Wireless Networks (2020) 26:3793–3830
https://doi.org/10.1007/s11276-020-02295-2

MAC-layer rate control for 802.11 networks: a survey

Wei Yin[1] · Peizhao Hu[2] · Jadwiga Indulska[1] · Marius Portmann[1] · Ying Mao[3]

Published online: 14 March 2020

Abstract

Rate control at the MAC-layer is one of the fundamental building blocks in many wireless networks. Over the past two decades, around thirty mechanisms have been proposed in the literature. Among them, there are mechanisms that make rate selection decisions based on sophisticated measurements of wireless link quality, and others that are based on straightforward heuristics. Minstrel, for example, is an elegant mechanism that has been adopted by hundreds of millions of computers, yet, not much was known about its performance until recently. The purpose of this paper is to provide a comprehensive survey and analysis of the existing solutions from the two fundamental aspects of rate control—metrics and algorithms. We also review how these solutions were evaluated and compared against each other. Based on our detailed studies and observations, we share important insights on future development of rate control mechanisms at the MAC-layer. This discussion also takes into account the recent developments in wireless technologies and emerging applications, such as Internet-of-Things, and shows issues that need to be addressed in the design of new rate control mechanisms suitable for these technologies and applications.

Keywords Rate adaptation · Rate control · 802.11 · WIFI

1 Introduction

Wireless connectivity is fundamental in supporting the paradigm shift from computing on desktops to Internet-of-Everything [1]. 802.11 networks are still one of the most common wireless networks, which offer high bandwidth and yet efficient energy consumption [2]. In the 802.11 networks, the wireless medium is shared, and channel conditions are unpredictable due to node mobility, channel fading, and interference [3]. Due to these channel dynamics, wireless link quality varies over time, which has an impact on the performance of the networks, especially for multi-hop networks.

The transmission rate is one of the network parameters, which determines how fast a node can send data onto the wireless medium. In principle, if the link quality is good such that signals are strong, higher rates can result in higher goodput and lower the channel occupancy time. In contrast, if the channel quality is poor, selecting a higher rate increases the probability of packet drop due to weak signals, hence causing additional retransmissions and wasting channel bandwidth. Instead, a lower rate should be utilized to ensure a higher delivery ratio. Therefore, selecting an appropriate transmission rate for the given channel quality becomes an important task in improving the performance of each wireless link. This is the task of rate control at the MAC layer.

There are two fundamental components at the core of every rate control mechanism: (1) a *metric* used to estimate the link channel quality. As examples, throughput, Frame Loss Ratio (FLR), and Signal Noise Ratio (SNR) are three commonly used metrics in rate control, and (2) an

✉ Wei Yin
yinwei168@gmail.com

Peizhao Hu
ph@cs.rit.edu

Jadwiga Indulska
jaga@itee.uq.edu.au

Marius Portmann
marius@itee.uq.edu.au

Ying Mao
ymao41@fordham.edu

[1] The University of Queensland, School of ITEE, Brisbane, Australia

[2] Rochester Institute of Technology, Rochester, USA

[3] Fordham University, The Bronx, USA

algorithm that defines strategies to quickly select a transmission rate appropriate for the estimated channel conditions. First, the objectives of the algorithms vary. The typical objectives are selecting an optimal rate that can maximize the achieved throughput or maintain the FLR below a predefined threshold. Second, the methods to update to the optimal rate include gradual rate increase/decrease [4] and best rate selection [5], and they affect the responsiveness of rate control algorithms to fast channel changes. Third, the subject which conducts the link quality estimation varies, and this may affect the accuracy of the link quality estimation. For example, the SNR is calculated more accurately at the receiver side [6]. Without the receiver‘s noise and interference information, the sender is unable to predict an accurate SNR value [7]. It is important to note that rate selection made at the sender side does not require changing the 802.11 standards, e.g. the frame format. However, notifying the rate decision from the receiver to the sender requires modification of the control frames [6] or the mandatory Acknowledgment (ACK) transmission rates [8, 9]. Fourth, the selected rate may be used for the next frame or all frames in the next Rate Adaptation Period (RAP). This may affect the responsiveness or stability of rate control mechanisms.

MAC-layer rate control has been an active research topic for almost two decades. Biaz et al. [10] and Das et al. [11] present a survey of rate control mechanisms designed at the early stage. But, rate control mechanisms designed for new technologies, e.g. 802.11n and 802.11ac, are not covered. They neither summarize the lesson learned from the study nor give guidelines for rate control researchers. Moreover, there is no indication of research directions for the next generation rate control mechanisms for technologies such as 802.11aa, 802.11ah, 802.11ax, etc.

This paper serves as a comprehensive survey of the existing MAC-layer rate control mechanisms, including state-of-the-art solutions. Drawing from our own experience in working on this research topic, we perform a qualitative and quantitative analysis of these solutions using our classification scheme and analytic criteria. Throughout this survey paper, we share the insight of “what works” and “what does not work” when designing a new MAC-layer rate control mechanism. We show important design principles and open research issues that can be of interest for researchers that work on this research topic.

The rest of this paper is organized as follows. Section 2 provides an overview of the fundamental mechanics of rate control at the MAC-layer. This is followed in Sect. 3 by a comprehensive survey of the existing rate control mechanisms developed in the last two decades. Similarities and differences between these existing solutions are drawn to form the analysis and technical discussions in Sect. 4. We share important insights into the development of rate control in the MAC-layer in Sect. 5. Section 6 discusses new opportunities and shares our view on the role of rate control for various emerging applications. Section 7 concludes the paper.

2 Background of MAC-layer rate control in 802.11 networks

Typically, the selection of transmission rate depends on the channel quality—good channel quality can support higher transmission rates. To accommodate the variation in channel quality, the IEEE standards define several discrete transmission rates for each coding method [12]. For example, the 802.11b mode has four available rates (1, 2, 5.5 and 11 Mbps), 802.11a and 802.11g support eight rates (6, 9, 12, 18, 24, 36, 48, 54 Mbps) and 802.11g supports all above 12 rates.

The goal of MAC-layer rate control in 802.11 networks is to select an appropriate rate that achieves the optimal throughput should the channel conditions change. In this section, we provide an overview of why MAC-layer rate control is needed and what means are available to perform rate adaptation.

2.1 Problems concerning rate control

Data transmission can be affected by changes in signal quality and collisions of data or control frames.

2.1.1 Changes in signal quality

In 802.11 networks, the signal quality (or strength) of received frames varies due to a number of reasons, including node mobility or signal fading [13]. When two nodes move towards or away from each other, there is a change in signal quality corresponding to the path loss. For example, the Free-Space Path Loss (FSPL), $FSPL = (\frac{4\pi d}{\lambda})^2$, describes the power loss as a function of distance d between the transmitter and receiver for a line-of-sight link in an environment that is free of interfering sources; λ is the signal wavelength (in meters) [14]. Figure 1 shows the performance of all the 802.11a fixed rates as a function of path loss in free-space [15]. There is an optimal rate that can achieve the optimal throughput for a specific path loss. If the objective is to maximize throughput in the FSPL model, the task of rate control is to determine the optimal rate for a given path loss value. However, the 802.11 networks use the unlicensed spectrums (2.4 GHz and 5 GHz bands), which are shared with other technologies, such as microwave, Bluetooth and cordless phones. These

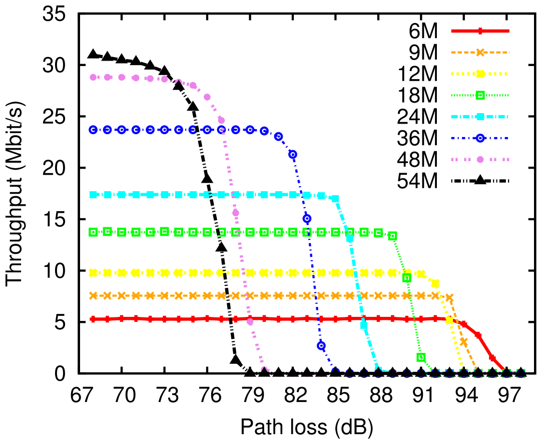


Fig. 1 Performance of 802.11a fixed rates

technologies share the wireless medium and introduce interference in the form of noise. Substantial increase in noise can significantly reduce the signal quality as reflected in the signal-to-noise ratio, $SNR = \frac{SignalPower}{NoisePower}$.

2.1.2 Frame collisions

While destructive interference degrades the signal quality in the form of noise, decodable interfering frames can collide with transmissions and reduce goodput. This interference is typically reflected in the Signal-and-Interference-to-Noise Ratio (SINR), $SINR = \frac{Signal}{Noise+Interference}$. Commonly, there are two types of collisions: hidden terminal collision and contention collision. The former occurs when frames from two or more non-carrier-sensing nodes collide [16], causing retransmissions. To address this problem, a typical approach is to lower the rate when the throughput drops [5] or frame loss ratio increases [4]. However, this will not address the issue and often makes the matter worse because lowering the rate increases the airtime of a frame and therefore increases the probability of collision. Request to Send/Clear to Send (RTS/CTS) is an effective protocol to address the hidden terminal collision by reserving the channel. In the case of contention collision, frame losses are caused by a large number of nodes competing for channel access [17]. Under high channel utilization, two or more nodes' backoff timers of the Distributed Coordination Function (DCF) mechanism [12] are likely to expire at the same time. Thus, they attempt to transmit simultaneously, causing contention collision. Following the general rule of rate control, the rate will be lowered, making it worse because doing so increases the airtime, prolonging the channel utilization. The use of RTS/CTS cannot address the contention collision problem completely, as RTS frames can collide due to simultaneous backoff timeout. However, it can reduce the channel occupying time and improve network congestion, as RTS is usually shorter than data frames.

Techniques, such as Clear Channel Assessment (CCA) or Channel State Information (CSI), can help to detect the channel activities, but they are not an ideal solution to improve performance under both collision environments. CCA checks whether the Received Signal Strength Indicator (RSSI) is above a pre-defined threshold. However, the transmission of a hidden node is often overlooked by the peer. The RSSI is not increased at the peer hidden node. The CSI carried in the pilot/training signals of a frame is unseen by the hidden node. Both CCA or CSI cannot reduce the probability of simultaneous backoff timeout events. Therefore, the contention collision cannot be addressed by either method.

In the situation of contention collision, a scheduling algorithm is needed to coordinate the channel access. For example, 802.11ah [18] divides mass stations into Restricted Access Window (RAW) [19–21] groups. The time is sliced into RAW periods. Each RAW group is given a RAW period to send data frames by turns. The whole process is scheduled by the Access Point (AP). During a RAW period, stations belonging to the RAW group access the channel using the Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. Therefore, the contention problem is solved by the RAW grouping method.

2.2 Taxonomy of rate control mechanisms

While rate control mechanisms are categorized by the metrics they use, they can also be characterized by the following factors based on the hardware processing capability, responsiveness of the mechanism, and the subject which makes the rate decision:

2.2.1 Per-frame or adaptation window

The rapid changes in channel conditions require a responsive adaptation of transmission rates. Ideally, it should select an optimal rate for every frame transmission. However, this requires low-latency hardware that may be dedicated to processing MAC-layer frames. For example, SoftRate [22] uses Software-Defined Radio (SDR) to achieve this per-frame rate selection. Some wireless drivers on the Linux operating system, e.g. Madwifi, make a rate selection every *Rate Adaptation Period* (RAP). RAP is a duration for which an optimal rate will be used for frame transmissions. For every RAP, the performance of one or more rates is calculated based on the desired objectives. The statistics are used to determine the optimal rate for the next RAP. The length of a RAP has an impact on the responsiveness of the rate control mechanism.

2.2.2 "Rate Ladder" or direct rate selection

The defined discrete rates form a rate ladder for rate adaptation. Given a current rate, $R_{current}$, there are three possible adaptations when channel conditions change: (1) to lower the current rate $R^{\downarrow}_{current}$ one level down the ladder, $R^{-}_{current}$, until the base rate; (2) to increase the current rate $R^{\uparrow}_{current}$ one level up, $R^{+}_{current}$, until the maximum rate; and (3) to maintain the current rate. Generally speaking, mechanisms can increase or decrease the rate one level at a time. On the other hand, the direct rate selection computes the performance metric for each rate against a given objective (e.g., maximizing achievable throughput in Minstrel [5]) and selects a rate that achieves the optimal utility. It is sensitive to rapid and significant changes in channel conditions.

2.2.3 Estimation of metrics at sender or receiver

For most metrics, the computation can be done at the sender. There are metrics that use the MAC-layer ACK frame to infer the success or failure of the transmission, while there are others that use the physical layer feedback (retry count and rate). In contrast, there are metrics, such as Bit Error Rate (BER) or SNR, which require measurements of the transmitted frames at the receiver. These measurements are then piggybacked via modified ACK frames to the sender in the rate adaptation processes. There is a delay in piggybacking the measurements. Besides, modifying the ACK frames introduces overhead and causes compatibility issues.

2.3 Common features for rate control

Most rate control mechanisms use the status of frame transmission (Tx) or reception (Rx) of the last frame or adaptation window to make a rate adaptation decision for the next frame or adaptation window. A number of metrics have been considered as the Tx/Rx status, including frame transmission time, throughput, SNR, and BER.

As an example, throughput can be calculated as $\frac{1sec}{T_{tx\_perfect}} * FrameSize$, and the IEEE 802.11 standard [12] defines how we can compute the perfect transmission time (transmission time in error free channel), $T_{tx\_perfect}$:

$$T_{tx\_perfect} = DIFS + BO + T_{DATA} + SIFS + T_{ACK} \tag{1}$$

where DCF Interframe Spacing (*DIFS*) and Shortest InterFrame Spacing (*SIFS*) are a type of inter-frame spacing defined by the IEEE 802.11 standard; *BO* is the average CSMA/CA backoff time without a frame failure. *BO* is calculated as $\frac{CW_{min} * T_{slot}}{2}$ where T_{slot} is the slot duration, and CW_{min} is the minimal Congestion Window (CW) size. However, CW is exponentially increased upon a transmission failure until it reaches the maximum, CW_{max}. It is reset to CW_{min} if the frame succeeds or finally fails after the maximum retry count is reached. T_{DATA} and T_{ACK} are the transmission time of the data and acknowledgement frames respectively, and they are calculated, as a general form T_{Frame}, as

$$\begin{aligned} T_{Frame} = {} & Ceil((16 + 8 * LEN + 6)/NDBPS) * T_{SYM} \\ & + T_{Preamble} + T_{Signal} \end{aligned} \tag{2}$$

where T_{SYM} is the symbol interval (defaults to 4 μs for 20 MHz channel spacing), $T_{Preamble}$ is the Physical Layer Convergence Protocol (PLCP) header preamble duration (defaults to 16 μs), and T_{Signal} is the duration of the signal Binary Phase Shift Keying-Orthogonal Frequency Division Multiplexing (BPSK-OFDM) symbol (defaults to 4 μs) as defined in the IEEE 802.11 standard; *LEN* is the respective frame size, and *NDBPS* is calculated as $k * r$ where k is a coefficient and r is the bit rate. While the standard describes a reference specification and the fundamental principle is the same, actual implementation depends on individual mechanisms; for example, Minstrel [5] calculates throughput as $\frac{1sec}{SIFS + T_{DATA}} * FrameSize$, which ignores DIFS and the ACK time. In [23], our results showed that the link capacity estimated according to the specification can be far from reality. In our study, the abnormal link capacity was caused by a wrong estimation of the back-off time.

More example metrics are discussed in the specific rate control mechanisms. A collection of other common techniques for rate control is shown in Table 1 and briefly described below.

CSI describes various channel properties of a wireless link, including how a signal propagates from the transmitter to the receiver, and the combined effect of fading, power decay, and scattering. The sender can adapt transmissions according to the measured CSI values by the receiver and hence improves performance. In a Single Input Single Output (SISO) channel, CSI is only SNR, and the terminology of CSI is not used. In a Multiple Input and Multiple Output (MIMO) configuration, a CSI matrix is constructed by transmitting a signal on each antenna and recording the responses on the receiving antennas. Based on the measured received signals, the node determines the most effective antenna according to the strength of the received signal.

Multi-Rate-Retry (MRR) is supported in several wireless chipsets, including Atheros. In an Atheros chipset, four retry rates (r0, r1, r2 and r3) and corresponding maximum retry counts (c0, c1, c2 and c3) can be specified by the driver and notified to the chipset. r0 can be retried for c0 times if attempts are failed, then r1 will be used, following

Table 1 Common techniques for rate control

Technique	Description
Channel State Information (CSI)	Describes various channel properties of a wireless link
Multi-Rate-Retry (MRR)	Defines the retry policies for failed frames in the driver; when a frame fails to transmit, it will retransmit it using the specified retry rates
Rate Adaptation Period (RAP)	Defines how long a rate will be utilized for transmitting frames once it is selected as the transmission rate.
Lookaround rate	Transmits a small portion of frames using other (typically higher) rates rather than the current rate to probe the performance of these rates
Frame aggregation	Bundles multiple frames in a single frame in order to minimize the communication overhead, e.g., backoff, interframe spacing, frame header and acknowledgement

down to r3 until success. The rate control mechanism usually decides the setting of retry rates and maximum retry counts. MRR is used to deal with short term channel variations, while rate control is for long-term channel variations.

Rate adaptation Period (RAP), as described earlier, defines a period of time that a rate will be utilized for transmitting once it is selected as the best transmission rate. In many common systems, rate selection is based on RAP, which does not require low-latency hardware, such as Field-Programmable Gate Array (FPGA), that is capable of selecting rate on a per-frame basis.

Lookaround rate is a sampling approach used by some rate control mechanisms, including Minstrel. A rate is used for a whole rate adaptation period once it is selected. The problem for this method is that the mechanism cannot gather fresh statistics about other rates than the selected rate. Therefore, it may make wrong rate decisions with outdated statistics. Minstrel uses a small percentage of frames to try Lookaround rates (rates other than the selected best rate) to probe performance. It sacrifices throughput for up-to-date statistics about Lookaround rates because a Lookaround rate usually performs worse than the best-selected rate.

Frame aggregation is a feature used in a MIMO environment. As the MIMO technology significantly increases the transmission rate, the time to transmit a frame reduces sharply. Assuming that the channel coherence time is constant, we can transmit more data using a high MIMO rate compared to a lower SISO rate during the channel coherence time. Hence, bundling multiple frames in a single frame is proposed to minimize the communication overhead, e.g., backoff, interframe spacing, frame header and acknowledgement.

The acronyms are summarized in the "Appendix".

3 Existing mechanisms for rate control

MAC-layer rate control has been an active research area for almost two decades. In this section, we present a survey of the existing MAC-layer rate control mechanisms for 802.11 networks. Figure 2 shows a family tree (sorted by the timeline) of these mechanisms. In the figure, arrows indicate the inheritance relationship between two connected mechanisms. In the discussion, we group the mechanisms according to the *metrics* that they use to evaluate the channel or link quality. Through our literature review, we have identified metrics based on (1) consecutive transmission result, (2) frame loss ratio, (3) transmission time, (4) throughput, (5) signal to noise ratio, (6) bit error rate, (7) frame error rate, and (8) combined metrics.

3.1 Consecutive transmission result (CTR)

The earliest rate control mechanisms select rates based on the results of previous transmissions, usually on the number of successive transmissions.

