Rate Adaptive Broadcast in Internet of Things

Linghe Kong[1(✉)], Zhe Wang[1], Yongshuai Duan[1], Tong Meng[2], Fan Wu[1], and Guihai Chen[1]

[1] Shanghai Jiao Tong University, Shanghai, China
{linghe.kong,wang-zhe,dys1998}@sjtu.edu.cn, {fwu,gchen}@cs.sjtu.edu.cn
[2] University of Illinois at Urbana-Champaign, Urbana, IL, USA
tongm2@illinois.edu

Abstract. This paper presents Rate Adaptive Broadcast (RAB), a novel wireless design that enables the rate adaptive broadcast in Internet of things (IoT). Broadcast is common in IoT due to the ubiquitous tree topologies. Channel resource is usually underused in broadcast because there is no rate adaptation in conventional broadcast and the data rate is always set as the lowest one by default. Existing rate adaptation methods work only for unicast or multicast, relying on information interaction between senders and receivers. These methods cannot directly apply in broadcast, which is a one-way transmission without acknowledgement (ACK). It is also impractical to transplant conventional ACK into broadcast, otherwise, massive ACKs will lead to a heavy overhead. To tackle this dilemma, we propose RAB, which allows the sender to broadcast data ceaselessly while adjusting the data rate according to real-time channel states. The core contribution is the subtly designed feedbacks that can be concurrently delivered and do not affect any reception. We implement RAB on USRPs and establish a 20-node IoT testbed. Experiment results demonstrate that the throughput is largely improved. The throughput of RAB is 2.8x of the standard WiFi and 1.3x of MuDRA, the state-of-the-art multicast rate adaptation method.

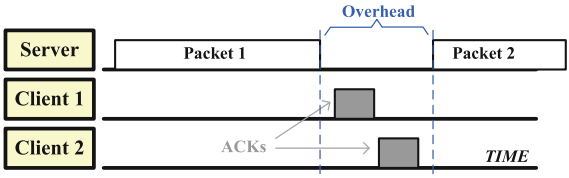
Keywords: Internet of things · Rate adaptation · Acknowledgement · Wireless broadcast

1 Introduction

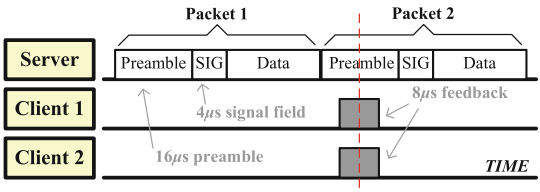
Wireless broadcast is an efficient solution to deliver data that one sender (server) transmits data to all neighbors (clients) simultaneously. Its value lies in plenty of Internet of things (IoT) applications. For example, data dissemination in Internet of vehicles [19] and data sharing in collaborative robots [7].

However, the throughput of wireless broadcast is extremely low. Using 802.11g WiFi as an example, the data rate of broadcast is always set at the lowest 6 Mbps. Field tests [10] reveal that even if three clients connecting to

S. Guo et al. (Eds.): CWSN 2019, CCIS 1101, pp. 61–75, 2019.
https://doi.org/10.1007/978-981-15-1785-3_5



(a) Time allocation of existing methods.



(b) Time allocation of RAB.

Fig. 1. Compared with existing rate adaptation methods, RAB removes the overhead.

one AP try to watch the same video, the performance is abysmally poor. The performance of wireless broadcast leaves much to be desired.

Rate adaptation is the fundamental technique to improve the throughput in dynamic wireless channel. Existing studies usually focus on unicast and multicast. In unicast, diverse rate adaptation methods have been investigated using frequency [9], constellation [11], collision [5], or SNR [18] analysis. All these methods depend on the information exchange between sender and receiver. In multicast, recent studies allow only partial clients to reply in order to balance accuracy and overhead. For example, in REMP [8], only NACKs are sent. In MuDRA [3], the server collects ACKs from representative clients by a lightweight protocol. Nevertheless, these methods are impractical in broadcast because: (i) there is no ACK in broadcast; (ii) the number of clients may be large and their channel states are different. If the conventional ACK mechanism is adopted, heavy overhead will be introduced due to a large amount of clients, largely reducing the throughput.

We observe that the preamble in WiFi has opportunity to improve this problem. The essence of preamble is a training sequence at the packet header. Even partial samples in the preamble cannot be decoded, the packet still can be successfully received. Our main idea is that the server keeps broadcasting packets while all clients concurrently transmit their feedbacks for rate adaptation during the preamble time as shown in Fig. 1.

