEE 802.11 WLANs. IEEE Transactions on Wireless Communications **7**(5) (May 2008) 1764–1775

5. Kamerman, A., Monteban, L.: WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. Bell Labs Technical Journal **2**(3) (Aug 1997) 118–133
6. Lacage, M., Manshaei, M., Turletti, T.: IEEE 802.11 Rate Adaptation: A Practical Approach. In: MSWiM '04: Proc. of the 7th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, New York, USA, ACM Press (Oct 2004) 126–134
7. Kim, J., Kim, S., Choi, S., Qiao, D.: CARA: Collision-Aware Rate Adaptation for IEEE 802.11 WLANs. In: INFOCOM '06: Proc. of the 25th International Conference on Computer Communications, Washington, USA, IEEE Computer Society Press (Apr 2006) 1–11
8. Holland, G., Vaidya, N., Bahl, P.: A Rate-Adaptive MAC Protocol for Multi-Hop Wireless Networks. In: MobiCom '01: Proc. of the 7th International Conference on Mobile Computing and Networking, New York, USA, ACM Press (Jul 2001) 236–251
9. Khan, S., Mahmud, S., Loo, K., Al-Raweshidy, H.: A cross layer rate adaptation solution for IEEE 802.11 networks. Computer Communications **31**(8) (May 2008) 1638–1652
10. Heusse, M., Rousseau, F., Berger-Sabbatel, G., Duda, A.: Performance Anomaly of 802.11b. In: INFOCOM '03: Proc. of the 22th International Conference on Computer Communications. Volume 2., Washington, USA, IEEE Computer Society Press (Apr 2003) 836–843
11. Thomas, R.W., Friend, D.H., Dasilva, L.A., Mackenzie, A.B.: Cognitive networks: adaptation and learning to achieve end-to-end performance objectives. IEEE Communications Magazine **44**(12) (Dec 2006) 51–57