3.1.1 Automatic rate fallback (ARF)

ARF [24] gradually increases or decreases the rate based on the results of previous successive transmissions. It maintains two counters, the number of successes C_{succ} and failures C_{fail} depending on whether acknowledgement frames are received. Figure 3 illustrates the operations of ARF. Starting with the highest rate, as the current rate $R_{current}$, it checks the transmission result of $R_{current}$. If success, it increases C_{succ} and checks whether $C_{succ} \geq \alpha$, where α is a rate increase threshold (defaults to 10 in ARF). It will maintain the current rate, if $C_{succ} \leq \alpha$. After ten consecutive successes, ARF will probe the performance of a higher rate $R^{+}_{current}$ that is one level above the current rate. If this probing frame (or *trial frame*) succeeds, it will

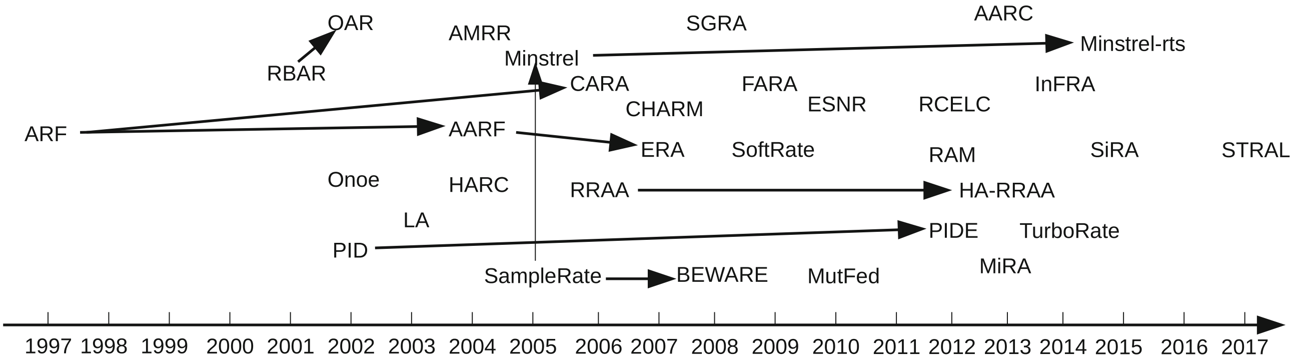
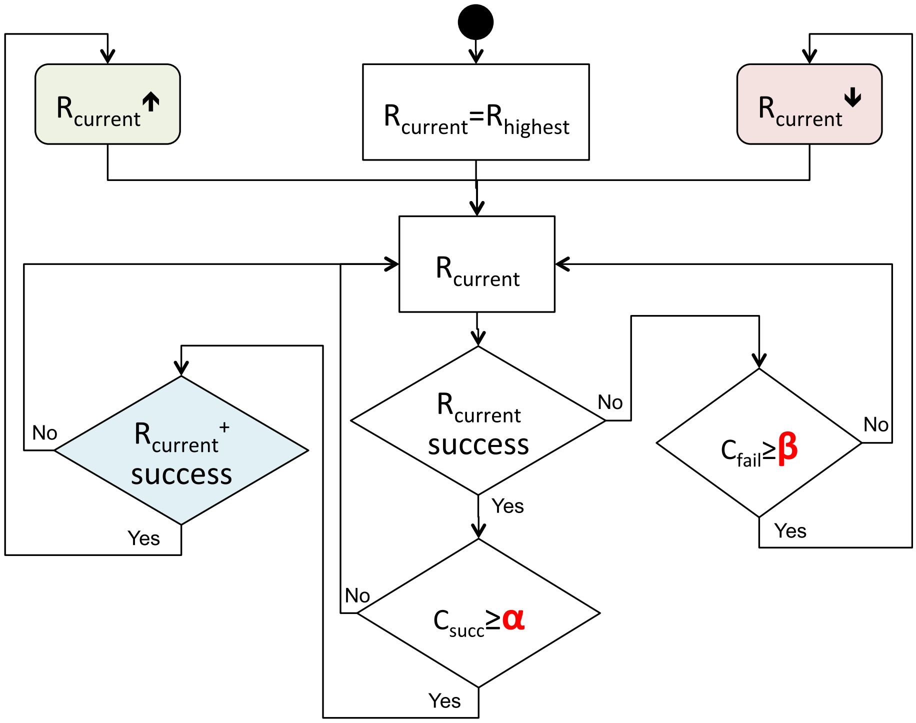


Fig. 2 A family tree of MAC-layer rate control mechanisms. Each arrow indicates an inheritance relationship

Fig. 3 Operations of ARF



increase the rate from $R_{current}$ to $R^{+}_{current}$; Otherwise, it falls back to the current rate. In the cases when the current rate has failed, ARF increases the C_{fail} and checks $C_{fail} \geq \beta$; where β is the rate drop threshold (defaults to 2). When the failure count reaches the threshold, it decreases from the current rate, $R_{current}$, until the base rate. Otherwise, it falls back to the current rate. Because ARF uses the two counters to accumulate the successive transmission results, a successful transmission resets C_{fail}, and a failed transmission resets C_{succ}.

One advantage of ARF is its simplicity. It requires only a timer and two counters to keep track of previous transmission results. ARF generally works well in extremely good or bad channel conditions. However, it is not an efficient mechanism. As ARF ignores the performance of the current rate, it will attempt to increase the rate after ten consecutive successful transmissions even though the current rate is the optimal rate. It will not converge at an optimal rate. Besides, ARF gradually increases or decreases rate rather than directly jumping to the best rate. This algorithm is unresponsive to rapid, yet significant, changes in channel conditions.

3.1.2 Adaptive ARF (AARF)

Lacage et al. [25] proposed AARF aiming to improve the performance of ARF. The goal of AARF is to converge at the optimal rate by dynamically adjusting the threshold α in ARF. The idea is to double the value of α whenever a trial frame has failed using the next higher rate $R^{+}_{current}$. This change reduces the number of failed probing frames (overhead) at the higher rate. The threshold α is reset to the initial value 10 when the rate is decreased.

AARF slows the response of the mechanism when the rate needs to be increased due to the doubling of the α value. Moreover, both ARF and AARF will increase the rate to the next level if the transmission of the first trial frame at the next higher rate $R^{+}_{current}$ is successful after α successful transmissions. In [26], Wong et al. show that the probability of success for a single trial frame is usually higher than 50%. Increasing the rate solely based on this criterion can potentially lead to incorrect rate adaptation decision, causing frame loss.

3.1.3 Collision-aware rate adaptation (CARA)

In wireless networks, frame losses may be caused by collisions or channel errors. CARA [16] improves ARF by incorporating two techniques to differentiate collision frame losses from channel error losses.

The mandatory technique is called *RTS probing*. If a transmission fails, RTS/CTS will be activated for the first retransmission because the failure may be due to hidden terminal collisions. If this retransmission fails as well, the failure is assumed to be caused by channel errors because the RTS/CTS exchange has already reserved the channel. Therefore, a lower rate should be used for the next retransmission. This mechanism works effectively to avoid occasional frame collisions. However, in situations where collisions are more than an occasional event, this solution will suffer a failure of every first transmission attempt. An optional technique is called *CCA detection*, which serves as a supplement to RTS probing. CARA uses the *busy* counter to determine the channel occupancy during a frame transmission. The CCA technique is used to support the backoff procedures of DCF [16]; for example, if CCA is busy, a node freezes its backoff process and resumes when the channel becomes idle. CCA is used to detect collisions during the SIFS time after a node transmitted a data frame and expects an acknowledgement. CARA will conclude the occurrence of a collision if the CCA is assessed as busy during this time, while the expected ACK reception does not start. As a result, the node will retransmit the data frame without increasing C_{fail} and lowering the transmission rate.

3.1.4 Effective rate adaptation (ERA)

The rate adaptation in ERA [27] is similar to AARF, but it offers a strategy to differentiate losses due to hidden terminal collisions and channel errors. This strategy does not incorporate the RTS/CTS mechanism as CARA does, and it can provide better throughput. If a frame is lost, the frame is divided into two fragments, one very short and the remainder, for the retransmission. The 802.11 standards specify that the channel is reserved for all remaining fragments once the first fragment gets access to the channel. Since the first fragment is very short, it is less likely to experience a collision. If it is successful, the mechanism infers that the loss is due to collisions. Otherwise, it is due to channel errors, and the base rate is utilised for retransmission. If a collision is inferred, the rate will not be decreased. The mechanism is based on the assumption that the short fragment will never experience collisions. However, the short fragment may be lost due to contention or hidden terminal collisions. This assumption may attribute the collision loss to channel errors, which causes the mechanism to unnecessarily decrease the rate.

3.2 Frame loss ratio (FLR)

Some mechanisms adjust the rate with a goal to maintain the frame loss ratio, $FLR = 1 - \frac{frames\_received}{frames\_sent}$, to be within a predefined range.

3.2.1 Robust rate adaptation algorithm (RRAA)

Unlike ARF and AARF, RRAA [26] adapts rate based on the FLR calculated over a time window (e.g., RAP) instead of relying on the success or failure of a single trial frame at the higher rate. In RRAA, a rate is selected for the next time window rather than the next frame. For each supported rate, RRAA defines two thresholds: *Maximum Tolerable Loss threshold (MTL)* and *Opportunistic Rate Increase threshold (ORI)*. The MTL and ORI set the upper and lower bound for FLR. With these two thresholds, we adjust the next rate R_{next} as follows:

$$R_{next} = \begin{cases} R^{-}_{current}, & \text{if } FLR \geq MTL \\ R^{+}_{current}, & \text{if } FLR \leq ORI \\ R_{current}, & \text{otherwise} \end{cases}$$

The value of MTL and ORI is determined by $\alpha L(R)$ and $\gamma L(R)$ respectively; where α and γ are tunable parameters, and the *critical loss ratio* $L(R)$ is calculated as

$$L(R) = 1 - \frac{TP(R_{current})}{TP(R^{-}_{current})} \tag{3}$$

where $TP(x)$ is the throughput of a rate under perfect channel conditions and is calculated as $\frac{1sec}{T_{frame}} * FrameSize$; T_{frame} is the perfect transmission time of a frame defined in Eq. (2). When calculating the *critical loss ratio* $L(R)$, RRAA assumes that the next lower rate has a frame loss ratio of zero, which overestimates the link quality.

Similarly to the RTS probing in CARA, RRAA has a technique called *adaptive RTS* (A-RTS) to differentiate collision losses from channel error losses. In the A-RTS process, RRAA defines a window (*RTSwnd*, and it is initialized to zero to disable RTS/CTS at the beginning) in which the RTS frame will be used to avoid collisions. The value of *RTSwnd* depends on whether (1) the frame is lost without the RTS frame, (2) the frame is lost with RTS frame or succeeds without the RTS frame, and (3) transmission succeeds with the RTS frame.

$$RTSwnd = \begin{cases} RTSwnd + 1, & \text{if } \ case\ (i) \\ half(RTSwnd), & \text{if } \ case(ii) \\ unchanged, & \text{otherwise} \end{cases}$$

Within each *RTSwnd* time interval, a counter is used to keep track of the remaining number of frames to be transmitted with the RTS frame. The A-RTS process decreases the counter by one each time a frame is sent out using RTS/CTS and is terminated when this counter reaches zero. Note that the use of RTS/CTS will introduce overhead, similarly to other RTS/CTS approaches.

3.2.2 History-aware RRAA (HA-RRAA)

Pefkianakis et al. [28] proposed improvements for the following three problems in RRAA. The first problem is rate oscillation when RRAA fails to converge at a rate that helps to achieve optimal throughput. RRAA increases the rate when the FLR drops below a threshold. This causes an increase in the FLR. Subsequently, RRAA reduces the rate to compensate for the FLR increase. Due to a low FLR, RRAA again increases the rate and creates the oscillation. HA-RRAA addresses this problem by dynamically adjusting an adaptive window that specifies the duration for which the current rate should be used. The adaptive window is similar to the rate adaptation window in other mechanisms, but it is not static. The adaptive window $W_{adaptive}$ is computed as

$$W_{adaptive} = W * 2^n * max\left(1, \frac{FLR}{10\%}\right) \tag{4}$$

where W is the channel coherence time that characterizes channel stability [28], $1 \leq n \leq 10$ is the number of consecutive failures in attempting to increase rate, and FLR is the frame loss ratio for the current rate. Essentially, the length of $W_{adaptive}$ increases proportionally to the increase of FLR over 10% and is doubled for every failed attempt to increase the rate. This enhancement extends the duration a rate is used and reduces the occurrence of rate oscillation. The $W_{adaptive}$ will be reset if the rate increase is successful or the rate decreases.

For 802.11a networks, HA-RRAA predefines the number of frames to be transmitted for each rate, ranging from 6-40 frames. Normally, higher rates should be able to transmit more frames successfully, but this is not true in deteriorating link conditions. To address frame loss, HA-RRAA employs a monitoring process. Let n be the predefined number of frames for a rate. HA-RRAA will monitor the performance in frame loss every $min(n, 10)$ frames. It will abort using the selected rate if the monitoring results show poor performance.

Finally, HA-RRAA takes into account the RTS/CTS overhead before activating this collision avoidance technique. By comparing the transmission times for data and acknowledgement frames, $T_{DATA+ACK}$, to the transmission times for RTS and CTS frames, $T_{RTS+CTS}$, HA-RRAA engages the RTS/CTS mechanism if $T_{DATA+ACK} \geq k * T_{RTS+CTS}$, where k is a coefficient fixed to 1.5 in the prototype.

3.2.3 Onoe

As shown in Fig. 4, Onoe [29] makes an indirect use of the FLR. Rather than making rate adaptation decisions based on FLR, it introduces a cumulative credit-based scheme which is changed depending on the average number of frame retries in a one-second window.

By default, Onoe sets the initial rate to 24 Mbit/s for both IEEE 802.11g and IEEE 802.11a while it uses 11 Mbit/s for IEEE 802.11b. The credit for the initial rate is initialized to 0. The mechanism increases the number of credits when less than 10% of frames, in a one-second window, need a retry, while decreasing it otherwise. When the current rate achieves ten or more credits, Onoe increases the rate to the next higher level, whereas it

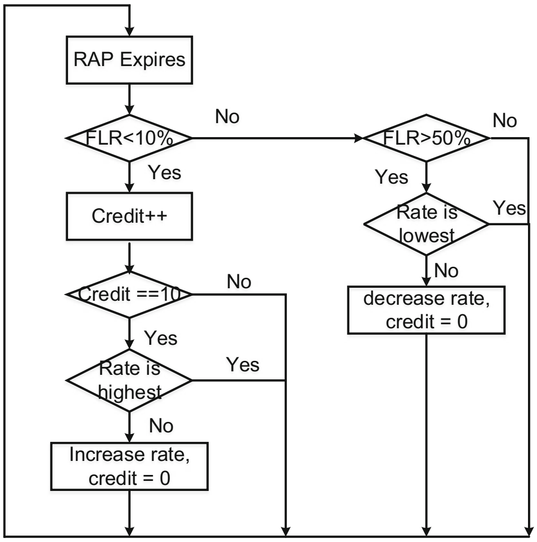


Fig. 4 How Onoe works

decreases the rate to the next lower level if ten or more frames have been sent and the average of retries per frame was greater than one. Onoe is relatively conservative because it does not increase the current transmission rate when it detects good quality channel opportunities but waits until the credit value reaches the threshold. As a consequence, Onoe misses some opportunities for performance enhancement. However, this enables Onoe to be stable. Experiments in [30] show that Onoe performs better than ARF and AARF on links where the highest rate experiences significant losses but still has the best throughput. This is because ARF and AARF will decrease the transmission rate when two consecutive failures occur while Onoe can tolerate 50% frame loss ratio because it only reduces the rate when the average of retries per frame was greater than one.

3.2.4 Adaptive multi-rate retry (AMRR)

AMRR [25] is proposed for high latency systems. Here, latency refers to the communication delay between the rate control mechanism at the MAC-layer and the wireless interface at the Physical layer. AMRR introduces a new feature called the *multi-rate retry* (MRR) chain, which proposes four candidate rates (i.e., $r0$, $r1$, $r2$ and $r3$) to attempt in case retransmissions are necessary. It shares similar characteristics as other MRR techniques described in Sect. 2. For example, the algorithm will first attempt to use rate $r0$ for transmission; if the transmission fails $c0$ times, then rate $r1$ is attempted for retransmission; in the case where rate $r1$ also fails $c1$ times, then rate $r2$ and $r3$ are attempted in the same manner until the base rate is reached. In AMRR, $r0$ is the best rate. $r1$ and $r2$ are subsequent lower rates of $r0$. $r3$ is always set to the base rate.

If more than ten frames have been transmitted in the last period and the FLR is less than 10%, AMRR increases r_0 to the next higher level. The retry chain fails if the FLR is more than 33% during the last period. Thus, the transmission rate r_0 is decreased to the next lower level, until the base rate. AMRR can handle the short-term channel variation. The long-term channel variation is handled by a binary exponential backoff mechanism which adapts the length of the period to change the value of the four rate-count pairs.

3.2.5 Proportional integral derivative (PID)

PID [4] adapts rates based on a *PID* controller. In essence, the controller is a control loop feedback mechanism that tries to minimise the difference (i.e., *error* in the control system's term) of the current and target frame loss ratio (i.e., $FLR_{current}$ and FLR_{target} respectively) as a result of switching to a new transmission rate. By default, the FLR_{target} is set to 14% for all rates.

To determine the appropriate transmission rate, the controller computes an adjustment value, *adj*, as follow.

$$adj = \gamma * (1 + sharpening) * (error_{current} - error_{last}) + \alpha * error_{current} + \beta * error_{avg} \tag{5}$$

where $error_{current}$ is the current error, and its value is calculated as $FLR_{target} - FLR_{current}$. $error_{avg}$ is the average of recent errors, while $error_{last}$ is the last error. In addition, there are four tuneable parameters. *sharpening* is a smoothing factor (non-zero when fast response is needed), whereas β, α and γ are the corresponding *proportional*, *integral* and *derivative* coefficients.

Using Eq. (5) the mechanism computes the adjustment value, *adj*, at the end of each rate adaptation period and decides on whether to switch to a new transmission rate. When *adj* is positive and its error (i.e., the difference between the target and respective frame loss ratio) is not higher than the error of the current rate $R_{current}$, the new rate R_{new} is set to the highest rate in the range of $R_{current} \leq R_{new} \leq (R_{current} + adj)$. When *adj* is negative and its error is not higher than the error of the current rate, the new rate R_{new} is set to the lowest rate in the range of $(R_{current} + adj) \leq R_{new} \leq R_{current}$. No rate adaptation is required if *adj* is equal to zero.

3.2.6 PID enhancement (PIDE)

The idea of PID is based on the control system's principle of a feedback loop. That is, defining a target frame loss ratio (FLR)—e.g., the target FLR in PID is fixed at 14%—let the rate control mechanism to adapt its rate to meet that target by minimizing the difference between the current FLR and the target FLR. The most significant problem with PID is its instability in rate selection, as discovered in [15].

During the adaptation process, PID will increase the rate whenever the current FLR is below the target threshold, regardless of the current channel conditions which may not be sufficient to support the higher rate. The consequence of this is that the FLR of the higher rate is higher than the target threshold, and this causes PID to drop the rate again. Hence, this results in oscillations in rate selection.

Yin et al. [15] implemented a verification mechanism in PIDE, which probes the achieved throughput of the proposed rate compared to the current throughput. If the proposed rate achieves higher throughput than the current sending rate, the algorithm should select the proposed rate. Otherwise, the rate adaptation requests should be ignored. The same rule applies for requests either to increase the rate or to decrease the rate.

The mechanism uses information passed by the PID controller to detect requests on rate adaptation at the end of each rate adaptation period. When a new rate is proposed, and the proposed rate has not been verified, PIDE sends n frames (three frames by default) using the proposed rate in the current rate adaptation period. Other frames within the adaptation period will continue to use the current rate. Each frame is associated with a status that records a number of statistics regarding the transmission (e.g., retry count). PIDE collects these statistics and maintains a table of performance information for each rate. At the end of the current adaptation period, PIDE compares the achieved throughput to decide whether to use the proposed rate in the next rate adaptation period. The achieved throughput TP is calculated as

$$TP = (1 - FLR) * (1s/T) * FrameSize \tag{6}$$

where FLR is the frame loss ratio, $1s$ is one second, and T is calculated as $DIFS + T_{DATA} + SIFS + T_{ACK}$ [12].

3.3 Average transmission time (ATT)

As described in Sect. 2, the IEEE 802.11 standards specify the transmission time in *error-free* channel conditions. Typically, the transmission time includes all the PHY and MAC overheads such as PHY preamble, SIFS, DIFS, slot time, ACK, and random backoff.

3.3.1 SampleRate

SampleRate [30] periodically sends a number of data frames as *sample* frames at a rate other than the current rate to gather statistics and to estimate whether another rate would provide better performance.

For each transmission rate, SampleRate calculates the ATT every ten seconds based on the transmission results, including frame retries. The estimated transmission time for a successful frame with one transmission attempt T_{succ} is computed according to Eq. (1). The transmission time for a successful frame but with n failed retries is calculated as $T = T_{succ} + T_{retry}$. T_{retry} is computed according to Eq. (7).

$$T_{retry} = n * (DIFS + T_{DATA} + T_{ACK\_TO}) + \sum_{i=1}^{n} BO_i \tag{7}$$

where BO_i is the random backoff before transmission and it doubles every time the frame fails; T_{ACK_TO} is the ACK timeout period; T_{DATA} and T_{ACK} are the transmission time of the data and acknowledgement frames respectively, which are calculated according to Eq. (2). The transmission time for a finally failed frame with n failed retries is computed according to Eq. (7). Here, n reaches the maximum retry count. The rate which has the smallest ATT is selected for use in the next control period.

Every tenth frame is treated as a sample frame to be transmitted at another rate, which is randomly selected from a set of available rates. Otherwise, it sends packets at the rate which has the lowest ATT. Experiments in [30] show that SampleRate performs similarly to, sometimes better than, the best of ARF, AARF and Onoe even on lossy links.

3.3.2 BEWARE

Wang et al. [31] proposed another rate control mechanism—Background traffic-aWAre RatE adaptation (BEWARE), which selects an optimal rate that has the lowest average transmission time. The authors argued that background network traffics should be considered when selecting a rate. SampleRate estimates the transmission time using Eq. (1), which does not consider the time that background traffic causes the CSMA/CA to freeze its counting down. BEWARE introduces a mechanism to measure this time and adds it to the total transmission time of a rate.

In the mechanism, the probability of a backoff slot to be idle/busy (P_0/P_1) is estimated. As the slot is busy, the backoff mechanism is frozen due to background transmissions. The duration for a frozen slot, T_1, is the average duration of which the medium is occupied by background traffic transmissions, which is calculated according to Eq. (2). It depends on the average frame size and transmission rates. When the slot is idle, the elapsed time, T_0, is only the slot duration. By summarizing the product of the probability of the slot state (idle/busy) and its corresponding duration, BEWARE is able to estimate the elapsed time for a slot as shown in Eq. (8).

$$T_{slot} = \sum_{i=0}^{1} P_i T_i \tag{8}$$

The number of slots taken into account depends on the number of failed attempts for a frame, as the CW doubles upon a transmission failure. By accumulating the elapsed time for each slot, the time spent on the backoff procedure is estimated. It is added with other time including DIFS, the frame transmission time T_{DATA}, SIFS and T_{ACK} to calculate the average transmission time, where T_{DATA} and T_{ACK} are calculated according to Eq. (2). In this way, BEWARE considers the time occupied by the background traffic and becomes load-aware.

