

2009 IEEE 34th Conference on Local Computer Networks (LCN 2009)
Zürich, Switzerland; 20-23 October 2009

Understanding the Role of Multi-Rate Retry Mechanism for Effective Rate Control in 802.11 Wireless LANs

Neda Koçi and Mahesh K. Marina
School of Informatics
The University of Edinburgh, UK

Abstract—We consider the multi-rate retry (MRR) capability provided by current 802.11 implementations and carry out simulation-based study of its impact on performance with state-of-the-art rate control mechanisms in typical indoor wireless LAN scenarios. We find that MRR is more effective in non-congested environments, necessitating the need for a mechanism to differentiate between congested and non-congested situations to better exploit the MRR capability. We also observe that decoupling the long-term rate adaptation algorithm from the MRR mechanism is key to fully realizing the benefits of MRR.

I. INTRODUCTION

The 802.11 (a/b/g) physical layer provides multi-rate capabilities. The overall mechanism responsible for selecting a transmission rate from among multiple available transmission rates at run time is referred to as the *rate control*. Many rate control mechanisms have been proposed (e.g., [1], [2], [3], [4], [5]). Most of these proposals tend to focus on a strategy where multiple (re)transmits of a frame use the same rate even though current 802.11 implementations support the capability to change the rate between frame retransmits. This latter capability is the focus of this paper and henceforth referred to as *multi-rate retry (MRR)*.

The rate control mechanism in 802.11 can be seen to be composed of two interrelated mechanisms: (1) *rate adaptation algorithm (RAA)* responsible for determining the initial transmission rate of a frame and adaptation of that rate over time; (2) *MRR mechanism* responsible for selecting the transmission rate to be used for any frame retransmissions. The rate adaptation algorithm attempts to track the longer term fluctuations of the channel through link layer statistics or physical layer (PHY) based metrics, whereas the MRR mechanism is left to deal with short-term variations of the channel. Even though the MRR capability is implemented by the widely used MadWifi driver [6] and is part of all RAAs provided by the driver, its impact on rate adaptation performance surprisingly has not been studied in detail to date.

In this paper, we conduct a detailed simulation study investigating the impact of MRR with state-of-the-art rate control mechanisms [2], [3], [4] in typical indoor wireless LAN scenarios. Our study shows that the specifics of RAAs and associated MRR mechanism expectedly plays a crucial role in determining the MRR impact. More importantly, we find that MRR is particularly beneficial in non-congested scenarios,

thus motivating the need for a mechanism to distinguish between congested and non-congested situations to fully exploit MRR capability. We also observe that the coupling between RAA and MRR in existing rate control mechanisms limits the effectiveness of MRR.

II. MULTI-RATE RETRY

A number of rate adaptation algorithms (RAAs) have been proposed in recent years. In this study, we consider three well-known RAAs: ARF [1] and its variant AMRR [2]; Onoe [3]; and SampleRate [4]. We refer the reader to [7] for a summary of these algorithms.

Multi-Rate Retry (MRR) is a mechanism first introduced by the MadWifi driver project [6]. This project provides one of the most advanced Linux driver for 802.11a/b/g network interface cards (NICs) based on Atheros chipsets. The driver itself is open source but depends on the proprietary *Hardware Abstraction Layer (HAL)* that is available in binary form only. The driver maintains several FIFO (First In First Out) queues of transmission descriptors to schedule packets for transmission. Each transmission descriptor contains detailed control information related to a frame's transmission. The most important and relevant information contained by the descriptor is an ordered set of four pairs of rate and transmission count fields ($r_0/c_0, r_1/c_1, r_2/c_2, r_3/c_3$) that we refer to as the *multi-rate retry chain*. Whenever the wireless medium is available for transmission, the hardware triggers the transmission of the descriptor located at the head of the FIFO. Originally the frame is transmitted with the rate r_0 specified in the descriptor. If this transmission fails (determined by the absence of an ACK response), the hardware attempts to retransmit the frame with the rate r_0 up to $c_0 - 1$ times. If the transmission is still unsuccessful, the hardware tries the rate r_1, c_1 times then the rate r_2, c_2 times and finally the rate r_3, c_3 times. When the transmission has failed $c_0 + c_1 + c_2 + c_3$ times, the frame is discarded. Note that the sum of all counters implicitly indicate the frame retry limit.