3.4 Throughput

3.4.1 Minstrel

In Minstrel [5], the maximum achievable throughput (*TP*) of each rate is calculated periodically, and the rate that is capable to achieve the maximum throughput is selected. The throughput for the given channel conditions is calculated as

$$TP = P_{new} * \left(\frac{1sec}{T_{tx\_perfect}} \right) * FrameSize \tag{9}$$

where P_{new} is the weighted probability of success for the current rate adaptation window that is to be used by the rate selection process and is computed in Eq. (10), and $T_{tx\_perfect}$ is the time required to successfully deliver a frame in perfect channel conditions; therefore, $\frac{1sec}{T_{tx\_perfect}}$ is the number of packets that can be transmitted in one second window under the perfect channel conditions.

For every interval of 100 ms (default RAP length), Minstrel measures the statistics of frame delivery and computes P_{new} using the Exponential Weighted Moving Average (EWMA), which controls the balance of influence of both old and new packet delivery statistics.

$$P_{new} = (1 - \alpha) * P_{this\_interval} + \alpha * P_{previous} \tag{10}$$

where $P_{this\_interval}$ represents the probability of success for the interval before rate selection and is computed as the ratio of the number of successful transmissions against the number of attempts. $P_{previous}$ represents the moving average probability of success for the last interval. By default, the smoothing factor α is set to 75%, which means historical throughput measurements have more weight for new rate selection.

According to the mac80211 framework source code within the 2.6.35 Linux-wireless kernel, Minstrel calculates the perfect transmission time, $T_{tx\_perfect}$, as

$$T_{tx\_perfect} = SIFS + T_{Frame} \tag{11}$$

where *SIFS* is a type of inter-frame spacing defined by the IEEE 802.11 standards [12]; T_{Frame} is the transmission time of the data frame, and it is calculated as Eq. (2). For frame size, Minstrel uses a fixed size of 1200 bytes. We can derive from Eqs. (2) and (11) that Minstrel's throughput estimation depends on two parameters, the probability of success to deliver a frame and the bit rate it uses.

To account for retransmissions Minstrel uses the Multi-Rate Retry (MRR) chain, which proposes four candidate rates (i.e., *r*0, *r*1, *r*2 and *r*3) to attempt in case re-transmissions are necessary (its details are described in Sect. 2).

To determine the optimal rate for a given channel condition, Minstrel dedicates 10% of its traffic to probe the performance statistics of other rates by randomly selecting a rate (as *Lookaround rate*) that is not currently in use. For this 10% of data traffic, as shown in Table 2, the rate preferences are the best throughput rate (BTR), the random rate (RR), the best probability rate (BPR) and the base rate (BR) if the randomly selected rate (*RR*) is lower than the current best throughput rate (*BTR*); otherwise, they are the random rate, the best throughput rate, the best probability rate, and the base rate. For the other 90% of traffic (*normal packets*), the rate preferences are the best throughput rate, next best throughput rate (NBTR), the best probability rate, and the base rate.

As analysed in [13], in a collision environment, FLR based rate control mechanisms decrease the transmission rate to the lowest rate, which makes collisions even worse. Minstrel is robust in the collision environment because it does not decrease the rate. This is an advantage of throughput-based mechanisms over CTR or FLR based mechanisms.

3.4.2 Minstrel-rts

Although Minstrel is robust in a collision environment, hidden terminal collisions cause frame losses, which decrease the network performance. RTS/CTS is generally recognised as a good approach to mitigate hidden terminal collisions. However, it cannot be activated all the time due to the significant increase in overhead. To address the problem of hidden terminals and at the same time account for the overhead, Minstrel-rts [13] adaptively activates the RTS/CTS mechanism depending on whether using the mechanism achieves higher throughput. For *normal* data frames, Minstrel-rts turns off RTS/CTS for the best rate (i.e., r_0 in MRR). In a collision-free environment, the best rate is very likely to be successful, so RTS/CTS should be turned off for it. RTS/CTS is randomly turned on with a probability of 10% for all retry rates in the MRR. Retry rates are only used when the best rate fails. Frame loss may be due to hidden terminal collisions. Therefore, Minstrel-rts turns it on with a small probability to probe the performance statistics for having RTS/CTS turned on so that

Table 2 Retry preferences

Attempt	Lookaround rate		Normal
	RR < *BTR*	*RR* > *BTR*	rate
*r*0	BTR	RR	BTR
*r*1	RR	BTR	NBTR
*r*2	BPR	BPR	BPR
*r*3	BR	BR	BR

the hidden terminal problem can be detected earlier. For *lookaround* frames, RTS/CTS is not used for the random rate (higher than the current rate) because it will only be successful when the link quality increases. If this random rate fails, the frame will be sent with the best rate without RTS/CTS. The other retry rates use the RTS/CTS mechanism similarly to normal frames.

Figure 5 shows a state diagram for the collision-aware mechanism. When the wireless channel is collision free, data frames are sent according to the retry preference defined in the MRR; that is, within a RAP frames retry at a rate (e.g., r_i) for c_i times before attempting other lower rates. After every RAP, the throughput statistics of all supported rates are calculated (for both with or without RTS/CTS) to work out the best-performed rate for the next RAP. For each frame, RTS/CTS is turned on or off for each rate in the MRR chain, but in the rate adaptation period, many frames are transmitted with different MRR settings. Retry rates turn on RTS/CTS with a probability as discussed above. Therefore from a rate perspective, four pieces of information are gathered including attempts with RTS/CTS on, successes with RTS/CTS on, attempts with RTS/CTS off, successes with RTS/CTS off. Based on this information, for both with or without RTS/CTS, throughput is calculated. When a lower rate achieves higher throughput than the first rate (i.e., r_0), a rate drop request is issued. The throughput of the current rate for both with and without RTS/CTS is compared, and when the throughput for having RTS/CTS is greater than without it (i.e., $TP_{rts} \geq TP_{csma}$), it concludes that collision is occurring and the RTS/CTS mechanism is turned on in the next RAP rather than dropping the rate. The Collision Avoidance Window (CAW) specifies for how long the RTS/CTS mechanism should be used. It is initialised to one when the collision is detected and grows exponentially if the collision still exists when CAW expires. The maximum CAW is 16, considering that its large value could cause poor performance when the collision does not last long. As long as the CAW is bigger than zero, the mechanism is in the collision avoidance state. When CAW reaches zero, the mechanism enters the collision detection state, and the RTS/CTS will be turned off for one RAP. At the end of the RAP, the mechanism compares TP_{rts} and TP_{csma}. In the case when TP_{rts} is smaller than TP_{csma} there is no collision and rate adaptation follows the normal retry preference as described earlier. Otherwise, the CAW is doubled.

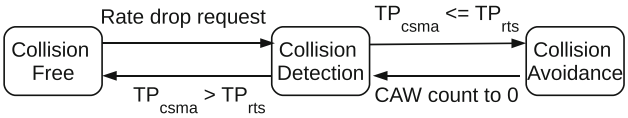


Fig. 5 State machine of the collision-aware mechanism

The throughput gain and the overhead of the RTS/CTS exchange are considered when deciding on a rate. The computation of TP_{rts} and TP_{csma} is based on Eq. (9) but when computing $T_{tx\_perfect}$, overhead (e.g, DIFS, ACK) is considered according to the IEEE 802.11 standards [12]. Specifically, the transmission time of the RTS and CTS frames is considered when RTS/CTS is utilized.

3.4.3 Rate control based on the estimated link capcity (RCELC)

RCELC rate control mechanism [32, 33] applies an accurate throughput estimation metric, Estimated Link Capacity (ELC), as shown in Eq. (12), to rate adaptation and picks up the highest throughput rate in every rate adaptation period.

$$ELC_{r_i} = \frac{MSDU}{T^{r_i}_{avg\_transaction}} \tag{12}$$

where *MSDU* is the MAC Service Data Unit (i.e., frame size), and $T^{r_i}_{avg\_transaction}$ is the average time spent on successfully delivering a frame using rate r_i in the last RAP, including all retransmissions. The averaged transaction time is calculated as

$$T^{r_i}_{avg\_transaction} = \frac{T^{r_i}_{total}}{Total\_successful\_attempt_{r_i}} \tag{13}$$

where $Total\_successful\_attempt_{r_i}$ is the number of successful transmission attempts in one RAP using rate r_i; while $T^{r_i}_{total}$ is the total time for all successful (i.e., $T^{r_i}_{success}$) and failed (i.e., $T^{r_i}_{failed}$) attempts of rate r_i in one RAP, that is the sum of time attempted to deliver a frame using rate r_i. The frame delivery time is calculated (either $T^{r_i}_{success}$ or $T^{r_i}_{failed}$) according to the CSMA/CA mechanism in the IEEE 802.11 standards as illustrated in Eqs. (1) and (7).

To compute the ELC metric, the statistics of frame delivery, including the attempted rates and their respective count, are gathered on the sender side based on the corresponding ACK frames arriving within the timeout. As it only requires sender-side information, the ELC metric incurs lower overhead than the approaches that require metric information from the receivers [9]. Besides, because it relies on per-frame delivery information, change of traffic load in the network has minimal impact on the accuracy of the ELC metric.

3.4.4 MiRA

MiRA [34] is a rate control mechanism designed for 802.11n networks which support both spatial-diversity-oriented-single-stream (SS) and spatial-multiplexing-driven-multiple-stream (double-stream, DS, in their platform).

The SS mode means that there is only one stream in transmission and the DS mode means that there are two streams being transmitted over the air simultaneously, so the 802.11n rates can be classified into the SS rates and DS rates. A metric called Sub-Frame Error Rate (SFER) is used to study the characteristic of the 802.11n rates and is defined as

$$SFER = \frac{nFrames * retries + nBad}{(retries + 1) * nFrames} \tag{14}$$

where *nFrame* is the number of sub-frames in the transmitted aggregated frame, *nBad* is the number of subframes received with errors, and *retries* is the number of hardware retries. The observation from extensive experiments is that *SFER* increases as rates increase when only the SS rates or DS rates are considered. However, when looking at the SS rates and DS rates together, the monotonicity does not hold.

MiRA uses a zigzag scheme to opportunistically probe the rate's goodput between the intra- and inter-mode. Within the intra-mode (either the SS or DS rates), it probes the goodput of rates higher and lower than the current rate until it reaches a rate with maximum goodput. When switching to the inter-mode, the probing mechanism only starts if significant changes occur in the measured moving average goodput of the current rate. The goodput is calculated as shown in Eq. (15)

$$G(t) = \frac{DATA * A(t)(1 - SFER)}{T_{overhead} + \frac{DATA*A(t)}{R}} \tag{15}$$

where DATA is the payload size of a MAC-layer frame, R is the bit rate. $T_{overhead}$ is the various 802.11 communication overhead including DIFS, SIFS and BlockACK. $A(t)$ is the moving average of the aggregation level (in terms of the moving average number of MAC Protocol Data Units (MPDUs) in an Aggregated MPDU (A-MPDU)). It is calculated according to Eq. (16), where $A'(t)$ is the aggregation level of the current probing frame.

$$A(t) = (1 - \alpha) * A(t - 1) + \alpha * A'(t) \tag{16}$$

MiRA also includes a collision aware mechanism to address the hidden terminal problem. The sender determines a collision loss if it satisfies the condition that the aggregate frame has experienced at least one retry and the loss ratio of its sub-frames is less than 10%. This information is discovered by various experiments [34]. For collision detection, an adaptive window is defined, which is initialized to 3. For any subsequent frame in the adaptive window, if it satisfies the condition again, MiRA confirms the existence of collisions and triggers the adaptive RTS/CTS mechanism to address collisions. Otherwise, the adaptive window will be decreased by one for each frame that does not satisfy the condition. RTS/CTS will be turned on only when $\frac{AFRAME}{R} \geq kT_{RCTS}$, where *AFRAME* is the aggregate frame size, *R* is the bit rate of rate R, T_{RCTS} is the transmission time of RTS and CTS frames, and *k* is a benefit-cost ratio, set to 1.5 in the prototype. *k* represents the minimum number of collisions needed to compensate for the cost of turning on RTS/CTS. If $\frac{AFRAME}{R} < kT_{RCTS}$, the adaptive window will be reset to zero and RTS/CTS is not triggered.

MiRA starts probing only when the goodput of the current rate changes significantly. This may not be a good reaction. If the channel condition degrades, the goodput of the current rate will be decreased sharply. This probing reaction works fine. However, in a scenario where the current rate is the optimal rate for a long time, and the frame loss ratio of the current rate is very low, the goodput of the current rate will not change significantly if the channel condition improves. Therefore, MiRA will not probe rates, and it will lose opportunities to improve throughput. In addition, comparing $\frac{AFRAME}{R}$ and kT_{RCTS} does not directly consider the benefits and drawbacks of the RTS/CTS mechanism, so it cannot guarantee that turning off RTS/CTS could provide a throughput gain. The benefit of RTS/CTS is mitigating frame losses, and the drawback of RTS/CTS is the RTS and CTS transmission overhead.

3.4.5 STRALE

The STandard-compliant and mobility aware PHY RAte and A-MPDU LEngth adaptation (STRALE) mechanism [35] is proposed for rate adaptation in mobile 802.11n/ac networks with the frame aggregation feature. In such a scenario, the channel quality decreases during an Aggregated MPDU (A-MPDU) reception due to mobility.

To address this issue, STRALE derives the optimal A-MPDU length in terms of time, T_O, after obtaining the BlockAck. By considering the sub-frame errors and adjusting the number of sub-frames in the last aggregation frame, the sender expects to find the optimal number of sub-frames to achieve the highest throughput for the last transmission. Therefore, T_O is computed according to Eq. (17).

$$\begin{cases} T_O = \sum_{i=1}^{n} \frac{l_i}{r} + T_{PHY} \\ n = \arg\min_{k \leq m} \left(\dfrac{\sum_{i=1}^{k} l_i * s_i}{\sum_{i=1}^{k} \frac{l_i}{r} + T_{PHY} + T_{MAC}} \right) \end{cases} \tag{17}$$

where l_i is the length of the *i*-th sub-frame, r denotes the physical rate. T_{PHY} represents the transmission time for the PLCP preamble and header, T_{MAC} is the MAC overhead including backoff and DIFS, and s_i indicates the binary transmission status of the i-th sub-frame. m is the total number of sub-frames, and n is the number of sub-frames which could achieve the maximum throughput for the last A-MPDU ($n \leq m$). The expected optimal length for the next A-MPDU, T_{next}, is computed based on an exponentially weighted moving average method, as denoted in Eq. (18)

$$T_{next} = (1 - \alpha) * T_p + \alpha * T_O \tag{18}$$

where T_P is the moving average length of the last A-MPDU, and $\alpha = min\left(\frac{T_p - T_O}{T_O}, 1\right)$. As the mobility increases, the channel quality decreases rapidly, and T_O could be much shorter than T_p. Therefore, more weight will be put on T_O. In this way, STRALE can quickly adapt to channel degradation. STRALE also compares the difference of the moving average A-MPDU lengths for the next transmission and the last transmission.

$$T_p - T_{next} > T_{MPDU} \tag{19}$$

where T_{MPDU} is the average transmission time of a single MPDU. It is calculated according to

$$T_{MPDU} = \sum_{i=1}^{m} \frac{l_i}{m * r} + T_{PHY} \tag{20}$$

If the condition in Eq. (19) is satisfied, it indicates that the difference is significant and the A-MPDU length should be decreased or the rate should be decreased. The throughput that can be achieved for both methods is compared according to Eq. (21) where r is the transmission rate, and T is the moving average of the A-MPDU length. If $THP(r\_, T_p) \geq THP(r, T_{next})$, the rate is increased from the current rate $r\_$ to the next higher rate r.

$$THP(r, T) = \frac{(T - T_{PHY})r}{T + T_{MAC}} \tag{21}$$

3.5 Signal noise ratio (SNR)

3.5.1 Receiver-based AutoRate (RBAR)

In RBAR [6], the RTS frame format is modified to carry the modulation rate and the size of the data frame instead of the duration of the channel reservation. Neighbouring nodes calculate the duration of channel reservation based on the rate and frame size information. Upon receiving an RTS frame, the receiver selects the appropriate transmission rate for the next data frame to be used by the sender based on the SNR measurement of the RTS frame. A simple threshold-based technique is adopted by the receiver to select an appropriate transmission rate, in which each rate has a lower SNR threshold and an upper SNR threshold. The rate selection is based on the comparisons of the estimated SNR and these thresholds. If the measured SNR falls between a rate's upper SNR threshold and lower SNR threshold, the rate is optimally chosen.

The receiver puts the selected rate into a CTS frame along with the data frame size and transmits the CTS frame back to the sender. Neighbouring nodes overhear the CTS frame and calculate the duration of the channel reservation. Thus the sender's neighbours may have a different value of the channel reservation duration from the receiver's neighbours. Accounting for this, RBAR includes the final channel reservation in a special sub-header of data frames, called the Reservation SubHeader (RSH).

The modification to RTS/CTS control frames has two goals. One is to provide a mechanism by which the receiver can communicate the selected rate to the sender. The other is to provide neighbouring nodes with enough information to calculate the duration of the requested channel reservation. However, the modification of the RTS and CTS frames causes compatibility problems to the standard, and the necessity of the RTS/CTS functionality introduces significant overhead, which is usually disabled in deployment. RBAR is compared with ARF in [6], and it shows better performance than ARF in a static channel and Rayleigh fading channel conditions.

The Opportunistic Auto Rate (OAR) protocol [36] is an extension for the MAC layer rate control mechanisms in which it sends multiple back-to-back data frames whenever the channel quality is good. As illustrated from experiments in [36], RBAR with the OAR extension achieves significant throughput gains compared with RBAR alone. However, this may introduce the fairness problem as a station can transmit multiple frames once it detects a good channel.

Another SNR based mechanism that also requires changes in the standard is Mutual Feedback (MutFed) [9]. In MutFed, the receiver measures SNR of frames that it receives from a sender. Every tenth frame, the receiver selects a rate by looking up a table based on SNR measurement and notifies it to the sender by transmitting an ACK at the proposed rate. However, the 802.11 standards say that ACKs must be transmitted at mandatory rates. Apart from this, it is hard to synchronize the receiver and the sender because the number of frame attempts at the sender is not equal to the number of frames received at the receiver if frame losses exist. Therefore, a sender may not be able to identify which ACK is the new rate notification except with additional information in the ACK, which again violates the standard.

Rate Adaptation in Mobile environments (RAM) [8] is also a receiver-based mechanism. To avoid the synchronization problem, the receiver uses the transmission rate of ACK to notify the sender about the selected rate for the next frame rather than the tenth frame. The same problem for RAM is that it modifies the ACK transmission rate. Another problem is that there may be a long period between the ACK of the last data frame and the next data frame so that the channel quality has already changed during that period of time.

3.5.2 Channel aware rate adaptation algorithm (CHARM)

RBAR and MutFed rely on information collected at the receiver and use existing mechanisms such as CTS or ACK packets to piggyback the rate information to the sender. Added information requires a change in the standard. CHARM [7] addresses this problem by allowing the sender to estimate the signal to interference and noise ratio (SINR) metric at the receiver side.

Within a wireless network, neighbouring nodes periodically exchange information regarding transmission power and noise level in the form of beacons, probe requests, and probe responses. CHARM makes use of this ongoing information to estimate the SINR value at a particular receiver and store the weighted moving average SINR value in a table. Before sending a frame, the sender checks a SINR threshold table for the intended receiver and determines a set of transmission rates. In the set, the minimal SINR threshold of each rate should be above the measured SINR. For the first transmission, CHARM selects the highest rate in the set while selecting a lower rate from a fast decreasing rate sequence for retransmissions. Each rate has a minimum required SINR, and the threshold is updated by observing frame success rate as a function of predicted SINR. A big advantage for CHARM over RBAR is that it does not require the RTS/CTS mechanism to coordinate rate control and thus eliminates overhead.

Similar rate control algorithms are SNR Guided Rate Adaptation (SGRA) [37] and Link Adaptation (LA) [38]. SGRA uses ACKs' RSSI from a receiver as a predictor of its data frame's RSSI at the receiver side. Thus, using the RSSI, SGRA calculates the SNR which is mapped to the frame delivery ratio (FDR) for each rate R. Then it selects the rate which has the maximum throughput computed by $FDR * R$. LA uses the RSSI of beacons and other frames from the receiver. The optimal rate is selected when the measured RSSI passes some thresholds. They both assume a symmetric link between the sender and receiver, which is not valid in wireless channels due to mobility, fading and interference.

3.5.3 Frequency-aware rate adaptation (FARA)

Rahul et al. [39] observe that different frequencies have different SNR over the same sender and receiver. FARA leverages the Orthogonal Frequency Division Multiplexing (OFDM) technology to divide the entire frequency band into subbands. Like RBAR, the SNR metric is monitored at the receiver side. However, a difference from RBAR is that FARA estimates the SNR for each OFDM subband for each sender. Besides, FARA maintains an SNR table that is similar to the one in CHARM. In the table, a minimal SNR threshold is defined for each transmission rate.

At the initialization time, the sender sends the first frame at the lowest rate on all subbands. The receiver measures the SNR for each subband. With it, FARA can determine a list of supported rates for each subband. The highest transmission rate is selected as the optimal. The MAC-layer ACK frame is modified to carry the selected transmission rates.

To reduce the overhead of rate feedback, FARA uses a two-bit field for each subband to indicate whether to increase/decrease the rate to the next level or stay at the current rate. This means that FARA adopts the 'Rate Ladder' approach. The sender and the receiver may be out of sync if the data frame or the ACK is lost. To address this problem, a sequence number is included in each ACK. The data frame includes a new field to notify the receiver about the sequence number of the last received ACK. Both the sender and the receiver track several recent rate states. If the receiver detects a frame/ACK loss, it puts the original ACK sequence number and the rate choices into the new ACK. In this way, the sender can update the rate correctly.

The novelty of FARA is that it transmits simultaneously multiple frames to a number of next-hops, i.e. one frame per next-hop. This is done by assigning each of these next-hops a non-overlapping OFDM subband. The allocation of frequency is achieved by a randomized greedy approach, which allocates to each next-hop the subband that has better performance for it than other next-hops.

3.5.4 SNR-aware intra-frame rate adaptation (SIRA)

In 802.11n networks, a long aggregated frame is utilized, which enables multiple sub-frames to be transmitted as a single frame, thus minimizing MAC overhead and improving throughput. However, transmitting a longer frame requires a longer frame duration. The maximum value is 10 ms. This is much longer than the channel coherence time in mobile environments. Experiments [40] reveal that sub-frames located at the latter part of the aggregated frame experience a higher sub-frame error rate if the whole aggregated frame is transmitted at the same rate. This motivates SIRA [41] to transmit the aggregated frame at multiple rates.

The metric used by SIRA is SNR. SNR is estimated based on information extracted from the PHY layer, i.e., the amount of Symbol Dispersion (SD), as the root mean squared value of the difference between the received symbol, $S_r(n)$, and the corresponding transmitted symbol, $S_t(n)$. Specifically, it is calculated according to Eq. (22), where N is the number of symbols in the frame, E is the energy level and N_o is the noise level.

$$SNR = \frac{E}{N_0} \approx \frac{1}{SD^2} = \frac{\frac{1}{N}\sum_{n=1}^{N}|S_t(n)|^2}{\frac{1}{N}\sum_{n=1}^{N}|S_r(n) - S_t(n)|^2} \tag{22}$$

It is difficult to obtain SD^2 from unknown data symbols, so SIRA leverages the known pilot symbol sequence. Each data symbol has a small number of pilot symbols. For example, the number of 20 MHz channels is 4. Let N_p represent the number of pilots per symbol. SIRA groups N_i pilots corresponding to N_s data symbols into a received pilot group where $N_i = N_p \times N_s$. Then, one aggregated frame is divided into K pilot groups. SIRA estimates the SNR for each group i, i.e., SNR_i $(1 < i < K)$ based on Eq. 22. Therefore, SIRA obtains an SNR distribution, SNR= $SNR_1, SNR_2 ... SNR_K$.

The notable idea for SIRA is that it selects two transmission rates for a single frame transmission, i.e. the primary rate R_p and the secondary rate R_s. The key point is to find the starting symbol I where to shift the transmission rate from R_p to R_s. Through experiments, Okhwan et al. [41] observe that the estimated SNR_i in SNR monotonically decreases during a frame reception in a mobile environment. Using this monotonicity feature of SNR, SIRA detects mobility. SIRA finds I from the beginning of the frame when the condition $SNR_i < SNR_{th}(R_p)$ is satisfied. $SNR_{th}(R_p)$ is the minimum SNR at which the theoretical BER of R_p is less than 10^{-4}. Then I is fed back to the sender.

The primary R_p is determined by inter-frame rate control mechanisms, like SGRA. The secondary rate R_s is chosen by table look-up. Each row in the table specifies R_p and the corresponding R_s and SNR_{th}. In [41], the performance of SGRA [37] and SGRA+SIRA is evaluated, and the results demonstrate that SIRA can improve SGRA's throughput significantly in mobile environments. One limitation of SIRA is that it can only determine two rates for an aggregated frame, which may not be enough for a fast-changing channel.

3.6 Effective SNR

3.6.1 Effective SNR (ESNR)

SNR does not capture the effect of frequency-selective fading. ESNR [42] computes the effective SNR to address this issue. In 802.11n networks, the OFDM technology is used. In these networks, 20 MHz or 40 MHz channels are divided into 312.5 kHz bands named subcarriers. Each of the subcarriers sends independent data simultaneously. Each subcarrier in a frame is modulated equally, using BPSK, Quadrature Phase Shift Keying (QPSK), Quadrature Amplitude Modulation (QAM)-16 or QAM-64, with 1, 2, 4 or 6 bits per symbol, respectively. The transmission rate depends on the modulation and coding.

Effective SNR is not the average subcarrier SNR. The 802.11n CSI is used to calculate the effective SNR. The CSI is a collection of M*N matrixes H_s. Each matrix represents the RF path (SNR and phase) between all pairs of N transmitting and M receiving antennas for one subcarrier s. Let SNR_s be the SNR for each subcarrier s. ESNR calculates the average BER across all subcarriers as displayed in Eq. (23)

$$BER_{eff,k} = \frac{1}{52}\sum BER_k(SNR_s) \tag{23}$$

where k is the modulation rate. Then ESNR [42] uses the inverse mapping from BER to get the effective SNR as denoted in Eq. (24)

$$SNR_{eff,k} = BER_k^{-1}(BER_{eff,k}) \tag{24}$$

Either the sender or receiver can perform the effective SNR calculation. For a sender to make rate decision, the receiver's effective SNR thresholds for different rates will be sent to the sender during association. The sender uses two methods to get the up-to-date CSI. The receiver sends it to the sender, or the sender estimates it from the reverse path. Given the recent CSI, the highest rate that is predicted to have a frame delivery ratio above 90% is selected.

3.6.2 In-frame rate adaptation (InFRA)

The novelty of InFRA [40] is that a frame is divided into multiple groups of symbols (GOS) and the sender selects the transmission rate for each GOS.

In InFRA, each GOS includes 5 OFDM symbols. Suppose the estimated SNR at the n^{th} subcarrier of the s^{th} symbol in a GOS is $\gamma_{s,n}$. Let $BER_{eff,k}$ denote the average BER for the GOS using transmission rate r_k. The transmission rate in 802.11n depends on the modulation, coding and the number of streams. $BER_{eff,k}$ is calculated as

$$BER_{eff,k} = \frac{1}{5N_{sc}}\sum_{s=1}^{5}\sum_{n=1}^{N_{sc}} BER_k(\gamma_{s,n}) \tag{25}$$

where N_{sc} is the number of subcarriers in an OFDM symbol. $BER_k(\gamma)$ is a bit error rate function using transmission rate r_k for SNR γ. Table 3 displays $BER_k(\gamma)$ as a function of r_k, where Q is the standard normal Cumulative Distribution Function (CDF) [43]. Then, using an inverse mapping function, the measured BER is mapped into the effective SNR, γ_{eff}.

Table 3 Bit error rate versus SNR γ

Modulation	Bits/symbol	$BER_k(\gamma)$
BPSK	1	$Q(\sqrt{2\gamma})$
QPSK	2	$Q(\sqrt{\gamma})$
QAM-16	4	$\frac{3}{4}Q\left(\sqrt{\frac{\gamma}{5}}\right)$
QAM-64	6	$\frac{7}{12}Q\left(\sqrt{\frac{\gamma}{21}}\right)$

$$\gamma_{eff} = BER^{-1}(BER_{eff,k}) \tag{26}$$

Then, InFRA selects r_i as the best rate for the next GOS when $\beta_i < \gamma_{eff} < \beta_{i+1}$ where β_i is the minimal SNR for transmission rate i.

These GOSs are transmitted to the receiver sequentially. The receiver decodes every GOS and uses the Cyclic Redundancy Check (CRC) checksum embedded in the pilot subcarriers of the GOS to determine whether it is delivered successfully. An ACK/Negative ACK (NACK) will be sent via a separate channel to the sender depending on whether a GOS is successful or not. Negatively acknowledged GOS will be retransmitted until it is successful.

After decoding each GOS, the receiver selects the best transmission rate based on the calculated SNR and sends it back to the sender via ACK or NACK. After all GOSs are received successfully, the receiver sends the end-of-stream (EOS). This method avoids frame-level retransmission, and therefore reduces retransmission overhead. However, in order for the receiver to decode the GOS correctly, InFRA encodes six bits of the transmission rate into the GOS header. This is a type of added communication overhead. Also, the requirement of using another channel for ACK/NACK also consumes precious frequency resources.

3.6.3 TurboRate

In a Multi-User MIMO (MU-MIMO) networks, a client has a single-antenna, and the AP is installed with multiple antennas. Clients transmit data frames to the AP simultaneously. For example, Fig. 6a shows that the blue client and the red client are transmitting to the AP concurrently. To decode the blue client's signal, the AP projects its signal on a direction that is orthogonal to the red client's signal, as shown in Fig. 6b. Always, there is an angle between the two signals, θ. Thus, the projection reduces the SNR, and its value depends on the original SNR value and the angle. These two parameters vary from frames to frames [44]. Traditional rate adaptation mechanisms fail to consider this, and the rate should be adapted on a per-frame basis.

TurboRate [44] is proposed for uplink (clients to an access point) transmissions in this scenario. By listening to the AP's transmissions, the client calculates two variables: (1) the direction along which the client's signals arrive at the AP, and (2) its SNR at the AP if it is transmitted alone, i.e. the SNR without projection.

Consider the scenario in Fig. 6a. The direction of a client's signal at the AP is obtained by the channel vector of a client. For example, the direction for the blue client is $h_b = (h_1, h_2)$. The elements, h_1 and h_2, are obtained based on the channel reciprocity feature [45] that the forward channel has the same transmission property with the reverse channel, as the forward and backward electromagnetic waves transmit the same way. Therefore, the client measures the direction vector from the AP's transmissions as its own direction vector. The SNR for a client is calculated according to Eq. (27)

$$SNR = \|h\|^2 P/N \tag{27}$$

where h is the channel vectors by measuring the AP's transmissions passively, P is the transmission power of the client, and N is the AP's noise level. N is broadcasted in beacons.

When multiple clients are contending for transmissions, the client that gains the channel first puts the direction information, from which the AP receives its signal, in a special header of the data frame. A client that attempts to transmit simultaneously projects its own signal orthogonal to the first client based on the direction information and calculates the SNR reduction due to projection. The client computes the effective SNR using the same method proposed in [42]. It then looks up the SNR-bitrate table [39]. A set of rates are determined if the measured SNR is higher than their minimal SNR thresholds. Only the highest rate in the set is selected as the current rate. Additionally, the following clients that want to concurrently transmit calculate the rate using the same method.

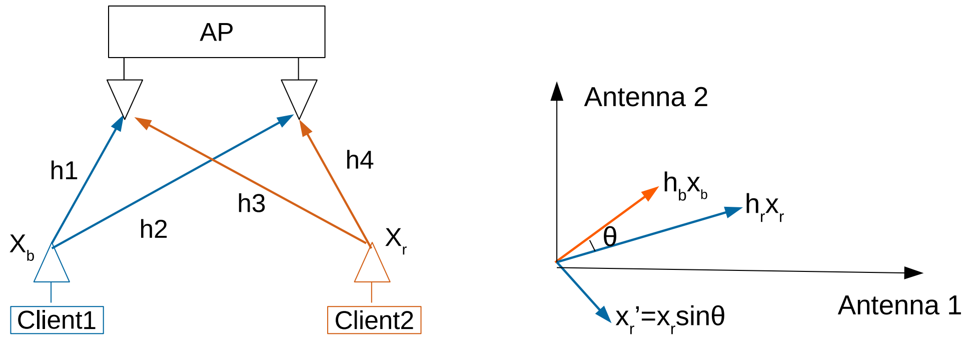
3.7 Bit error rate (BER)

3.7.1 SoftRate

SoftRate [22] estimates the bit error rate (BER) metric which is calculated over each received frame to adjust the transmission rate.

The average BER of the channel at the current transmission rate $R_{current}$ is calculated over all bits in a frame $BER(R_{current})$, while the prediction of BER at other transmission rates is based on two heuristics. One is that the BER is a monotonically increasing function of the rate at

Fig. 6 Simultaneous transmissions and decoding



(a) Red and Blue Clients send simultaneously

(b) Projection makes SNR change

any given SNR. The other is that a rate's BER at a given SNR is at least a factor of 10 higher than that of the next-lower rate [22].

Based on the two heuristics, SoftRate calculates BER for multiple rates and computes optimal thresholds for each rate. For each rate, SoftRate calculates the lower and upper limit, BER_{lower} and BER_{upper} respectively. The SoftRate selects the next rate R_{next} as

$$R_{next} = \begin{cases} R^{-}_{current}, & \text{if } \; BER(R_{current}) \geq BER_{upper} \\ R^{+}_{current}, & \text{if } \; BER(R_{current}) \leq BER_{lower} \\ R_{current}, & \text{otherwise} \end{cases}$$

In addition, if the measured BER is far from the lower or upper limits, SoftRate jumps multiple levels to improve the responsiveness of significant change in channel conditions. However, the details of the rate jump are not described in [22]. The BER metric is measured at the receiver and sent back to the sender to assist with the rate decision. This requires modification to the MAC-layer frames and can cause compatibility issues.

3.7.2 AARC

A posteriori Bit Error Probability (ABEP) is proposed in [46], which is a similar metric to BER. ABEP is the error probability of each bit based on the received wireless signal symbols. The ABEP-enabled Adaptive Rate Control (AARC) mechanism is based on this metric.

For a received frame, the receiver calculates the ABEP metric. If the frame is received correctly, the receiver decides whether to increase, decrease, or maintain the rate based on the comparison of the measured ABEP and two thresholds. These two thresholds are defined for each rate separately. If the measured ABEP is above the higher threshold, the rate is decreased. If the calculated ABEP is below the lower threshold, the rate is increased. Otherwise, the rate is unchanged. The rate decision is notified back to the sender. If the frame is received incorrectly, the receiver determines if it is due to collisions or fading and notifies the sender. ABEP is used to differentiate collisions and fading. For collisions, ABEP will experience an upsurge when collisions start, and the upsurge will last for a relatively long period until the collisions finish. Slow fading has no upsurge. Fast fading has an upsurge, but it does not last long.

3.8 Frame error rate (FER)

A rate control mechanism [47] is proposed to support the IEEE 802.11e quality of service (QoS). Rate adaptation is conducted according to the FER metric [48]. The FER metric is calculated based on measured SNR and BER, as shown in Eq. (28).

$$FER = 1 - (1 - BER(SNR))^{N} \tag{28}$$

where N is the number of bits of the received frame. It increases the transmission rate when the number of consecutive ACKs reaches ten. It decreases the current rate r_h to its next lower rate r_l when the measured FER exceeds the threshold $e_h = 1 - \frac{r_l}{r_h}(1 - e_l)$ where e_l is the FER of r_l, and $r_1(r_h)$ is the low (high) bit rate. This guarantees that decreasing the rate will have a higher throughput than staying at the higher rate.

If a transmission fails and the current FER is above e_h, it means that the frame is corrupted, and the retransmission will use the next lower rate. The CW is not doubled for the retransmission to improve throughput. However, this violates the 802.11 standards.

This mechanism gradually increases and decreases the transmission rate, so it may respond slowly to channel improvement and degradation. The channel quality metric calculation is more complex than any mechanism that is based on a sole metric. To obtain the FER metric, SNR and BER need to be calculated first, but this implies that the mechanism is more accurate in channel estimation.

3.9 Combined metrics

3.9.1 Hybrid automatic rate control (HARC)

In contrast to the mechanisms we have described so far, HARC [49] bases its rate control decisions on multiple metrics, including throughput, FLR, and SNR. The key component of HARC is a throughput based controller. The controller generates the optimal rate and decides whether to probe adjacent rates. The rate decision made by the core controller can be overwritten by an SNR-based controller. For each rate, the SNR controller measures the Signal Strength indication of the Acknowledged frames (SSIA). The SNR-based controller has a look-up table. In this table, each rate has three SSIA thresholds of which two are for stable channel conditions (a high and a low threshold), and an additional low threshold is for dynamic link conditions. The low threshold for the stable channel refers to the one which can provide acceptable performance (defined as FLR $< 10\%$). The dynamic channel condition is detected by comparing the last three consecutive SSIA values, $SSIA_1$, $SSIA_2$ and $SSIA_3$. If both $SSIA_2 - SSIA_1$ and $SSIA_3 - SSIA_2$ exceed a predefined value, a dynamic channel is detected, and the corresponding low threshold is used. Otherwise, the stable channel low threshold is used. Both the static and dynamic channel use the same high threshold. In HARC, there are other two threshold rates, $R_{upbound}$ and $R_{lowbound}$, which provide a bound for the minimum and maximum rate to be used. Their values are determined by the corresponding SSIA values. If the newly selected rate is below the $R_{lowbound}$, $R_{lowbound}$ is used as the new rate and vice versa for the $R_{upbound}$.

In [49], the authors do not compare HARC with other rate control mechanisms. However, HARC is based on many thresholds and parameters. We doubt the ability of this mechanism to find a set of parameters that will allow HARC to achieve the maximum performance in all scenarios. We learned from the PID algorithm that this task is challenging.

4 Analysis and comparison

In this section, we compare the rate control algorithms in six aspects: (1) metric selected by the mechanism, (2) how the rate is updated—gradual adaptation or best rate selection, (3) where the channel quality evaluation and rate decision are made, at the sender or the receiver side, (4) when the rate is updated—per-frame adaptation or based on rate adaptation period, (5) what the rate adaptation algorithm is, and (6) how mechanisms perform. The comparisons are summarized in Tables 4 and 5.

4.1 Metrics

As shown in Table 4, various metrics are utilised in rate control mechanisms, including CTR, FLR, transmission time, throughput, SNR, effective SNR, BER and combined metrics. For the CTR and FLR metrics, the calculation is simple, and only some registers are required to maintain the transmission status. They are updated based on the transmission failure/success of the last frame. One drawback of these metrics is that the bit rate information of a transmission rate is not taken into account. In a hidden terminal environment, all transmission rates have severe frame losses [13]. In response to those losses, CTR-/FLR-based mechanisms decrease the rate. However, in this scenario, high rates should be preferred, as with similar frame losses, they have better throughput compared to lower rates due to the bit rate difference. Similarly, higher rates should also be preferred in noisy environments with significant channel errors. Transmission time and throughput are a bit more complicated than CTR and FLR. To calculate these metrics, knowledge of physical layer features is required including symbol interval and PLCP header preamble duration. The bit rate and the loss ratio of the transmission rate are also considered. Therefore, even with a high loss ratio, the high rate will be selected if it can achieve the best throughput or lowest transmission time. SNR is not directly connected to performance. Some research findings show that SNR is directly related to loss ratio in controlled environments, but this does not hold in realistic environments. This means that SNR is not a good indicator of the throughput. Effective SNR is a new metric, and the calculation requires the BER information which is not connected to throughput as well. For example, in a collision environment, a collided frame can have a very high BER, but a single frame loss has little impact on the aggregate throughput.