If the MRR mechanism is disabled or the RAA is disabled¹, the frame will be (re)transmitted in total for ten times by default in the MadWifi implementation. However, if MRR is

¹Disabling RAA means that a fixed rate is used for frame transmissions and MRR is also disabled.

RAA	Tx Rate	Count
AMRR [2]	r_0	1
	$r_1 = r_0 - 1$	1
	$r_2 = r_0 - 2$	1
	$r_3 = \text{lowest rate}$	1
Onoe [3]	r_0	4
	$r_1 = r_0 - 1$	2
	$r_2 = r_0 - 2$	2
	$r_3 = \text{lowest rate}$	2
SampleRate [4]	r_0	2
	$r_1 = r_0^a$	3
	$r_2 = \text{lowest rate}$	3
	$r_3 = 0$	0

^aIf no ACKs are registered yet for the specific rate r_0 , then r_1 is set to the lowest rate.

TABLE I
MULTI-RATE RETRY CHAIN PARAMETERS.

enabled, then the number of transmission attempts for a frame is determined by the retry chain parameters corresponding to the RAA used.

The current stable release (v0.9.4) of the MadWifi driver provides four RAAs [2], [3], [4], [5] with SampleRate as the default algorithm². MRR functionality is also enabled by default and is used by all the algorithms. The general understanding justifying the MRR usability is that RAAs are able to estimate the best long-term transmission rate to use and because of some short term fluctuation of the channel some frames are lost. To recover quickly from this state, some fast countermeasures should be undertaken such as lowering the rate for ensuing retransmissions of a frame without having to wait for the adjustment of the long-term rate. Table I summarizes how the parameters of the retry chain are initialized for each of the RAAs considered in this work.

We conclude this section by noting that MRR mechanisms outlined above have similarity to the incremental redundancy and hybrid ARQ approaches explored in the context of cellular networks. Note that in contrast to MRR, incremental redundancy is applied at the physical layer with greater receiver feedback. Recently, mechanisms such as PPR [8] aim to improve the utilization of available capacity by selectively retransmitting only those bits that get corrupted over the air. However, neither traditional incremental redundancy mechanisms nor PPR-like mechanisms are directly implementable with commodity 802.11 hardware as they require more flexible and configurable radios such as software-defined radios.

III. SIMULATION ENVIRONMENT

Our evaluations are based on simulations using the well-known QualNet simulator (version 4.0) [9]. We implemented SampleRate, Onoe and MRR functionality in QualNet, strictly following the MadWifi implementation.

²Minstrel [5] is a variant of the SampleRate algorithm. It dynamically adjusts the multi-rate retry chain parameters in contrast to the static approach adopted by other rate control mechanisms. However, the specific adjustment mechanism chosen is tightly coupled to TCP traffic, so we do not consider this mechanism in order to make our study relevant for all types of applications.

Physical Layer		MAC Layer	
Frequency band	5 GHz	MAC Protocol	DCF
Noise floor	-94 dBm ^a	CWmin	15
Tx Power	17 dBm	CWmax	1023
Max. Rx Sensitivity	-65 dBm	RTSThresh	2043 Byte
Antenna	Omni	SIFS time	16 μ s
Antenna Efficiency	0.8	SLOT time	9 μ s
Antenna Loss	0.5 dB	Access Mode	Basic
Antenna Height	1.5m	Connection	AP Mode

^aThe background noise level is set to -94 dBm as suggested by measurements presented in specialized studies [10].

TABLE II
SIMULATION PARAMETERS.

Table II lists the values for simulation parameters common to all our experiments; these parameters are compliant with the IEEE 802.11a standard considered in our evaluations.