According to the strategy to gather link quality metrics, rate control mechanisms can be categorised into three classes. The first is a probing and testing based approach which sends a number of data frames at an attempted rate other than the current rate to test how the attempted rate works, such as ARF, AARF. The benefit of this approach is that these rate control mechanisms know how two of these available rates (current and adjacent higher rate) perform. However, it lacks a global view of the rate performance. In some cases, e.g., 802.11n networks [34], the adjacent rate may not perform better than the current rate, but other rates may do. This approach could fail in these cases. The second scheme is a non-probing approach which uses transmission results of the current rate to predict an appropriate transmission rate for the future, such as PID, RRAA and SoftRate. The problem with this approach is that it should

Table 4 Classification of MAC-layer rate control mechanisms

Mechanism	Network	Classification factors			Adaptation strategy
		Metric (Where)	How	Duration	
AARC [46]	802.11abg	ABEP (S)	Gradual	Per-frame	$R^{\uparrow}$ if $ABEP(R_{current}) < threshold_{lower}$, $R^{\downarrow}$ if $ABEP(R_{current}) > threshold_{upper}$
AARF [25]	802.11abg	CTR (S)	Gradual	Per-frame	Same as ARF, dynamically adapt the α threshold to achieve convergence
AMRR [25]	802.11abg	FLR (S)	Gradual	RAP	$R^{\uparrow}$ if $FLR < 10\%$, $R^{\downarrow}$ if $FLR > 33\%$
STRALE [35]	802.11n/ac	Throughput (S)	Gradual	Per-frame	$R^{\uparrow}$ if $Throughput(R^{+}_{current}) > Throughput(R_{current})$, $R^{\downarrow}$ if $Throughput(R^{-}_{current}) > Throughput(R_{current})$
ARF [24]	802.11abg	CTR (S)	Gradual	Per-frame	$R^{\uparrow}$ if $C_{success} \geq 10$, $R^{\downarrow}$ if $C_{fail} \geq 2$
BEWARE [31]	802.11abg	T_{tx} (S)	Best rate	RAP	Select rate R, such that $min(T_{tx}(R))$
CARA [16]	802.11abg	CTR (S)	Gradual	Per-frame	Improve ARF with RTS probing and CCA detection to address collisions
CHARM [7]	802.11abg	SNR (S)	Best rate	Per-frame	Using measured SNR to find all supported rates from a table listing the minimal SNR for each rate and select the highest one
ERA [27]	802.11abg	CTR (S)	Gradual	Per-frame	Same as AARF but uses fragmentation to address hidden terminal collisions
ESNR [42]	802.11n	Effective SNR(R)	Best rate	Per-frame	Given measured effective SNR, computes the highest rate of which the packet delivery ratio is greater than 90%
FARA [39]	802.11abg	SNR (R)	Gradual	Per-frame	Using a similar method as CHARM, but it gradually increases/decreases the rate to the optimal rate. It also enables multiple simultaneous transmissions over multiple bands.
HA-RRAA [28]	802.11abg	FLR (S)	Gradual	RAP	Same as RRAA, but prolongs the time a rate is used when selected to avoid rate oscillation, uses a shorter window to monitor FLR to make it responsive to channel degradation, and considers RTS overhead to enhance efficiency in addressing collisions.
HARC [49]	802.11abg	SNR, TP (S)	Best rate	RAP	Select rate R, such that $max(TP(R))$ while using measured SNR to verify whether it is applicable
InFRA [40]	802.11n	Effective SNR (R)	Best rate	Per-frame	Different rates will be used for different parts, called GOS, of a frame. Using measured effective SNR,γ_{eff}, it selects r_i as the best rate for next GOS when $\beta_i < \gamma_{eff} < \beta_{i+1}$ where β_i is the minimal SNR for transmission rate i
LA [38]	802.11abg	SNR (S)	Best rate	Per-frame	The optimal rate is selected when the measured SNR passes some thresholds
Minstrel [5]	802.11abg	TP (S)	Best rate	RAP	Select rate R, such that $max(TP(R))$
Minstrel-rts [13]	802.11abg	TP (S)	Best rate	RAP	Same as Minstrel, but added RTS probing to address hidden terminal problems
MiRA [34]	802.11abg	TP (S)	Best rate	RAP	Select rate R, such that $max(TP(R))$
MutFed [9]	802.11abg	SNR (R)	Best rate	Per-frame	Rate selection is based on a table lookup, where the rate selected is most appropriate at the given average SNR
Onoe [29]	802.11abg	FLR (S)	Gradual	RAP	$R^{\uparrow}$ if credit ≥ 10, $R^{\downarrow}$ if FLR $> 50\%$
PID [4]	802.11abg	FLR (S)	Best rate	RAP	$R_{current} \leq R_{new} \leq (R_{current} + adj)$ if $adj > 0$; $(R_{current} + adj) \leq R_{new} \leq R_{current}$ if $adj < 0$
PIDE [15]	802.11abg	FLR, TP (S)	Best rate	RAP	Same as PID, but switching rate only if $TP(R_{new}) > TP(R_{current})$
RAM [8]	802.11abg	SNR (R)	Best rate	Per-frame	Maintain a throughput-vs-(rate, SNR) table. Using measured SNR, RAM looks up the table and selects the rate R that can maximise throughput.
RBAR [6]	802.11abg	SNR (R)	Best rate	Per-frame	Select rate R, such that $threshold_{upper} > SNR(R) > threshold_{lower}$
RCELC [32]	802.11abg	TP (S)	Best rate	RAP	Select rate R, such that $max(TP(R))$

Table 4 (continued)

Mechanism	Network	Classification factors			Adaptation strategy
		Metric (Where)	How	Duration	
RRAA [26]	802.11abg	FLR (S)	Gradual	RAP	$R^{\uparrow}$ if $FLR < threshold_{lower}$, $R^{\downarrow}$ if $FLR > threshold_{upper}$
SampleRate [30]	802.11abg	T_{tx} (S)	Best rate	RAP	Select rate R, such that $min(T_{tx}(R))$
SGRA [37]	802.11abg	SNR (S)	Best rate	Per-frame	Use measured SNR to get the frame delivery ratio (FDR) for each rate R, then choos 10 e the rate which has the maximum throughput computed by $FDR * R$
SIRA [41]	802.11n	SNR(R)	Best rate	Per-frame	Two rates will be chosen for two parts of a frame from looking up a table with each row listing two rates and correspondent SNR
SoftRate [22]	802.11abg	BER (S, R)	Best rate	Per-frame	$R^{\uparrow}$ if $BER(R_{current}) < threshold_{lower}$, $R^{\downarrow}$ if $BER(R_{current}) > threshold_{upper}$
TurboRate [44]	802.11ac	Effective SNR (S)	Best rate	Per-frame	Using measured effective SNR to find all supported rates from a table listing the minimal effective SNR for each rate and select the highest one

have a mechanism to predict how other rates perform. Otherwise, blindly increasing the rate could cause the rate oscillation problem [15]. The third approach is a hybrid approach, which is not only based on the transmission results of the current rate but also uses probing data frames to test the network conditions, such as Minstrel. The benefit of this approach is that it has a global view of how all rates perform so that it is robust in collision environments [13]. The problem for this approach is that it uses a number of trial frames to be transmitted at some rates other than the best throughput rate, which adds overhead.

4.2 Adaptation decision

According to where the link quality metric is measured and the rate decision is made, rate control mechanisms are classified into three groups: sender, receiver or hybrid based approaches.

CTR, FLR, transmission time, and throughput based mechanisms, including ARF, RRAA, and Minstrel, are in the sender based group because these metrics are calculated based on received acknowledgements. Some SNR-based mechanisms are sender based, e.g., CHARM. In such mechanisms, the sender predicts the SNR at the receiver side based on measured RSSI and transmission power indicated in the probing packets. Those mechanisms assume an asymmetric link between communicating pairs. Unfortunately, it is not valid in many realistic environments due to different noise and interference levels. Other SNR-based mechanisms, e.g., RBAR, are receiver-based. This can enhance the measurement accuracy of the SNR. However, getting the rate feedback back to the sender requires modification of the current frame format, which violates the standard. In the hybrid approach, the metric estimation and rate selection are made on different sides. SoftRate is such an example. The receiver estimates the BER which is returned to the sender for rate selection.

4.3 "Rate ladder" or direct rate selection

According to how the rate is adapted, we can classify rate control mechanisms into two schemes. The first strategy is called a gradual rate adaptation (or *rate ladder*) scheme in which the transmission rate is increased or decreased to the next level, whereas the second is the best rate selection scheme which directly selects the best rate.

As shown in Table 4, most of the CTR and FLR based mechanisms, including ARF, AARF, Onoe, AMRR, and RRAA, take the first scheme and attempt to upgrade to the next higher lever rate whenever the channel is good. For CTR based mechanisms, a good channel is inferred by ten consecutive successful transmissions while for FLR based mechanisms, it means that the measured FLR is less than a threshold. If the channel condition is bad, gradual rate adaptation mechanisms decrease to the next lower level rate. A bad channel is detected by two consecutive frame failures or when the measured FLR is bigger than another threshold. SNR, BER, transmission time, and link capacity estimation based mechanisms, including SampleRate, RBAR, and SoftRate, adopt the second method and are able to select the best rate in only one step.

The best rate selection is more responsive than gradual rate selection when the channel suddenly changes or the rate control mechanism starts or restarts. The best rate

Table 5 Performance comparison as reported in literature

Env	Experiment settings	Comparison ($A < B$: B achieves a higher throughput compared to A)	References
SIM	Slow fading	SampleRate < RRAA < CHARM < RBAR < SoftRate	[22]
	Fast fading	RBAR < SampleRate < RRAA < SoftRate	
	Hidden terminal	RRAA < SoftRate(*nid*) < SampleRate < SoftRate(*id*)	
SIM	Slow fading	Minstrel < RRAA < CARA < AARC	[46]
SIM	Mobile	ARF < AARF < MutFed	[9]
SIM	Vary contending stations	ARF < ARF-RTS < BEWARE < CARA	[31]
	Ricean fading	ARF < CARA < ARF-RTS < BEWARE	
SIM	Walking speed from 0 to 1m/s	SGRA < SGRA+SIRA	[41]
SIM	Static, mobile, contention	ARF < CARA < ERA	[55]
EMU	Walking speed at 0.5m/s	Onoe < SampleRate < AMRR < CHARM	[56]
	Walking speed at 1, 2m/s	Onoe < AMRR < SampleRate < CHARM	
	Hidden terminal	AMRR < Onoe < SampleRate < CHARM	
EMU	Human speed mobile	SoftRate < SampleRate < ESNR	[42]
	Fast mobile		
EMU	Vary interference load	PID < PIDE < Minstrel < Minstrel-rts	[13]
EMU	Static	PID < Minstrel < PIDE	[15]
	Channel quality linearly increases		
	Channel quality linearly decreases		
	Channel quality sudden changes		
EMU	Static	Minstrel < RCELC	[32]
	Channel quality linearly increases	Minstrel < RCELC	
	Channel quality linearly decreases	Minstrel < RCELC	
	Channel quality sudden changes	Minstrel < RCELC	
EMU	0.1 ms < coherence time < 1 ms	RBAR < RBAR with OAR < ARF < RRAA	[57]
	1 ms < coherence time < 0.1 s	RBAR < ARF < RRAA < RBAR with OAR	
EMU	Static	Onoe < AMRR < Minstrel < SampleRate	[58]
	Vary interference strength	Onoe < AMRR < SampleRate < Minstrel	
	Vary interference interval	Onoe < AMRR < SampleRate < Minstrel	
	Vary interference duration	Onoe < AMRR < SampleRate < Minstrel	
	Vary hidden terminal load	Onoe < SampleRate < AMRR < Minstrel	
	Vary hidden terminal link quality	Onoe < SampleRate < AMRR < Minstrel	
OTA	100 MHz channel	SampleRate < FARA	[39]
	20 MHz channel		
OTA	802.11a, UDP	RRAA < SampleRate < ARF < HA-RRAA	[28]
	802.11a, TCP (4 flows)	SampleRate < ARF < RRAA < HA-RRAA	
	802.11a, TCP (1 flow)	SampleRate < RRAA < ARF < HA-RRAA	
	802.11a, mobile, UDP	SampleRate < RRAA < HA-RRAA < ARF	
	Hidden terminal	ARF < SampleRate < RRAA < HA-RRAA	
	2.4 GHz channel	ARF < SampleRate < RRAA < HA-RRAA	
	5 GHz channel	SampleRate < ARF < RRAA < HA-RRAA	
OTA	Indoor	ARF < AARF < Onoe < SampleRate	[30]
	Outdoor		
OTA	TCP, static, 802.11a	ARF < AARF < SampleRate < RRAA	[26]
	UDP, static, 802.11a	ARF < AARF < SampleRate < RRAA	
	UDP, static, 802.11b	ARF < AARF < SampleRate < RRAA(A-RTS) < RRAA	
	UDP, mobility, 802.11b	SampleRate < AARF < ARF < RRAA(A-RTS) < RRAA	
	UDP, hidden terminal, 802.11b	AARF < SampleRate < ARF < RRAA(A-RTS)	

Table 5 (continued)

Env	Experiment settings	Comparison ($A < B$: B achieves a higher throughput compared to A)	References
OTA	5 GHz, static, UDP	Athreos MIMO RA < RRAA < SampleRate < MiRA	[59]
	5 GHz, static, TCP	SampleRate < Athreos MIMO RA < RRAA < MiRA	
	2.4 GHz, static, UDP	SampleRate < RRAA < Athreos MIMO RA < MiRA	
	UDP, mobility	SampleRate < Athreos MIMO RA < RRAA < MiRA	
	TCP, mobility	SampleRate < Athreos MIMO RA < RRAA < MiRA	
	Hidden terminal	Athreos MIMO RA < SampleRate < MiRA(*nid*) < RRAA < MiRA	
OTA	Static	Onoe < ARF < CHARM < RRAA < SampleRate < AMRR < RAM	[8]
	Mobile (walking)	Onoe < SampleRate < AMRR < ARF < RRAA < CHARM < RAM	
	Mobile (driving)	Onoe < AMRR < ARF < SampleRate < RRAA < CHARM < RAM	
	Mobile, indoor	AMRR < Onoe < ARF < SampleRate < RRAA < CHARM < RAM	
	Indoor with interference	Onoe < AMRR < ARF < SampleRate < RRAA < CHARM < RAM	
OTA	UDP 802.11a, static	ARF < RRAA < SGRA	[37]
	Hidden terminal	ARF < RRAA < SGRA	
	2.4 GHz interference	RRAA < SGRA	
OTA	Indoor, minor contention	Onoe < AMRR < SampleRate	[60]
	Indoor, high contention	AMRR < Onoe < SampleRate	
	Outdoor, directional	Onoe < AMRR < SampleRate	
	Outdoor, omni-directional	AMRR < Onoe < SampleRate	

SIM, simulation; EMU, emulation; OTA, over-the-air; Underline: mechanism proposed in Ref; (*id*)/(*nid*), with/without interferer detect

selections are likely to perform better than gradual rate selections in a hidden terminal environment because high frame losses caused by collisions trigger those mechanisms to decrease the rate to the lowest rate. Gradual rate selection mechanisms (FLR based) stick to the lowest rate and cannot increase the rate due to the high FLR. Usually, a higher rate performs better than a lower rate in hidden terminal environments because the higher rate has less channel air time and less collision probability. For the link capacity based best rate selection mechanisms, e.g., Minstrel, it is likely that the best throughput rate is selected [13] as the current statistics of all rates are available.

4.4 Per-frame or adaptation window

According to how long the selected rate will be used, we can classify rate control mechanisms into two schemes. The first strategy is called Per-frame rate adaptation in which the transmission rate is decided for only the next frame, whereas the second is the RAP-based scheme which uses the selected rate for the next RAP.

As shown in Table 4, most rate control mechanisms on the Linux system, including Minstrel, PIDE, and RCELC, are RAP-based mechanisms. On the Linux system, there is no dedicated processor to handle frame transmission and reception. Therefore, the transmission statistic cannot be gathered and processed as soon as it is available. RAP-based mechanisms may not be responsive to very fast channel variations within a frame transmission time. However, the rate selection in those mechanisms is more stable because they make rate decisions based on the averaged statistics over a rate adaption period, On the other hand, rate control mechanisms designed on the SDR platform, e.g. SoftRate, can achieve per-frame rate selection. They are responsive to fast channel changes. However, they require specialized hardware to achieve this feature.

4.5 Experimental comparison

In this section, we briefly describe evaluation environments (simulation, over-the-air, and controllable platforms) used for the evaluation of MAC layer rate control algorithms and also show which rate control approaches were evaluated/compared in particular evaluation environments.

In simulations, a network simulator, e.g. NS3 [50] is usually utilized to simulate and predict the behaviour of a network without the presence of an actual network. Many rate control mechanisms are available on the NS3 platform, including AARF, AMRR, ARF, CARA, RRAA, Minstrel, etc. It is easy to write scripts. Besides, network simulations are also inexpensive, as some simulators, eg. NS2/NS3, are free to use and still some simulators, e.g. Qualnet [51] provide a trial version for academics. In addition, simulators provide engineers and researchers with the means to test network scalability that might be extremely difficult using real hardware, for instance, simulating a large network with hundreds of computer nodes. Another advantage of simulations is that the results are repeatable. Using the same configuration and parameter setting, experimental results can be reproduced. However, simulations also have some drawbacks. For example, there are no actual stations or wireless signals in the simulated networks and the simulation is based on very simple noise models. Those models cannot accurately simulate the real environment dynamics, such as the movement of obstacles (e.g. moving vehicles) and the presence of various types of interference (e.g. microwave).

NS3 cannot simulate some hardware features. For example, NS3 does not provide a data structure for the multi-rate-retry chain. In the mac802.11 framework, a Linux Kernel component, which integrates common features, e.g., rate control, to simplify a wireless driver development, such a data structure is provided to notify the driver which rates will be utilized if retransmission is needed. After finishing transmission, the driver updates the data structure and notifies the mac802.11 framework which rates are actually attempted and the number of attempts, respectively. Based on this information, the mac802.11 framework can calculate statistics for the metric of a rate control mechanism. It is not difficult to set the rate-retry-chain on the NS3 platform, but it is a tough task to obtain the transmission status after transmission because there is no driver at the lower layer to update the data structure. Another example is Minstrel in NS3. The current approach for the Minstrel implementation regards the retransmission times of all rates in the multi-rate-retry chain as the retransmission count for the first rate in the chain. Therefore, the retransmission status for lower rates in the chain is neglected. This, without doubt, negatively affects the performance evaluation of the Minstrel mechanism.

Compared to simulations, the deployment of an over-the-air experiment platform (e.g. ORBIT [52]) is more expensive, particularly when building a large network. It is also difficult to maintain. The maintenance work includes trouble-shooting link breaks and security. In over-the-air experiments, the wireless signal travels over the air, so it captures the real environment dynamics caused by the movement of obstacles and various types of interference from competing stations, microwave ovens, and so on. However, it is hard to reproduce over-the-air results because the environmental conditions vary.

Controllable platforms [53, 54] (also called emulation) are designed to solve these problems. The wireless signal travels over co-axial cables instead of over-the-air, and a variable attenuator is utilised between a traffic source and sink to generate various link conditions. All stations are put into separate shielding boxes to isolate stations from external interference, so performance is compared fairly. However, the drawback of a controllable platform is that it can only construct a small network for evaluation so the contention environment may not be created using controllable platforms.

The performance comparison based on simulations, emulation platforms and over-the-air experiments is summarized in Table 5.

4.5.1 Simulation based evaluation

Vutukuru et al. [22] evaluate the performance of SoftRate, RRAA, SampleRate, an RBAR-like mechanism, and a CHARM-like mechanism using NS3 simulations. The RBAR-like mechanism enables the acknowledgements to carry the SNR feedback, and the RTS/CTS mechanism is disabled. The CHARM-like mechanism uses the average SNR over multiple frames. The topology in the experiment is an N-clients to 1-AP (Access Point) scenario, and the traffic is modelled as TCP flows with a frame size of 1400 bytes. The slow fading channel is simulated by the sender moving away from the receiver at a walking speed while the fast fading channel is simulated by the speed of a vehicle. The hidden terminal interference is simulated by the imperfect carrier sense. In the slow fading scenario, SoftRate shows higher throughput than SampleRate, RRAA, CHARM and RBAR. SampleRate has the lowest throughput in the scenario. In a fast fading channel, SoftRate shows the best throughput and RBAR has the lowest throughput. In the hidden terminal interference scenario, SoftRate also outperforms other mechanisms, and RRAA achieves the lowest throughput.

In [31], BEWARE, CARA, ARF and ARF-RTS (ARF with RTS turned on) are compared in the varying-contending-station and the Ricean-fading scenarios. In the varying contending station scenario, the number of contending stations is varied from 0 to 15. CARA shows the best throughput. In the Ricean fading scenario, the Ricean parameter K and doppler spread f_m are varied. When K increases, the line-of-sight component is stronger, and the SNR increases. When f_m increases, the channel condition changes faster. BEWARE achieves the best throughput.

In [55], ARF, CARA and ERA are compared in a static, mobile, and contention network. The contention network is simulated with different levels of hidden terminal collisions. ERA outperforms ARF and CARA in terms of throughput in all scenarios.

SIRA + SGRA is compared with SGRA in [41] by varying the walking speed from 0 to 1 m/s. SIRA is an intra-frame algorithm which selects multiple rates for a single aggregate frame to be transmitted. SIRA + SGRA combines the intra-frame rate adaptation, SIRA, and the inter-frame rate control, SGRA. The results reveal that

SIRA + SGRA improves SGRA's throughput significantly in mobile environments.

AARC, Minstrel, RRAA, and CARA are evaluated on the NS3 platform in a slow fading scenario [46]. The slow fading channel is simulated by the walking speed of 5 m/s. It shows AARC has the best throughput and Minstrel has the worst throughput.

In [9], MutFed, ARF and AARF are compared in a mobile scenario on the OPNET platform. The transmitting node moves away then towards the receiver. MutFed shows the lowest retransmissions, minimal delay, and highest throughput.

4.5.2 Emulation based evaluation

Three rate control mechanisms plus two of their extensions are implemented and evaluated by Camp et al [57] on the Wireless Open Access Research Platform (WARP). The evaluated mechanisms include ARF, RRAA, RRAA with the A-RTS extension, RBAR, and RBAR with the opportunistic Auto Rate (OAR) extension. OAR [36] is an extension which enables rate adaption mechanisms to send back-to-back frames. The authors use a Spirent Communication Channel Emulator (SR5500) to emulate various channel conditions. They vary the coherence time from 100 µs to 100 ms on a single channel with a high average Received Signal Strength (RSS) of -40 dBm to evaluate the throughput. The coherence time refers to the time interval over which the channel condition is sufficiently constant to decode the received symbols. For a long coherence time (bigger than 100 ms), all mechanisms converge to similar throughput except RBAR. In a short coherence time (less than 1 ms), the highest throughput is achieved by ARF and RBAR with OAR extension.

In [7, 56], CHARM, Onoe, SampleRate, and AMRR are evaluated on the Carnegie Mellon University (CMU) platform [61] in mobile and hidden terminal environments. The CMU platform forwards Radio Frequency (RF) signals to a Digital Signal Processing (DSP) engine that uses a number of FPGAs to model the effects of signal propagation, such as mobility, hidden terminal interference, etc. In the mobile environment, with the speed from 0.5 to 2 m/s, CHARM performs the best and Onoe has the lowest throughput. In the hidden terminal environment, CHARM and SampleRate outperform Onoe and AMRR.