The frame retry limit is a key MAC layer parameter in our work. The 802.11 standard defines a long and a short retry limit with seven and four as the suggested values for these two limits, respectively. However, the default retry limits of some wireless cards differ from the standard. MadWifi implementation of the 802.11 protocol uses a default value of ten for the retry limit when MRR is disabled³. When the MRR mechanism is enabled, the maximum number of transmission attempts for a frame is based on the retry chain parameters of the RAA, i.e., $c_0 + c_1 + c_2 + c_3$. For example, SampleRate's MRR mechanism uses a retry limit of eight as shown in Table I.

Channel Model. We consider indoor 802.11 scenarios. To model the radio propagation in indoor environments, we use a channel model that is composed of a large-scale path loss model, a log-normal shadowing model and a Ricean fading model. The average path loss PL (in dB) for a transmitter and receiver separated by distance d is calculated as:

$$PL_{dB} = PL_{dB}(d_0) + 10\alpha \log\left(\frac{d}{d_0}\right) + X_\sigma + \text{Fading} \quad (1)$$

where X_σ is a zero-mean Gaussian distributed random variable with standard deviation σ and α denotes the path loss exponent. We set α to 3.0 and σ to 4.0, which are reasonable values for the indoor environment. To account for environment mobility, we set the maximum velocity of objects in the environment to 1 m/s.

IV. MRR IMPACT ON PERFORMANCE

In this section, we study the impact of multi-rate retry (MRR) mechanism on 802.11 rate control performance. The key metric of interest is the application-level throughput computed as the amount of application data delivered at the destination per unit of time. We refer to this metric as *goodput* in the paper and measure it in Mbps. Maximizing goodput is the goal of all rate control techniques. We also

³Note that the MadWifi implementation lets the user customize this value.

consider two additional metrics: (a) *loss ratio* — the ratio of number of frames discarded by the MAC layer (after failing to successfully transmit even after several retries up to the specified frame retry limit) to the total number of original frames transmitted; (b) average number of retransmissions per frame (in short, *retx ratio*). Clearly, smaller loss ratio and smaller retx ratio are preferred.

We consider two scenarios:

- Scenario I: This is a simple scenario that models communication between an Access Point (AP) and a client station separated by a distance d ; we present results obtained from averaging over a large number of randomly chosen d values. This scenario is representative of a home WLAN setting.
- Scenario II: This scenario reflects a typical office WLAN scenario where multiple client stations contend to communicate with a AP node. Specifically, we consider four client stations distributed around the AP for our experiments. Note that collisions can occur in this scenario due to simultaneous transmission by multiple stations as well as due to hidden terminals (two stations hidden from each other communicating with an intermediate AP node).

In both scenarios, we setup a CBR/UDP traffic flow between each client station and the AP. We use fixed 1KB frame size. We consider a worst case situation where each flow is continuously backlogged, i.e., each flow generates enough traffic to saturate even a link with maximum capacity (54Mbps). Each data point presented is the result of averaging over multiple simulation runs, each having a duration of 180s.

A. Scenario I Results

Fig. 1 shows the impact of MRR with different RAAs in this scenario. To have a fair evaluation and to clearly identify the impact of MRR, we compare the goodput for each RAA in three cases: (1) with MRR disabled and retry limit set to ten, the default value in the MadWifi implementation; (2) with MRR disabled but retry limit set to the total number of retries specified by the MRR scheme corresponding to each RAA (i.e., four for ARF, eight for SampleRate and ten for Onoe); and (3) with MRR enabled and retry limit set to the total number of retries specified by the MRR scheme corresponding to each RAA. Goodput performance with ARF and SampleRate is represented by three bars following the above specified order, while Onoe performance is represented by two bars with and without MRR mechanism enabled — dark blue bar and light grey bar for Onoe are identical. From Fig. 1, the main observation is that the impact of MRR is different with different RAAs. However, this is expected given that each RAA and the corresponding MRR retry chain parameters are different. Comparing the two blue bars for ARF and SampleRate and the two bars from Onoe, we also observe that MRR reduces goodput with ARF/AMRR, whereas it the other way around with SampleRate and Onoe — the percentage values in the figure indicate the amount of goodput reduction/improvement.

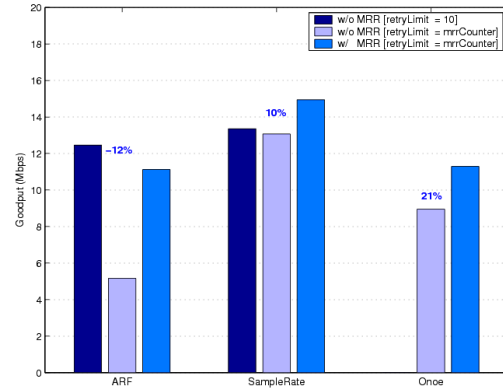


Fig. 1. Impact of MRR on goodput performance with ARF, SampleRate and Onoe in Scenario I.

To better understand the differences in MRR impact with different RAAs, Table III shows the results for loss ratio and retx ratio. To ease the analysis, results are shown for only two cases for these two metrics — `retryLimit=10` when MRR is disabled and `retryLimit` set based on MRR retry chain parameters of the RAA when MRR is enabled. Looking at ARF results, we see that retx ratio is reduced with MRR, whereas loss ratio increases. This is because the MRR mechanism associated with ARF permits only a small number of retries (specifically, four). While fewer retries aid in improving the responsiveness of the RAA, it can also increase losses as confirmed by the results. Greater number of frame losses could lead to disastrous performance of applications sensitive to loss. For example, if the application is based on TCP, goodput will be further reduced; for each frame loss reported by the MAC layer to higher layers, TCP will initiate its own loss recovery procedure, increasing the time to successfully deliver a packet and lowering the goodput. SampleRate, on the other hand, benefits (albeit marginally) from using MRR. This is because MRR helps in recovering from frame transmission failures arising from aggressive use of higher bit-rates, which is confirmed by the slight reduction in losses with MRR. Onoe benefits the most from using MRR as seen from the substantial improvement across all metrics. This is because Onoe multi-rate retry chain provides enough opportunity to recover from frame transmission failures arising due to short-term channel fluctuations, which in turn translates into improvement in goodput. However, the extent of goodput improvement is limited by the conservative nature of the Onoe RAA. In this scenario, SampleRate performs better overall with and without MRR.

B. Scenario II Results

We now look at the impact of MRR in the presence of channel contention using scenario II described earlier. In a congested environment, the rate control problem is more difficult because RAAs have to additionally guard against misinterpreting frame transmission failure due to multiple access collision as channel induced errors and vice-versa. Recent

	ARF ^a		SampleRate		Onoe	
Loss Ratio	0.2	0.4	0.1	0.09	0.1	0.05
Retx Ratio	5.7	3.6	4.3	4.3	5.0	4.1

^athe right-hand column for each algorithm denotes the values obtained when MRR is enabled.

TABLE III
IMPACT OF MRR ON LOSS RATIO AND RETX RATIO WITH ARF,
SAMPLERATE AND ONOE IN SCENARIO I.

work [11] presents comparative analysis of the performance of RAAs in a congested scenario, showing that RAAs that rely purely on link-layer information (e.g., frame transmission failures, transmission time) perform poorly, especially with increasing number of active nodes in the network.

Fig. 2 and Table IV show the impact of MRR on performance with each of the RAAs in scenario II. Note that here we only consider two cases for each RAA: (1) with MRR disabled and retry limit set to ten, the default value in the MadWifi implementation; and (2) with MRR enabled and retry limit set to the total number of retries specified by the MRR scheme corresponding to each RAA (i.e., four for ARF, eight for SampleRate and ten for Onoe).

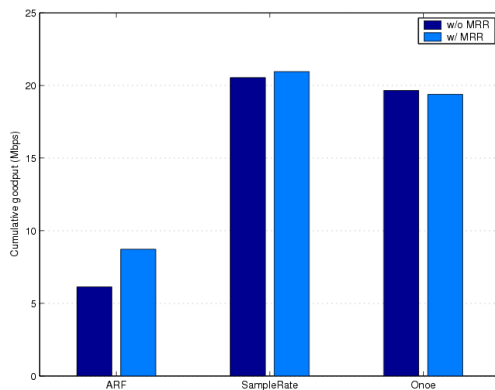


Fig. 2. Impact of MRR on goodput performance with ARF, SampleRate and Onoe in Scenario II.