In [58], AMRR, Onoe, SampleRate, and Minstrel are compared in the Conducted Testbed [54], which uses a programmable RF signal attenuator to generate varying channel conditions. The experiments are conducted in three scenarios. The first evaluates performance under fixed channel conditions with different SNR. The second compares the performance under a general interference environment. The impact of interference duration, interference interval, and interference strength are evaluated. The last scenario is to evaluate the effect of hidden terminal problems. In fixed channel conditions, SampleRate shows the best throughput. In a general interference environment, SampleRate, AMRR and Onoe cannot select the optimal transmission rate when there is a strong interference, so they achieve the lowest throughput. In the hidden terminal scenario, the impact of hidden terminal traffic and heterogeneous link quality are evaluated. Minstrel shows the best throughput since it does not behave too aggressively on the good link nor experiences starvation on the bad link. This is because the throughput metric in Minstrel considers both the bit rate and the FLR. This guarantees that Minstrel can select a high rate with a significant FLR as long as it can achieve the best throughput. Therefore, it does not decrease the rate under the hidden terminal interference. In [13], Minstrel-rts, Minstrel, PIDE, and PID are also evaluated on the Conducted Testbed [54] for the hidden terminal scenario. The hidden terminal traffic load is varied from 1 Mbps to 17 Mbps. Minstrel-rts displays the best throughput, followed by Minstrel. PID has the lowest throughput. This is because Minstrel-rts automatically enables the RTS/CTS to mitigate the impact of hidden terminals. Using the same platform, PIDE, PID, and Minstrel are evaluated in static, linearly increasing/decreasing, and suddenly changing channel conditions [15]. PIDE shows the best performance in all scenarios, as it improves the rate oscillation problem in PID. The same platform is utilized to make a performance comparison between RCELC and Minstrel [32], in static, linearly increasing/decreasing, and suddenly changing channel conditions. The results confirm that RCELC outperforms Minstrel in all scenarios because RCELC uses a more accurate metric to estimate the throughput achieved by each rate.

In [42], ESNR, SampleRate, and SoftRate are compared in a customized simulator that combines the Matlab tool and the GNU Radio. In Single Input Single Output (SISO) mobile environments, ESNR performs slightly better than SampleRate, and SampleRate performs slightly better than SoftRate in both human-speed and fast mobile experiments.

4.5.3 Over-the-air evaluation

ARF, AARF, Onoe, and SampleRate are evaluated in [30] on a 45-node indoor test-bed and a 38-node outdoor test-bed. Packets used are 1500 byte User Datagram Protocol (UDP) packets. ARF achieves the lowest throughput in both outdoor and indoor networks. The performance of fixed rates is also studied by disabling the rate control mechanism. Onoe performs to the best fixed rate in indoor experiments, and it performs poorly in low-quality links in the 802.11a and 802.11b outdoor networks. SampleRate achieves the best throughput on both the indoor and outdoor networks even on lossy links.

Wong et al. [26] evaluated RRAA, SampleRate, ARF, and AARF in the IEEE 802.11a/b networks with various settings such as static/mobile clients, with/without hidden stations. In a scenario with stationary clients, the results for four clients in the IEEE 802.11a network show that RRAA has the best performance and SampleRate has better performance than ARF and AARF in the IEEE 802.11b static networks. In a mobile-client scenario with UDP packets, SampleRate has lower throughput than RRAA, ARF and AARF. In the hidden terminal scenario, RRAA with the A-RTS mechanism has the best performance. The experiments are conducted in a real environment with limited nodes, and the result is subject to various client positions. It is not clear how the hidden terminal scenario is created in the real environment, and it is obvious that it will be difficult to achieve a repeatable evaluation environment.

SampleRate, AMRR, and Onoe are evaluated in [60] in both indoor and outdoor environments. The indoor environment has 12 nodes on one floor while the outdoor environment has five wireless routers deployed on the rooftops of three buildings. Experiments were carried out in the 802.11g mode using UDP traffic. In good quality channels in the indoor environment where the highest throughput is achieved by 54Mbps, AMRR and SampleRate show better performance than Onoe. A high contention channel condition is formed in the indoor environment by enabling eleven senders to transmit frames concurrently. SampleRate sends only 7% of frames at the optimal rate. AMRR and Onoe perform even worse than SampleRate. In the low medium contention channel condition, SampleRate also shows better performance than AMRR and Onoe. However, it is not clear how to determine the degree of contention in the environment. In the outdoor environment, SampleRate also shows better performance than AMRR and Onoe.

In [62], the Minstrel rate control mechanism is compared with fix rates in the 802.11g networks. The experiment results show that Minstrel performs well in static channel conditions but fails to select the optimal rates in some cases of dynamic channel conditions. Specifically, when the channel degrades from good to bad, Minstrel does not perform as well as in the channel condition when the channel improves from bad to good.

The MiRA, RRAA, SampleRate, and Atheros MIMO rate adaptation [59] are compared in static, mobile, and hidden terminal environments. The Atheros MIMO rate adaptation is based on Minstrel, but it modifies specific features to cater for the MIMO environments. The results show that MiRA has better performance than other rate control mechanisms in all scenarios.

SampleRate and FARA are evaluated in [39] for 17 different locations using 20 MHz and 100 MHz channels. FARA performs better than SampleRate for all locations.

HA-RRAA, SampleRate, RRAA, and ARF are compared in different scenarios in [28]. These scenarios include UDP and TCP flows, static and mobile settings, hidden terminals, and 2.4 GHz or 5 GHz bands. HA-RRAA has the best throughput in all scenarios except the mobile UDP environment using the 802.11a channel.

In [8], Onoe, ARF, CHARM, RRAA, SampleRate, AMRR, and RAM are evaluated in static, mobile at walking or vehicle speed, in-door mobile, and indoor with interference environments. RAM displays the best performance in all scenarios. Onoe almost achieves the worst for all experiments except the indoor mobile scenario.

SGRA, RRAA, and ARF are compared in [37] in the static 802.11 UDP environment, hidden terminal environment, and the environment with bluetooth interference. SGRA outperforms other mechanisms in all scenarios.

5 Lessons learned

In this section, we conclude our study of rate control mechanisms at the MAC-layer and share the following important lessons with researchers developing new rate control mechanisms.

5.1 Decreasing transmission rate upon severe frame loss is not a good reaction if collisions exist

It is generally recognized by most existing mechanisms [4, 24, 25, 29] that the transmission rate must be decreased if intense frame losses occur. The motivation for this reaction is that the underlying problem behind significant frame losses is the deteriorating link condition between the sender and receiver and that the current rate must be decreased to cater for the worsening link.

The above rule holds for most cases where the link condition between the communicating pair degrades but breaks if the collision occurs. Both hidden terminal and congestion collisions can cause significant frame losses that are independent of the channel quality. To decrease the transmission rate under such conditions will not solve the problem but make it worse, as lower rates have longer transmission times and the increased transmission time for each frame aggravates collision.

Therefore, the guideline of decreasing the rate upon severe frame losses does not hold in collision loss cases. The rate adaptation mechanism needs to differentiate various losses and react accordingly.

5.2 Increasing transmission rate only because of low frame loss may cause the rate oscillation problem

Most of the existing mechanisms [4, 26] suggest that the transmission rate must be increased if the frame loss ratio is low. The underlying conjecture is that the link condition supports the current rate so well that it should support the next higher rate.

This conjecture holds true for most cases but fails in some cases where the current rate is already the best, yet with a low loss ratio. From Fig. 1, the horizontal curve of a rate stands for its maximum achievable throughput with the frame loss ratio equal to zero. Its next higher rate's curve intersects with the rate's curve. The horizontal curve is divided into two parts by the intersection point. For example, the intersection point of 48 Mbit/s and 36 Mbit/s is path loss 77 dB. On any point of the left horizontal curve (before 77 dB), the rate must be increased because the throughput of the next higher rate is higher. On the right side of the horizontal curve (after 77 dB but before 81 dB), although the rate has a zero frame loss ratio, the rate cannot be increased because the throughput of the next higher rate is lower. For right side points, the loss ratio of the next higher rate is high. If the rate is increased, the high loss ratio will force the rate to decrease again. This causes the rate oscillation problem in rate selections, as discussed in PID [4].

Therefore, the proposed higher rate should be tested for better throughput before switching to it. This is one of the improvements we incorporated into PIDE in [15].

5.3 Using only success or failure information of probing frames may mislead rate adaptation decisions

The probing approach is utilized in several rate control mechanisms [5, 25, 30] for rate adaptation. Probing frames are those frames that are transmitted at rates other than the current rate. Without probing, a rate control mechanism's knowledge is limited to the current rate's performance, and it has to make a rate decrease/increase decision just based on the limited knowledge. As discussed in the last subsection, the horizontal region of a rate's throughput curve is divided into two parts by the intersection point with its next higher rate in Fig. 1. Throughout the horizontal region, the current rate performs very well, but the next higher rate's performance is very different among the two parts: good on the left but very bad on the right. This means that based on the current rate performance, we cannot predict higher rates' performance. Therefore, the probing is necessary.

However, using a binary probing result, i.e., the probing frame is successful or not (e.g. AARF [25]) is not an adequate method. For some point in Fig. 1, it can have near-perfect transmission at 12 Mbps, but almost 40% frame loss at 18 Mbps. This means that a probing frame at 18 Mbps has a probability of 60% to get through, and the rate will be likely increased. However, 18 Mbps with 40% frame loss achieves lower throughput than 12 Mbps with zero frame loss. Therefore, binary probing results can mislead the rate control mechanism to a wrong decision.

The throughput is a direct performance metric to reflect the most common goal in terms of network optimization, i.e., maximizing the aggregate throughput. The throughput metric is probably better than the binary transmission status of the probing frames. An appropriate way for probing could be estimating the throughput metric according to Eq. (6) [15] for the probing rate. In this way, only when the probing rate has a better throughput, the rate is switched to the probing rate.

5.4 Using the physical layer metric for rate adaptation decision may cause a standard compliance problem or is based on unrealistic assumptions

A number of proposals use physical layer metrics, including SNR and BER, to estimate the link quality and make rate adaptation decisions. However, BER and SNR cannot be measured locally at the senders, without making the assumption of a symmetric link. According to where the transmission rate is selected, SNR based mechanisms are classified into sender controlled, e.g., CHARM [7], SGRA [37] and LA [38], and receiver controlled mechanisms, e.g., MutFed [9] and RAM [8]. For sender controlled mechanisms, the sender estimates the SNR at the receiver side. These mechanisms assume link symmetry between the sender and receiver, i.e., they assume the path loss of sender-to-receiver and receiver-to-sender is the same and both the sender and receiver experience the same level of interference and noise. This assumption does not hold in wireless environments. For receiver controlled mechanisms, SNR is measured at the receiver, and the rate is selected by the receiver. Then the rate is notified to the sender by transmitting an ACK using the selected rate, which causes the standard compliance problem. Moreover, SNR is measured during the reception of the preamble. The latter part of the frame may experience fading and interference, which is not captured by the SNR at the preamble. Besides, SNR may not be accurate because of hardware calibration and interfering transmissions [8].

5.5 Using simple heuristics to address the hidden terminal problem cannot maximize throughput gain

Some rate control mechanisms use simple heuristics to address the hidden terminal problem. In CARA [16], RTS is turned on for the first retransmission attempt since the first transmission failure may be caused by collisions. If the first retransmission also fails, the loss is attributed to channel errors as RTS should have already mitigated the collisions. Therefore, a lower rate will be applied. RRAA [26] uses A-RTS to solve the hidden terminal problem. A window for monitoring the performance of RTS/CTS is updated in RRAA. Its initial value is zero, which means disabling RTS. If a frame is lost without RTS, the window is increased by one, as the frame may have collided. If a frame is lost with RTS or succeeds without RTS, the window is halved as RTS does not contribute in both cases. The window is decreased by one if the sender transmits a frame with RTS enabled.

In CARA, if the hidden terminal keeps sending frames, the first attempt of a frame will always fail, because RTS is switched off for the first transmission. Even if the next retransmission succeeds, the sender should have 50% frame losses. In RRAA, the window is increased by one if a frame is lost without RTS. This means that the RTS is switched on for the next attempt only if the last attempt fails. In this case, it also has 50% frame loss if the hidden terminal keeps sending frames. The hidden terminal collision condition should be reevaluated when the window reaches zero. If the hidden terminal is still causing collisions, then the window should be set to a greater value, other than being increased from zero to one. Another way to solve this problem could be to evaluate the throughput achieved with RTS and without RTS, and then turn on RTS if doing so can bring throughput gains [13].

Newer networks such as IEEE 802.11ac offer a feature, called *channel bonding*, to bind multiple 20 MHz channels together for higher bandwidth. In some cases, not all 20 MHz channels are free for transmission. IEEE 802.11ac extended the RTS and CTS to add *bandwidth signaling* [63]. This new feature makes use of RTS/CTS to detect/avoid hidden terminals and to identify which 20 MHz channels are free for channel bonding. To request a channel, the initiator sends separate RTS frames on the four primary 20 MHz channels. The receiver only responds with CTS on the 20 MHz channels that are free from interference. However, this approach still incurs the RTS/CTS overhead.

5.6 Considering surrounding traffic load causes rate adaptation not to converge to the best rate

Some mechanisms [31, 64] suggest that the background traffic and the traffic load must be considered. The underlying conjecture is that it prolongs the actual transmission time for delivering a frame.

BEWARE [31] makes rate adaptation based on the transmission time. When calculating the performance metric, it adds into calculation the backoff time caused by a neighbouring node's transmission. Therefore, varying traffic load in the background could cause rate changes. For example, suddenly increased traffic load increases the transmission time of the current rate, which may result in rate decreases. The dependency of rate adaptation decision on background traffic makes rate control mechanism load aware and difficult to converge to the best rate if the traffic load is continuously changing.

Should a rate control mechanism make use of network load in rate adaptation, it should be aware of the overhead and negative impacts involved.

5.7 Assuming a unique FLR threshold for all rates cannot maximize the throughput

Some rate control mechanisms assume a unique FLR threshold (PID-14%, AMRR-33% and Onoe-50%) to decrease the transmission rate. This is based on the assumption that the current rate with FLR that is above the threshold achieves lower throughput than its next lower rate.

Fig. 1 shows the throughput achieved by all fixed rates in the IEEE 802.11a mode. From Fig. 1, we can see that the absolute maximum throughput for a given path loss is on the envelope that connects the maximum throughput points of all the fixed rates. Hence the optimal maximum FLR of each fixed rate is the crossing point when a higher rate switches to a lower rate to maintain the highest throughput. When we compared the FLR of these throughput points in Fig. 1, we found that they are all different values. Therefore, using a fixed target FLR for all rates fails to achieve the maximum throughput.

When calculating the FLR threshold to decrease the rate, FLR based mechanisms should accurately compute the FLR of the crossing points of adjacent rates' throughput in order to ensure maximum throughput. In [15], Yin et al. conducted detailed experiments to verify that there is no single best FLR threshold which can help to achieve the maximum throughput.

6 Future research

In the last two decades, many rate control mechanisms have been proposed for 802.11 networks. However, rate control mechanisms are also required in other wireless networks. Other wireless networks, e.g. body area networks [65] and vehicular networks [66], have some features in common with 802.11 networks, such as wireless shared medium, interference, and fading characteristics. However, they also have some specific features. Therefore, rate control mechanisms designed for 802.11 networks may not be suitable for other networks. For example, rate control mechanisms for 802.11 networks operate in a unicast mode. Feedback, e.g. ACKs, are utilized to provide channel quality information to the sender. Vehicular networks use a broadcast mode, in which a sender will not receive ACKs but others' transmissions instead. The challenge for vehicle networks is how to estimate the channel condition without ACKs. In body area networks, a number of sensors are worn/implanted to exchange information and monitor data. The channel characteristics are highly dynamic as a human body moves with gestures, posture change, or human mobility [67]. This is different from traditional Wi-Fi networks, where nodes are relatively stationary. The suitability of rate control mechanisms from 802.11 networks for other networks should be investigated before being applied to these networks.

6.1 New opportunities

6.1.1 Limitations in existing rate control

The sender in ACK-based rate control mechanisms calculates the link quality metric such as CTR (e.g. ARF and AARF), FLR (e.g. PID and RRAA), transmission time (e.g. SampleRate), and throughput (e.g. Minstrel and RCELC) based on acknowledgements. They make rate adaptations according to the monitored metrics. If an attacker forged ACKs to acknowledge those lost frames, as shown in Fig. 7a, this can interfere with the link quality metric calculation, misleading the rate control mechanism to believe the channel condition is very good and select a higher rate that is not supported. In some SNR-based mechanisms, e.g. CHARM, the sender calculates the SNR based on the transmission power advertised in beacons or probes. Therefore, an attacker can fabricate beacons or probes with a fake transmission power, as shown in Fig. 7b, which affects rate selection, misleading the sender to choose an inappropriate rate. Therefore, rate control should be combined with a mechanism that can verify the ACK/beacon/probe authenticity.

In BER-based mechanisms, the BER is utilized for rate adaptation. An attacker can jam the channel, as shown in Fig. 7(c), to increase the BER to mislead the sender to choose a lower rate. Therefore, BER-based rate control should be able to differentiate jamming errors and channel errors.

When directional antennas are utilized in 802.11 networks, deafness [68] may pose a problem for rate control mechanisms as shown in Fig. 8. Node A may try to send a frame to node B which is transmitting a frame to node C. Because A cannot hear B's directional transmission and B cannot respond with an ACK to A, A concludes that the frame is lost. This can have a big impact on A's rate metric calculation, which leads A to select a low rate. RTS/CTS can solve this problem. When hearing B's RTS frame, A knows B is transmitting. However, when RTS/CTS is disabled, the rate control mechanism should figure out the deafness situation without decreasing the rate.

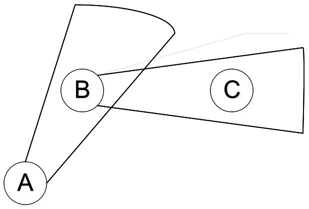
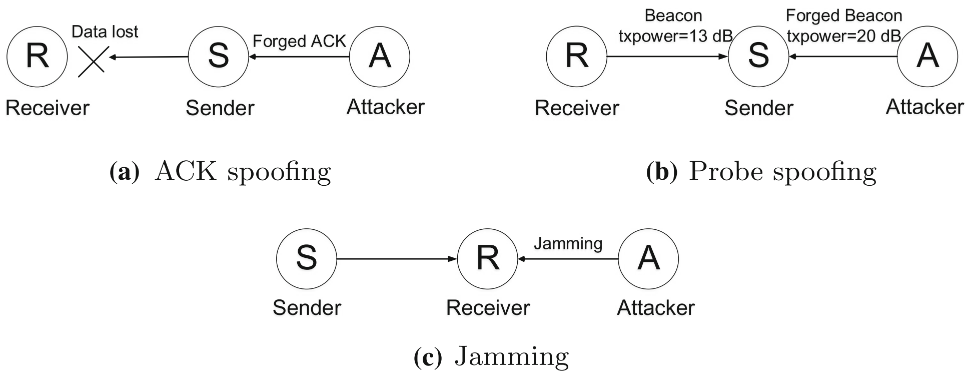


Fig. 8 Deafness problem for 802.11 networks

Fig. 7 Attacks against rate control mechanisms



(a) ACK spoofing

(b) Probe spoofing

(c) Jamming

6.1.2 Open issues in new standards

The new feature of 802.11n compared to 802.11abg is the Single User Multiple-Input Multiple-Output technology (SU-MIMO). Two operational modes are supported by the 802.11n standard. One is the diversity-oriented single-stream (SS) mode. The other is the spatial multiplexing-driven, double-stream (DS) mode. With the two modes, 802.11n allows many more rate options than the 802.11abg mode, ranging from 6.5 to 600 Mbps. As the rate increases, the loss does not monotonically grow with rates in different modes [59]. Therefore, the existing rate control mechanisms designed for 802.11abg cannot be applied to the 802.11n networks. Although a few rate adaptation algorithms, e.g. MiRA [34, 59], are designed for the 802.11n networks, the characteristics of the 802.11n technology have not been fully explored. For example, frame aggregation is an important MAC enhancement in the 802.11n networks. One type of frame aggregation methods is A-MPDU, which requires a long frame duration, e.g., the maximum of 10ms. This exceeds the coherence time in mobile environments [69]. Therefore, a rate control mechanism should be designed to adapt rates within an individual frame [41]. Even for the 802.11abg mode, recent work, e.g. [40], shows the necessity to make rate adaptation during a frame transmission in fast fading environments. Other rate control mechanisms designed for the 802.11n networks, including InFRA, are based on the ESNR. To calculate the metric, the receiver needs to compute BER and then map BER to ESNR. BER is not directly a reflection of the throughput, so it cannot guarantee to achieve maximum throughput.

In 802.11ac networks [70], multiple single-antenna clients communicate concurrently with a multi-antenna AP to form Multi-User MIMO transmissions [44]. The rate decisions made by clients are not independent, and they interact with each other, which is different from the existing 802.11 networks. This poses a new challenge for a rate control design. 802.11ac uses a compressed form of CSI (Channel State Information) to reduce the overhead of sending full CSI feedbacks. In this case, 802.11ac APs can only depend on MAC-layer feedbacks, e.g. Packet Error Rate (PER) and rate statistics to decide MU-MIMO groups [71]. Only users that have similar profiles (e.g. same channel bandwidth and throughput) should be classified into the same group. This is to overcome the efficiency problem caused by heterogeneous bandwidth users. However, the profile parameters depend on the rate control metric. Therefore, the rate control mechanism should be combined with group user decision to enhance network throughput in 802.11ac networks.