	ARF		SampleRate		Onoe	
Loss Ratio	0.075	0.425	0.013	0.025	0.013	0.008
Retx Ratio	5.7	4.1	4.6	4.8	4.6	4.6

TABLE IV
IMPACT OF MRR ON LOSS RATIO AND RETX RATIO WITH ARF,
SAMPLERATE AND ONOE IN SCENARIO II.

These results show that, in contrast to the previous scenario, ARF's goodput improves with MRR in presence of collisions. This is because it benefits from fewer number of retries which is more effective in congested scenarios. However, loss ratio also increases as before. Getting a higher goodput at the expense of link reliability may not be a desirable thing from the application standpoint as it may hurt overall quality of service for some applications. We also observe that ARF

performance degrades with increasing number of active nodes in the network (results not shown for brevity). SampleRate goodput is better compared to other algorithms in presence of contention. Though loss ratio with SampleRate is doubled from using MRR, it is still quite small to have any undesirable impact on application performance. Onoe with MRR enabled shows fewer losses, but also a slight decrease in goodput. Nevertheless, its performance is still satisfactory to make it a suitable choice for applications demanding high reliability.

V. CONCLUSIONS

In this paper, we have focused on the multi-rate retry capability available with current 802.11 implementations and studied its impact on performance with state-of-the-art rate control mechanisms in typical indoor wireless LAN scenarios using simulations. Our study has shown that MRR is more effective in a non-congested environment, thus necessitating a mechanism to differentiate between congested and non-congested situations in order to better exploit MRR capability. Results also show that the specific characteristics of the RAA and multi-rate retry chain parameters play a big part in determining the performance impact of MRR. Through our implementation and evaluation, we have also observed that the statistics used by RAAs become unreliable due to coupling between RAA and MRR, thus reducing the net benefit from using MRR.

We are using these crucial insights in on-going work to guide the design of rate control mechanisms that can better exploit MRR feature [7]. In future, we plan to validate our simulation results on a indoor wireless network testbed. We also plan to investigate the potential benefits of MRR in more dynamic and outdoor 802.11 wireless network scenarios.

REFERENCES

- [1] A. Kamerman and L. Monteban. WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band. *Bell Labs Technical Journal*, 1997.
- [2] M. Lacage, M. H. Manshaei, and T. Turletti. IEEE 802.11 Rate Adaptation: A Practical Approach. In *Proc. ACM MSWiM*, 2004.
- [3] Onoe Rate Adaptation Algorithm. http://madwifi-project.org/browser/madwifi/trunk/ath_rate/onoe.
- [4] J. Bicket. Bit-rate Selection in Wireless Networks. Master's thesis, MIT, Feb 2005.
- [5] Minstrel Rate Adaptation Algorithm. http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel.
- [6] MadWifi: Multiband Atheros Driver for WiFi. <http://madwifi-project.org/>.
- [7] N. Koçi and M. K. Marina. Understanding the Role of Multi-Rate Retry Mechanism for Effective Rate Control in 802.11 Wireless LANs. The University of Edinburgh, Informatics Research Report EDI-INF-RR-1335, Jul 2009. <http://homepages.inf.ed.ac.uk/marina/papers/EDI-INF-RR-1335.pdf>.
- [8] K. Jamieson and H. Balakrishnan. PPR: Partial Packet Recovery for Wireless Networks. In *Proc. ACM Sigcomm*, 2007.
- [9] QualNet Network Simulator. <http://www.scalable-networks.com>.
- [10] B. Fu et al. Wireless Background Noise in the Wi-Fi Spectrum. In *Proc. 4th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, 2008.
- [11] K. Ramachandran et al. Scalability Analysis of Rate Adaptation Techniques in Congested IEEE 802.11 Networks: An ORBIT Testbed Comparative Study. In *Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WOWMOM)*, 2007.