The IEEE 802.11ad technology [72] doubles the bandwidth provided by the 802.11ac technology but sacrifices it for a very short transmission range of a few meters [73]. This means the transmission rates are sensitive to the distance between the two communicating nodes. Mobility causes the distance to change very frequently, so rate control mechanisms should be very responsive to mobility. In this case, methods such as using a shorter rate adaption period and putting more weight on recent metric measurements and employing per-packet rate selection can be adopted to improve performance.

In the 802.11e standard [74], multicast frames are transmitted at one of the ’Basic Service Set’ rates. However, basic rates are generally lower than normal rates. Also, group members have different distances to the AP, and each member may experience different interference. Therefore, there is an anomaly of link quality among group members. Sending frames at one of the basic rates may not be able to achieve the optimal throughput. In addition, retransmission of lost multicast frames is disabled, so the robustness is not ensured. The 802.11aa standard [75–77] aims to enhance the efficiency and robustness of multicast video streaming in Wireless Local Area Networks (WLANs). The Block ACK mechanism, as shown in Fig. 9, is utilized to ensure reliability. The rate control mechanism is recommended to adapt all available transmission rates. However, the specification is not standardized [78]. The block ACK contains frame loss information so that it can be leveraged to calculate the link quality metrics, e.g. frame delivery ratio and throughput, to select an appropriate rate. One difference from rate control in 802.11 unicast networks is that the sender should select an appropriate rate for all multicast links instead of just one link. Therefore, a metric that considers all counted links should be designed. In [78], Mansour et al. present a mechanism based on the FLR metric. However, they fail to address the rate oscillation problem inherited from FLR-based mechanisms. In [79], Salvador et al. present the first 802.11aa implementation on commodity cards and make it publicly available; this can be utilized by researchers to develop and evaluate rate control mechanisms for 802.11aa in real environments.

The IEEE 802.11ah WLAN protocol [80] allows a longer transmission range between an AP and clients, up to multiple kilometers [81]. Therefore, it is widely used in many fields, including health care, Internet-of-Things, non-intrusive remote sensing, and UAVs. Most applications are deployed in outdoor environments with higher node density, mobility, and interference [82]. Therefore, rate control mechanisms should combine with WLAN configurations at the physical layer and the MAC layer [83] to tackle challenges from contention loss, mobility loss, and interference loss that are more severe than in an indoor environment. At a specific time, only a RAW group can transmit frames, and other RAW groups sleep [82]. As the number of RAW

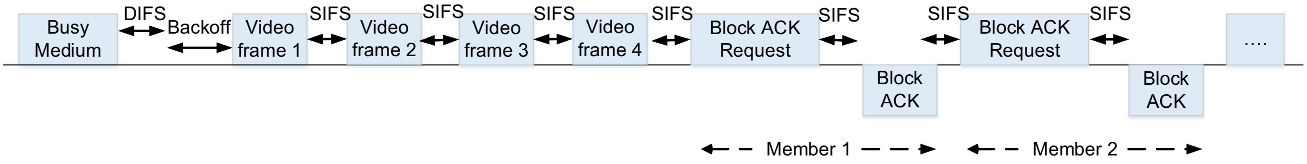


Fig. 9 Block ACK mechanism in 802.11aa

groups increases, the channel quality may change dramatically due to mobility and interference when waking up after a very long sleeping time. Therefore, rate control mechanisms for 802.11ah networks should be responsive to sudden channel quality changes.

The 802.11ax technology [84] is to replace the 802.11n and 802.11ac technology in the long run. In 802.11ax networks, the transmission power and CCA levels can be adaptively changed [85]. Increasing the CCA levels can reduce the influence area and increase the chance to transmit, hence improving the throughput. Decreasing the transmission power reduces the influence area, which improves the spatial reuse. However, the number of packet errors may increase, and lower transmission rate may be selected. Therefore, rate control mechanisms should consider the impact of transmission power and CCA level changes. As one of the representative techniques in 802.11ax, Multi-User MIMO, promises significant throughput gains by supporting multiple concurrent data streams to a group of users. However, this design introduces challenges in selecting the best-throughput for MU-MIMO groups due to, (a) lack of full CSI feedback from the user, which is commonly used for MU-MIMO grouping, (b) hardware capability and external interference, which results in heterogeneous bandwidth users, i.e. not all group members supporting the same channel bandwidth, (c) limited onboard resources on APs, which limits the portability of the complex mathematical and memory-intensive operations to such platforms [86]. With these challenges in mind, new approaches to rate adaption need to be studied to fully utilize the benefits of IEEE 802.11ax.

6.1.3 Opportunities in experimental platforms

The mac80211 framework is a wireless infrastructure designed for the new Linux Kernel. The framework incorporates common features, e.g. rate control, to simplify a wireless driver design and implementation [13]. This framework can support a large amount of research work related to rate control. First, new mechanisms can be designed, implemented, and evaluated. Underlying wireless drivers report to the framework the attempted rates, respective retry counts, and the signal strength for the received ACKs after delivering a data frame. Therefore, mechanisms based on FLR, SNR, throughput, transmission time, etc. can be implemented on this framework with ease. Second, most of the existing rate control mechanisms make rate adaptations based on the rate adaptation period (RAP), for example, 10 s for SampleRate, 100 ms for Minstrel and 1 s for Onoe. However, the effect of RAP has not been explored. A large RAP indicates less computation overhead but also less adaptivity to a fast-changing channel. Using the framework, a comprehensive study could be carried out to discover the underlying impact of various RAP settings. Third, a cross-layer design is supported by the framework. The 802.11s mesh protocol is built on the framework to enable ad hoc communications among clients, so the interactions between rate control mechanisms and the mesh routing protocol can be supported. Some link metrics that are calculated by the rate control can be shared with the mesh routing protocol. Performance enhancement may be achieved through a cross-layer optimisation. This can be implemented and evaluated on the mac80211 framework. Fourth, the multi-rate feature can cause a performance anomaly problem [87] where performance degradation will be suffered if one client has a very bad quality link to the AP. The mac80211 framework can be utilized as a platform to study and improve the fairness of air channel time in a multi-rate wireless environment [88].

The WARP v3 kit is a Field-Programmable Gate Array (FPGA) hardware platform built with the 802.11 reference design. It is no longer available for sale. The new product is the new 802.11 MAC/PHY design, built entirely with the Xilinx Vivado tools and running on third-party hardware. This new design provides all the same features as the original 802.11 reference design for the WARP v3 kit. However, it is built with modern tools and runs on modern hardware. The WARP v3 kit or the new 802.11 MAC/PHY design have the time-critical feature that the mac80211 framework does not have. Using the design, the researcher can have fine control of the Distributed Coordination Function (DCF) and the management of control frames, e.g. ACK and RTS. The mac80211 framework abstracts the common features of various wireless drivers, so it works at one layer above the driver. The inter-frame space for control frames is a SIFS time, and the driver is unable to send frames within such a short time. Thus, control frames are usually generated and sent by the hardware directly on the Linux system. Therefore, the mac80211 framework is neither able to send forged control frames in time nor customize the control frame. Using the WARP v3 kit or the new 802.11 MAC/PHY design, it is possible to implement an ACK spoofing attack [89] by sending a forged ACK to the sender before the ACK timer expires. In an SNR-based rate control mechanisms, the receiver can also piggyback the selected rate in a CTS frame to notify the sender [6]. The ACK frame can also be modified to carry the selected rate. Moreover, a new per-frame rate control mechanism can be designed, implemented, and evaluated using the WARP v3 kit or the new 802.11 MAC/PHY design, which is impossible on the mac80211 framework.

Using the NS3 platform, we can simulate a large network and study the performance of rate control mechanisms on a large scale. For example, the 802.11ah network supports up to 8192 stations connected to one AP. The 802.11ah rates have a much longer transmission range compared to 802.11n/ac networks. The crowded 802.11ah network may generate significant interference due to inappropriate RAW grouping methods. For example, having too many stations in a RAW group may introduce significant contention collisions, which would have an impact on rate control mechanisms. To configure a vast 802.11ah network using either the mac80211 framework or the WARP platform is costly. NS3 is the most cost-effective platform to study rate control performance for a large network with different features [20]. Moreover, there is no hardware on the NS3 platform. Thus, it is possible to implement time-critical features for rate control mechanisms, e.g. designing per-frame rate control mechanisms and ACK spoofing attacks.

6.2 Challenges from new applications

6.2.1 Increasing contention

The Internet-of-Things (IoT) [90] has gained large popularity nowadays due to its vision of anytime, anywhere and anymedia, which motivates the development of communication technologies. Sensor networks play an important role in IoT, and they help track the target's location, temperature, movement, etc., bridging the gap between the real and digital worlds. In a dense sensor network, nodes communicate in a wireless multi-hop fashion. Rate control mechanisms in sensor networks need to address the scalability and contention problem since the number of nodes in a sensor network will be very high. Furthermore, the largest frame size is usually 102 octets, which is very small. This means that the ratio of communication overhead (e.g. SIFS and DIFS) to payload transmission time is much higher. Frame aggregation may be used in MAC rate control mechanisms to enhance performance. Besides, sensor nodes spend a large amount of time in a sleeping mode, so the selection of transmission rates after waking up is a research question. Cars, trains, and buses are equipped with sensors to provide better navigation and avoid car collisions. The research question for rate control mechanisms is how to minimize transmission time, and enhance timeliness, and provide high-mobility support in dense networks, especially for the highway system.

Cognitive radio technology is capable of learning the status of a frequency band in the environment. Therefore, it is able to switch to a free frequency band when it senses that the currently used frequency is too crowded. It is observed in [39] that the link quality varies according to different frequency bands, exhibited by SNR values. Hence, although the free frequency band experiences less interference than the busy one, the link quality is not necessarily higher, and a higher rate is not always supported. A blind switching to a free channel band may cause a rate decrease and hence throughput degradation. Rate control mechanisms in cognitive radio networks should take into account the interference and link quality of a frequency band and then make appropriate frequency and rate selection. On one hand, interference, due to exclusive frequency sharing, determines what percentage of time slots is available. On the other hand, the link quality determines the maximum throughput. In addition, with a Quality of Service (QoS) guarantee, combining power and rate control to optimize the resource consumption is another research direction [91] in cognitive radio networks.

6.2.2 High mobility

Unmanned Aerial Vehicles (UAVs) [92] play a key role in modern military battlefields, conducting surveillance and reconnaissance. Communication is enabled between UAVs and the ground control station for command and control, as well as transmitting captured information, including image and video files. The Long-Term Evolution (LTE) technology is usually utilized in UAVs for communication, providing high speed and low latency compared to other technologies, including Joint Tactical Radio System (JTRS). Besides the high volume of transmitted data and its timeliness, UAVs are highly mobile and energy limited. MAC rate control mechanisms for the LTE technology used in UAVs have gained little attention so far. A good MAC layer rate control mechanism for this kind of network needs to improve the throughput, reduce the latency, deal with mobility, and power efficiency problems. Besides, when applied in military applications, UAVs become the targets for jamming attacks. Orakcal et al. [93] demonstrate that several rate control mechanisms, e.g. SampleRate and ARF, are vulnerable to the jamming attack. Jamming attacks increase the FLR, so some mechanisms may decrease the rate. Therefore, in UAVs, a robust mechanism should differentiate frame losses from jamming, mobility, and channel errors.

Mobile Cloud Computing (MCC) [94] has emerged as a new technology to move computing and storage from mobile devices to the cloud with powerful and centralized computing platforms. Therefore, services provided by the cloud are accessed via the wireless communication technology. The mobile devices then are confronted with many challenges such as limited battery life and bandwidth, and channel dynamics caused by mobility and interference [95]. Therefore, the design goals of rate control mechanisms may vary, and they should address the robustness problem in the face of mobility and interference. Besides, various technologies, e.g. Wideband Code Division Multiple Access (WCDMA) and WLANs, are used for communication in MCC. These technologies are wireless but have different features. Therefore, rate control mechanisms for those technologies need to be investigated separately in order to enhance the aggregate performance.

6.2.3 Increasing demands on energy efficiency

Traditionally, rate adaptation mechanisms have been designed to maximize network goodput. However, battery-powered devices, including smartphones, UAVs; and other smart Internet-of-Things (IoT) devices (lightbulbs, door-locks), start to dominate in wireless networks, so there are increasing needs to design rate adaptation algorithms that are energy-efficient. The measurements presented in [96] show that an 802.11n 3×3 MIMO receiver consumes about twice the power of 802.11a during an active transmission, and 1.5 times power when idle. The authors in [96, 97] claim that most existing solutions are practical to ensure high goodput but not energy savings. Their results show that CARA [16] and MiRA [34] incur per-bit energy waste at a network interface card as large as 54.5% and 52.9%, respectively. The factor of power consumption brings the designing of rate adaptation algorithms into a new era which can benefit power-constrained devices. For example, for the sender to transmit a data frame, a lower transmission rate consumes more energy than a higher rate due to longer transmission time. On the other hand, this is not the case for Software Defined Radio (SDR), in which power consumption involves the expenditure on frame transmission and the processing of the microprocessors running the SDR software. Low transmission rate reduces the speed and power consumption of the microprocessor. Recent research shows that the effect of running the SDR software on microprocessors is more significant than transmission time if power consumption is concerned [98]. Therefore, if the design goal for rate control is power efficiency, rate control mechanisms could consider using lower transmission rates when the channel utilization is low.

Also, the assumption that the optimality in throughput means optimality in power consumption in the 802.11 networks has been put in question recently [96, 99]. Ucar et al. [100] confirm that throughput and power efficiency cannot be simultaneously optimized. However, the discovery is based on theoretical analysis. A practical experimental study is still lacking. If energy efficiency is a concern in the 802.11 networks, the performance metric should be redesigned for existing throughput-oriented rate control mechanisms to maximize power efficiency. Power consumption metrics may be subject to many factors, including CSMA backoff, frame losses, and transmission rate, which are hard to be measured accurately for individual frame transmissions. An active learning approach based on machine learning [101] can be utilized to obtain feedback from the environments to achieve optimal power and rate control.

7 Conclusion

In this paper, we provided a comprehensive survey of rate control mechanisms at the MAC-layer of 802.11 networks that were developed over the past two decades. While many solutions have been proposed to address different goals, only a handful of them have been adopted in practice. Among them, some had not been studied and evaluated systematically until recently. This survey analyzes the existing solutions based on the two fundamental aspects in rate control—metrics and algorithms, and shares insights that have an impact on the development of new MAC-layer rate control mechanisms. Drawing observations from other published work, we further provided a comparison of the existing solutions and outlined their characteristics. We showed that the existing solutions for 802.11 networks cannot be directly adopted in other types of systems and networks. The findings from this survey offer guidelines for the development of new rate control mechanisms for emerging technologies and new applications.

Acknowledgements The work is partially supported by the NSFC Project 61702542. We would like to thank reviewers for their comments and suggestions that led to improvements of this paper.

Appendix

See Table 6.

Table 6 Table of acronyms used in this paper

Acronyms	Full name	Acronyms	Full name
FLR	Frame Loss Ratio	SNR	Signal Noise Ratio
ACK	Acknowledgment	FSPL	Free-Space Path Loss
SINR	Signal-and-Interference-to-Noise Ratio	RTS/CTS	Request to Send/Clear to Send
DCF	Distributed Coordination Function	CCA	Clear Channel Assessment
CSI	Channel State Information	RSSI	Received Signal Strength Indicator
RAW	Restricted Access Window	CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
AP	Access Point	SDR	Software Defined Radio
RAP	Rate Adaptation Period	BER	Bit Error Rate
CW	Congestion Window	PLCP	Physical Layer Convergence Protocol
BPSK	Binary Phase Shift Keying	OFDM	Orthogonal Frequency Division Multiplexing
DIFS	DCF Interframe Spacing	SIFS	Shortest InterFrame Spacing
SISO	Single Input Single Output	MIMO	Multiple Input and Multiple Output
MRR	Multi-Rate-Retry	CTR	Consecutive Transmission Result
ARF	Automatic Rate Fallback	AARF	Adaptive ARF
CARA	Collision-Aware Rate Adaptation	ERA	Effective Rate Adaptation
RRAA	Robust Rate Adaptation Algorithm	MTL	Maximum Tolerable Loss threshold
ORI	Opportunistic Rate Increase threshold	A-RTS	Adaptive RTS
HA-RRAA	History-Aware RRAA	AMRR	Adaptive Multi Rate Retry
PID	Proportional Integral Derivative	PIDE	PID Enhancement
ATT	Average Transmission Time	BEWARE	Background traffic-aWAre RatE adaptation
EWMA	Exponential Weighted Moving Average	BTR	Best Throughput Rate
RR	Random Rate	BPR	Best Probability Rate
BR	Base Rate	NBTR	Next Best Throughput Rate
CAW	Collision Avoidance Window	RCELC	Rate Control based on the Estimated Link Capcity
ELC	Estimated Link Capacity	MSDU	MAC Service Data Unit
SS	Single Stream	DS	Double Stream
SFER	Sub-Frame Error Rate	STRALE	STandard-compliant and mobility aware PHY RAte and A-MPDU LEngth adaptation
A-MPDU	Aggregated MPDU	RBAR	Receiver-based AutoRate
RSH	Reservation SubHeader	OAR	Opportunistic Auto Rate
MutFed	Mutual Feedback	RAM	Rate Adaptation in Mobile environments
CHARM	Channel Aware Rate Adaptation Algorithm	QoS	Quality of Service
SGRA	SNR Guided Rate Adaptation	LA	Link Adaptation
FARA	Frequency-Aware Rate Adaptation	SIRA	SNR-aware Intra-frame Rate Adaptation
ESNR	Effective SNR	QPSK	Quadrature Phase Shift Keying
QAM	Quadrature Amplitude Modulation	InFRA	In-Frame Rate Adaptation
GOS	Groups Of Symbols	CDF	Cumulative Distribution Function
CRC	Cyclic Redundancy Check	NACK	Negative ACK
EOS	End Of Stream	ABEP	A posteriori Bit Error Probability
AARC	ABEP-enabled Adaptive Rate Control	FER	Frame Error Rate
HARC	Hybrid Automatic Rate Control	RSS	Received Signal Strength
CMU	Carnegie Mellon University	RF	Radio Frequency
UDP	User Datagram Protocol	SU-MIMO	Single User Multiple-Input Multiple-Output technology
MU-MIMO	Multi User MIMO	PER	Packet Error Rate
WLANs	Wireless Local Area Networks	WARP	Wireless Open Access Research Platform

Table 6 (continued)

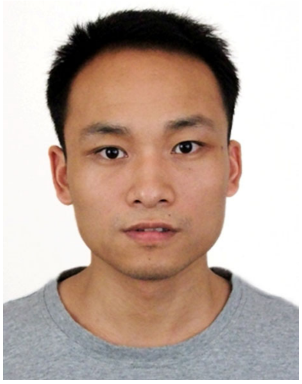
Acronyms	Full name	Acronyms	Full name
MCC	Mobile Cloud Computing	UAV	Unmanned Aerial Vehicle
LTE	Long-Term Evolution	IoT	Internet-of-Things

References

1. Cisco, Inc. (2016). *The internet of everything*. Retrieved 2016 from http://www.cisco.com/web/about/ac79/innov/IoE.html
2. Balasubramanian, N., Balasubramanian, A., & Venkataramani, A. (2009). Energy consumption in mobile phones: A measurement study and implications for network applications. In *Proceedings of SIGCOMM IMC'09, IMC '09*. (pp. 280–293). ACM.
3. Ngo, T. (2016, June). Why wi-fi stinks and how to fix it. Retrieved 2016 from http://spectrum.ieee.org/telecom/wireless/why-wifi-stinksand-how-to-fix-it?
4. PID. Retrieved 2016 from http://wireless.kernel.org/en/developers/Documentation/mac80211/RateControl/PID
5. Minstrel specification. Retrieved 2016 from http://madwifi-project.org/browser/madwifi/trunk/ath_rate/minstrel
6. Holland, G., Vaidya, N., & Bahl, P. (2001). A rate-adaptive MAC protocol for multi-hop wireless networks. In *Proceedings of MOBICOM*. (pp. 236–251). ACM.
7. Judd, G., Wang, X., & Steenkiste, P. (2007). Low-overhead channel-aware rate adaptation. In *Proceedings of MOBICOM*. (pp. 354–357), Montreal, QC: ACM.
8. Chen, X., Gangwal, P., & Qiao, D. (2012). RAM: Rate adaptation in mobile environments. *IEEE Transactions on Mobile Computing*, *11*(3), 464–477.
9. Khan, S., Mahmud, S.A., Noureddine, H., & Al-Raweshidy, H.S. (2010). Rate-adaptation for multi-rate IEEE 802.11 WLANs using mutual feedback between transmitter and receiver. In *Proceedings of PIMRC*. (pp. 1372–1377). IEEE.
10. Biaz, S., & Wu, S. (2008). Rate adaptation algorithms for ieee 802.11 networks: A survey and comparison. In *IEEE Symposium on Computers and Communications*. (pp. 130–136). IEEE.
11. Das, S., Barman, S., & Bhunia, S. (2014). Performance analysis of ieee 802.11 rate adaptation algorithms categorized under rate controlling parameters. In *Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies*. ACM.
12. IEEE Comptuer Society. IEEE standard 802.11-2012 (2012).
13. Yin, W., Hu, P., & Indulska, J. (2015, March 4). Rate control in the mac80211 framework: Overview, evaluation and improvements. *Computer Networks*.
14. IEEE standard definitions of terms for antennas. (1983, June). *IEEE Std*, *145–1983*, 1–31.
15. Yin, W., Hu, P., Indulska, J., & Bialkowski, K. (2012, October). Performance of mac80211 rate control mechanisms. In *Proceedings of ACM MSWiM*. Miami, FL, USA.
16. Kim, J., Kim, S., Choi, S., & Qiao, D. (2006, April). CARA: collision-aware rate adaptation for IEEE 802.11 WLANs. In *Proceedings of INFOCOM*. (pp. 1 –11).
17. Acharya, P. A. K., Sharma, A., Belding, E. M., Almeroth, K. C., & Papagiannaki, K. (2008). Energy-efficient pcf operation of ieee 802.11a wireless lan. In *Proceedings of SECON'08*. (pp. 1–9).
18. Adame, T., Bel, A., Bellalta, B., Barcelo, J., & Oliver, M. (2014, 01 Dec). IEEE 802.11AH: The WiFi approach for M2M communications. In *IEEE Wireless Communications*. (pp. 144–152).
19. Tian, L., Famaey, J., & Latré, S. (2016, June). Evaluation of the IEEE 802.11ah Restricted Access Window mechanism for dense IoT networks. In *Proceedings of IEEE 17th WoWMoM*. (pp. 1–9).
20. Tian, L., Mehari, M., Santi, S., Latré, S., De Poorter, E., & Famaey, J. (2018). IEEE 802.11ah restricted access window surrogate model for real-time station grouping. In *Proceedings of IEEE WoWMoM*, 06 2018.
21. Khorov, E., Lyakhov, A., & Yusupov, R. (2018). Two-Slot Based Model of the IEEE 802.11ah Restricted Access Window with Enabled Transmissions Crossing Slot Boundaries. In *Proceedings of IEEE WoWMoM*. 06 2018.
22. Vutukuru, M., Balakrishnan, H., & Jamieson, K. (2009). Cross-layer wireless bit rate adaptation. In *Proceedings of SIGCOMM*. (pp. 3–14). ACM.
23. Glass, S., Guerin, J., Hu, P., Portmann, M., & Tan, W. L. (April 2013). Specification versus reality: Experimental evaluation of link capacity estimation in ieee 802.11. In *Wireless Communications and Networking Conference (WCNC)*. 2013 IEEE, (pp. 380–385).
24. Karmerman, Ad, & Monteban, L. (1997). WaveLAN-II: A high-performance wireless LAN for the unlicensed band. *Bell Labs Technical Journal*, *2*(3), 118–133. (Special Issue: Wireless Autumn (Fall) 1997)
25. Lacage, M., Manshaei, M. H., & Turletti, T. (2004). IEEE 802.11 rate adaptation: A practical approach. In *Proceedings of the 7th ACM MSWiM*. (pp. 126–134), Venice: ACM.
26. Wong, S. H. Y., Yang, H., Lu, S., & Bharghavan, V. (2006). Robust rate adaptation for 802.11 wireless networks. In *Proceedings of MOBICOM*. (pp. 146–157). Los Angeles, CA: ACM.
27. Shaoen, W., & Biaz, S. (2007, June). ERA: Efficient rate adaptation algorithm with fragmentation. In *Auburn University*, Technique Report.
28. Pefkianakis, I., Wong, S. H. Y., Yang, H., Lee, S.-B., & Lu, S. (2013). Toward history-aware robust 802.11 rate adaptation. *IEEE Transactions on Mobile Computing*, *12*(3), 502–515.
29. Onoe specification. Retrieved 2016 from http://sourceforge.net/project/madwifi.
30. Bicket, J. C. (2005). *Bit-rate Selection in Wireless Networks*. PhD thesis, MIT Master's Thesis.
31. Wang, S.-C., & Helmy, A. (2011). BEWARE: Background traffic-aware rate adaptation for IEEE 802.11. *IEEE/ACM Transactions on Networking (TON)*, *19*(4), 1164–1177.
32. Yin, W., Hu, P., Indulska, J., Portmann, M., & Guerin, J. (2012). Robust MAC-layer rate control mechanisms for 802.11 wireless networks. In *Proceedings of LCN*. (pp. 419–427).
33. Guerin, J., Portmann, M., Bialkowski, K., Tan, W, L., & Glass, S. (2010). Low-cost wireless link capacity estimation. 01. In *ISWPC'10: Proceedings of the 5th IEEE international conference on Wireless pervasive computing*. (pp. 343–348).
34. Pefkianakis, I., Lee, S.-B., & Lu, S. (2013). Towards MIMO-aware 802.11n rate adaptation. *IEEE/ACM Transaction on Networking*, *21*(3), 692–705.
35. Byeon, S., Yoon, K., Yang, C., & Choi, S. (2017, May). Strale: Mobility-aware phy rate and frame aggregation length adaptation in WLANS. In *Proceedings of IEEE INFOCOM*. (pp. 1–9).

36. Sadeghi, B., Kanodia, V., Sabharwal, A., & Knightly, E.. (2002). Opportunistic media access for multirate ad hoc networks. In *Proceedings of MOBICOM*. (pp. 24–35). ACM.
37. Zhang, J., Tan, K., Zhao, J., Wu, H., & Zhang, Y. (2008, April). A practical snr-guided rate adaptation. In *Proceedings of INFOCOM*. (pp. 2083 –2091).
38. Pavon, Jd. P., & Choi, S. (2003). Link adaptation strategy for IEEE 802.11 WLAN via received signal strength measurement. In *Proceedings of ICC*, (vol. 2, pp. 1108 –1113).
39. Rahul, H., Edalat, F., Katabi, D., & Sodini, C. G. (2009). Frequency-aware rate adaptation and mac protocols. In *Proceedings of MOBICOM*. (pp. 193–204), Beijing, China: ACM.
40. Lee, H., Kim, H.-S., & Bank, S. (2014). InFRA: In-frame rate adaptation in fast fading channel environments. In *Proceedings of IEEE ICC*, (pp. 2885–2890).
41. Okhwan, L., Jihoon, K., Jongtae, L., & Sunghyun, C. (2015, January). SIRA: SNR-Aware intra-frame rate adaptation. In *IEEE Communications Letters*, (pp. 90–93).
42. Halperin, D., Hu, W., Sheth, A., & Wetherall, D. (2010). Predictable 802.11 packet delivery from wireless channel measurements. In *Proceedings of SIGCOMM*, (pp. 159–170). New York, NY: ACM.
43. Goldsmith, A. (2005). *Wireless communications*. Cambridge: Cambridge University Press.
44. Shen, W.-l., Tung, Y.-C., Lee, K.-C., Lin, K. C.-J., Shyamnath, G., Dina, K., & Chen, M.-S. (2014, January). Rate adaptation for 802.11 multiuser mimo networks. In *IEEE Transaction on Mobile Computing*, (pp. 35–47).
45. Guillaud, M., Slock, D. T. M., & Knopp, R. (2005, August). A practical method for wireless channel reciprocity exploitation through relative calibration. In *Proceedings of the Eighth International Symposium on Signal Processing and Its Applications, 2005*, (vol. 1, pp. 403–406).
46. Song, L., & Wu, S. (2013). AARC: Cross-layer wireless rate control driven by fine-grained channel assessment. In *Proceedings of IEEE International Conference on Communications (ICC)*. (pp. 3311–3316).
47. Shamy, K., Assi, C., & EI-Najjar, J. (2008). Efficient rate adaptation with QoS support for wireless networks. In *Proceedings of IEEE GlOBECOM*. (pp. 1–6).
48. Lee, T.-H., Marshall, A., & Zhou, B. (2006). A qos-based rate adaptation strategy for IEEE a/b/g PHY Schemes using IEEE 802.11e in ad-hoc networks. In *Proceedings of International Conference on Networking and Services (ICNS)*, (pp. 113–118).
49. Haratcherev, I., Langendoen, K., Lagendijk, R., & Sips, H. (2004). Hybrid rate control for IEEE 802.11. In *Proceedings of MobiWac'04*. (pp. 10–18), New York, NY: ACM.
50. Kamoltham, N., Nakorn, K. N., & Rojviboonchai, K. (2012). From NS2 to NS3 implementation and evaluation. In *Proceedings of Computing, Communications and Applications Conference (ComComAp)*, (pp. 35–40).
51. Comparetto, G., Hallenbeck, P., Mirhakkak, M., Schult, N., Wade, R., & DiGennaro, M. (2011). Verification and validation of the QualNet JTRS WNW and SRW waveform models. In *Proceedings of Military Communications (MILCOM)*, (pp. 1818–1826).
52. Raychaudhuri, D., Seskar, I., Ott, M., Ganu, S., Ramachandran, K., Kremo, H., Siracusa, R., Liu, H., & Singh, M. (2005). Overview of the ORBIT radio grid testbed for evaluation of next-generation wireless network protocols. In *Proceedings of IEEE Wireless Communications and Networking Conference*. (pp. 1664–1669).
53. Bonney, J., Bowering, G., Marotz, R., & Swanson, K. (2008). Hardware-in-the-loop emulation of mobile wireless communication environments. In *Proceedings of IEEE Aerospace Conference*, (pp. 1–9).
54. Bialkowski, K., & Portmann, M. (2010). Design of test-bed for wireless mesh networks. In *IEEE Antennas and propagation International Symposium*, Toronto, Canada.
55. Biaz, S., & Wu, S. (2008). Rate adaptation algorithms for IEEE 802.11 networks: A survey and comparison. In *Proceedings of IEEE Symposium on Computers and Communications*.
56. Judd, G., Wang, X., & Steenkiste, P. (2008). Efficient channel-aware rate adaptation in dynamic environments. In *Proceedings of ACM MobiSys*, (pp. 118–131). New York, NY.
57. Camp, J., & Knightly, E. (2008). Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. In *Proceedings of MOBICOM*. (pp. 315–326), San Francisco, CA: ACM.
58. Yin, W., Bialkowski, K., Indulska, J., & Hu, P. (2010). Evaluations of MadWifi MAC layer rate control mechanisms. In *Proceedings of IWQoS*.
59. Pefkianakis, I., Hu, Y., Wong, S. H. Y., Yang, H., & Lu, S. (2010). MIMO rate adaptation in 802.11n wireless networks. In *Proceedings of MOBICOM*. Chicago, Illinois, USA, Sep. 20–24.
60. Ancillotti, E., Bruno, R., & Conti, M.. (2008). Experimentation and performance evaluation of rate adaptation algorithms in wireless mesh networks. In *Proceedings of the 5th ACM symposium on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks*, (pp. 7–14), Vancouver, BC: ACM.
61. CMU. (2006). A controlled wireless networking testbed based on a wireless signal propagation emulator. http://www.cs.cmu.edu/emulator
62. Xia, D., Hart, J., & Fu, Q. (June 2013). Evaluation of the minstrel rate adaptation algorithm in IEEE 802.11g wlan. In *Proceedings of ICC, Budapest*. Hungary.
63. Gast, M.S. (2015). 802.11ac: A survival guide. Second release, O'REILLy.
64. Wang, S.-C., & Helmy, A. (2008, June). Beware: Background traffic-aware rate adaptation for ieee 802.11. In *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, (pp. 1–12).
65. Lin, C.-H., Kate Lin, C.-J., & Chen, W.-T. (2014). Rate adaptation for highly dynamic body area networks. In *Proceedings of CPSCom*.
66. Yao, Y., Zhou, X., & Zhang, K.. (2014, July). Density-aware rate adapation for vehicle safety communications in the highway environment. In *IEEE Communications Letters*, (pp. 1167–1170).
67. Maskooki, A., Soh, C. B., Gunawan, E., & Low, K. S. (2013). Ultra-wideband real-time dynamic channel characterization and system-level modeling for radio links in body area networks. *IEEE Transactions on Microwave Theory and Techniques*, *61*(8), 2995–3004.
68. Choudhury, R. R., & Vaidya, N. H. (2003). *Deafness: A mac problem in ad hoc networks when using directional antennas*. Technical Report.
69. Camp, J., & Knightly, E. (2010). Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation. *IEEE/ACM Transactions on Networking*, *18*(6), 1949–1962.
70. Choi, H., Gong, T., Kim, J., SHin, J., & Lee, S.-J. (2019). Dissecting 802.11ac performance—Why you should turn off mu-mimo (poster). In *Proceedings of the 17th ACM MObISys, MobiSys '19*. (pp. 510–511), New York, NY, USA. ACM.
71. Sur, S., Pefkianakis, I., Zhang, X., & Kim, K.-H. (2016). Practical mu-mimo user selection on 802.11ac commodity networks. In *Proceedings of ACM MoBiCom, MobiCom '16*. (pp. 122–134).
72. Akhtar, A., & Ergen, S. C. (2018). Directional mac protocol for ieee 802.11ad based wireless local area networks. *Ad Hoc Networks*, *69*, 49–64.

73. Kiran, M. P. R. S., & Rajalakshmi, P. (2019). Saturated Throughput Analysis of IEEE 802.11ad EDCA For High Data Rate 5G-IoT Applications. *IEEE Transactions on Vehicular Technology*, *68*(5), 4774–4785.
74. Praveen Kumar, K., & Govindaraj, E. (2019). Quality enhancement with fault tolerant embedding in video transmission over wmsns in 802.11e wlan. *Ad Hoc Networks*, *88*, 18–31.
75. Part 11: Wireless lan medium access control (mac) and physical layer (phy) specications - amendment 3: Mac enhancements for robust audio video streaming. IEEE Amendment 802.11aa, (2012).
76. Zawia, H. I., Hassan, R., & Dahnil, D. P. (2018). A survey of medium access mechanisms for providing robust audio video streaming in ieee 802.11aa standard. *IEEE Access*, *6*, 27690–27705.
77. Gringoli, F., Serrano, P., Ucar, I., Facchi, N., & Azcorra, A. (2018). Experimental qoe evaluation of multicast video delivery over ieee 802.11aa wlans. *IEEE Transactions on Mobile Computing*, (pp. 1–1).
78. Mansour, K., Jabri, I., & Ezzedine, T. (2017). A multicast rate adaptation algorithm over IEEE 802.11aa GCR block ack scheme. In *Proceedings of 13th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE*.
79. Salvador, P., Cominardi, L., Gringoli, F., & Serrano, P. (2013). A first implementation and evaluation of the ieee 802.11 aa group addressed transmission service. *ACM SIGCOMM Computer Communication Review*, *44*(1), 35–41.
80. Yin, W., Hu, P., Wang, W., Wen, J., & Zhou, H. (2020). FASUS: A fast association mechanism for 802.11ah networks. *Computer Networks*, 03.
81. Aust, S., Venkatesha Prasad, R., & Niemegeers, I. G. M. M. (2012). IEEE 802.11ah: Advantages in standards and further challenges for sub 1 GHz Wi-Fi. In *Proceedings of IEEE ICC*. IEEE.
82. Ali, M. Z., Mišić, J., & Mišić, V. B. (2019). Performance evaluation of heterogeneous iot nodes with differentiated qos in ieee 802.11ah raw mechanism. *IEEE Transactions on Vehicular Technology*, *68*(4), 3905–3918.
83. Aust, S., Venkatesha Prasad, R., & Niemegeers, I. G. M. M. (2015). Outdoor long-range wlans: A lesson for ieee 802.11ah. *IEEE Communication Surveys & Tutorials*, *17*(3), 1761–1775.
84. Khorov, E., Kiryanov, A., Lyakhov, A., & Bianchi, G. (2019, First quarter). A tutorial on IEEE 802.11ax High Efficiency WLANs. *IEEE Communications Surveys Tutorials*, 21(1):197–216.
85. Bellalta, B. (2016). Ieee 802.11ax: High-efficiency wlans. *IEEE Wireless Communications*, *23*(1), 38–46.
86. Sur, S., Pefkianakis, I., Zhang, X., & Kim, K.-H. (2016). Practical mu-mimo user selection on 802.11 ac commodity networks. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, (pp. 122–134). ACM.
87. Garroppo, R.G., Giordano, S., Lucetti, S., & Tavanti, L. (2007). Providing air-time usage fairness in IEEE 802.11 networks with deficit transmission time (DTT) scheduler. In *Wireless Networks*, (pp. 481–495).
88. Bhanage, G., Vete, D., Seskar, I., & Raychaudhuri, D. (2010). SplitAP: Leveraging wireless network virtualization for flexible sharing of WLANs. In *Proceedings of GLOBECOM*, (pp. 1–6).
89. Yin, W., Peizhao, H., Wen, J. & Zhou, H. (2020). Ack spoofing on mac-layer rate control: Attacks and defenses. *Computer Networks*.
90. Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer networks*, *54*(15), 2787–2805.
91. Kim, D.I., Le, L.B., & Hossain, E. (2008). Joint rate and power allocation for cognitive radios in dynamic spectrum access environment. In *IEEE Transactions on Wireless Communications*, (pp. 5517–5527).
92. Wu, D., Ci, S., Luo, H., Zhang, W., & Zhang, J. (2012). Cross-layer rate adaptation for video communications over LTE networks. In *Proceedings of GLOBECOM*, (pp. 4834–4839). IEEE.
93. Orakcal, C., & Starobinski, D. (2014). Jamming-resistant rate adaptation in Wi-Fi networks. *Performance Evaluation*, *75*, 50–68.
94. Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, *13*(18), 1587–1611.
95. Yin, W., Hu, P., Indulska, J., & Bialkowski, K. (2010). A method to improve adaptability of the minstrel mac rate control mechanism. In *UIC'2010 Proceeding of IEEE International Conference on Ubiquitous Intelligence and Computing (submitted)*.
96. Li, C.-Yu, Peng, Chunyi, Lu, Songwu, & Wang, Xinbing. (2012). Energy-based rate adaptation for 802.11n. In *Proceedings of ACM Mobicom*. (pp. 341–352).
97. Li, C.-Y., Peng, C., Cheng, P., Lu, S., Wang, X., Ren, F., et al. (2016). An energy efficiency perspective on rate adaptation for 802.11 n nic. *IEEE Transactions on Mobile Computing*, *15*(6), 1333–1347.
98. Jung, K.-H., Suh, Y.-J., & Yu, C.. (2013). Joint rate and voltage adaptation to save energy of software radios in underutilized WLAN. In *Proceedings of WCNC*, (pp. 163–168). IEEE.
99. Khan, M. O., Dave, V., Chen, Y.-C., Jensen, O., Qiu, L., Bhartia, A., & Rallapalli, S. (2013). Model-driven energy-aware rate adaptation. In *Proceedings of ACM MobiHoc*, (pp. 217–226).
100. Ucar, I., Donato, C., Serrano, P., Garcia-Saavedra, A., Azcorra, A., & Banchs, A. (2016). Revisiting 802.11 rate adaptation from energy consumption's perspective. In *Proceedings of ACM MSWiM*, (pp. 27–34).
101. Ghadimi, E., Calabrese, F. D., Peters, G., & Soldati, P. (2016). A reinforcement learning approach to power control and rate adaptation in cellular networks. *Optimization and Control*, *12*, 1–7.



Wei Yin received his Ph.D. from the University of Queensland in 2012. His research interest includes wireless networks, Internet routing, and network security. He has published many papers on prestigious international journals and conferences, such as Computer Networks, IEEE LCN, IEEE IWQoS, ACM MSWiM, etc. He is the reviewer for many journals and conferences including Computer Networks, Computer Communcations, IEEE LCN and IEEE WCNC.

Peizhao Hu is an Assistant Professor in Rochester Institute of Technology. His research interest includes mobile and pervasive computing, homomorphic encryption, network security, and wireless networking.

Jadwiga Indulska is an Emeritus Professor of Pervasive Computing in the School of Information Technology and Electrical Engineering at The University of Queensland, Brisbane, Australia. Her research interests are in the areas of pervasive computing, distributed computing and computer networks. Over the last two decades, her research has addressed many research problems in pervasive and autonomic computing, including context information models for context-aware applications; autonomic management of context information; privacy of context information; software engineering of context-aware applications; balancing user control and software autonomy; autonomic rapidly deployable mesh networks and Software Defined Networks.

Marius Portmann received his Ph.D. in Electrical Engineering from the Swiss Federal Institute of Technology (ETH, Zurich) in 2002. He currently is an Associate Professor at The University of Queensland, Australia. His research interests include general networking, in particular SDN and wireless networks, as well as Cyber Physical Systems.

Ying Mao is an Assistant Professor in the Department of Computer and Information Science at Fordham University in the New York City. In addition, Dr. Mao is a Fordham-IBM Research Fellow. He received his Ph.D. in Computer Science from University of Massachusetts Boston advised by Dr. Bo Sheng In addition, he obtained a M.S. degree in Electrical Engineering from University at Buffalo and a B.S. degree in Telecommunication from Commanding Communication Academy (now National University of Defense Technology, Wuhan Campus, China). Previously, Dr. Mao was an Assistant Professor at The College of New Jersey.