There are two major challenges to realize this idea. First, since all feedbacks and the preamble are parallel, they are collided together. Processing this overlapped signal is difficult because the interference is from not only the server but also all clients. Second, the duration of preamble is only 16 μs in WiFi, equal to four OFDM symbol length. When numerous clients exist, it is not easy to design such short feedbacks containing all required and decodable information.

Table 1. IEEE 802.11a/g data rate information.

Modulation	Coding	Data rate	Effective SNR
BPSK	1/2	6 Mbps	<6.6 dB
BPSK	3/4	9 Mbps	6.6–8.7 dB
QPSK	1/2	12 Mbps	8.7–9.6 dB
QPSK	3/4	18 Mbps	9.6–17.3 dB
16QAM	1/2	24 Mbps	17.3–18.4 dB
16QAM	3/4	36 Mbps	18.4–26.0 dB
64QAM	2/3	48 Mbps	26.0–28.1 dB
64QAM	3/4	54 Mbps	>28.1 dB

To tackle these challenges, we propose the *rate adaptive broadcast* (RAB). Fully exploiting the WiFi subcarrier, every client just transmits a two-symbol-length feedback, while the server extracts the needed information from the collisions of all clients. The other advantages of RAB include: the subtly designed feedback has the same structure of standard OFDM symbols. So it is easy to be implemented and is compatible to commercial WiFi devices; RAB is also a general solution, which can be extended to other wireless protocols.

The main contributions of this paper are as follows:

- We design RAB that enables rate adaptive broadcast in IoT. The core design of RAB leverages the preamble to realize the concurrent transmission and leverages the orthogonal subcarriers to develop the short feedbacks.
- We implement RAB on USRPs, and establish a 20-node testbed. Experiment results demonstrate that RAB achieves the mean throughput gain of 2.8x and 1.3x compared with the standard WiFi and the state-of-the-art MuDRA, respectively.

2 Problem Statement

Before introducing RAB, we review the background and understand the WiFi preamble. Then, we state our problem and summarize the design challenges.

2.1 Background

Rate adaptation is a fundamental primitive in wireless networks. Since the channel state varies unpredictably, a sender has to measure the dynamic channel and selects the appropriate data rate to maximize the throughput. Taking IEEE 802.11a/g WiFi as an example, eight different data rates are available including 6, 9, 12, 18, 24, 36, 48, 54 Mbps. We list the data rate information in Table 1 and with the effective SNR measured by our experiment (details in Sect. 4).

Broadcast, multicast, and unicast are three general communication modes in WiFi. If clients have interests in the same content, broadcast is the most

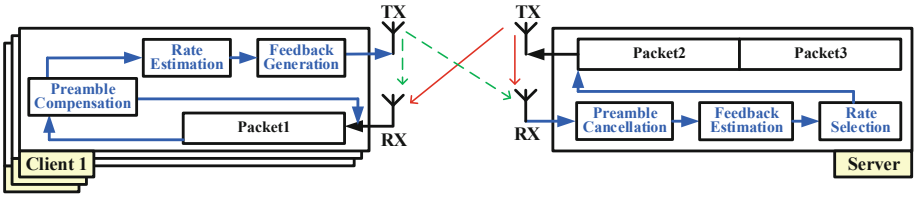


Fig. 2. RAB architecture.

efficient mode that utilizes one channel to transmit data to all clients. In IEEE 802.11a/g, if the broadcast mode is chosen, the data rate is fixed at the lowest 6 Mbps. Due to no ACK in broadcast, the server cannot acquire different channel states from all clients, so the lowest rate is the conservative setting.

2.2 Understanding of WiFi Preamble

Preamble is a pre-defined training signal at the packet header. Nearly all wireless protocols require preambles, because packet detection and time synchronization between sender and receiver totally rely on it. WiFi's preamble is 16 µs length consisting of an 8 µs *short training field* (STF) and an 8 µs *long training field* (LTF). STF is the 10 repetitions of a given 16-sample sequence and LTF is the 2.5 repetitions of a 64-sample sequence, where the repetition pattern is a 32-sample *cyclic prefix* (CP) and then two 64-sample signals. When a client receives a preamble, it operates the correlation by known STF and LTF sequences. Through the 10 correlation peaks in STF and 2 correlation peaks in LTF, the client can detect the start-of-packet and complete the time synchronization.

Following the preamble is the *signal field* (SIG), which contains the information of data rate and packet length. Specially, the *reserved* bit is intended for future use.

2.3 Problem, Challenges, and Observations

In IoT, the lowest data-rate broadcast obviously lowers the quality of experience. Motivated by fully exploiting the channel resource and improving the throughput, we propose to study the rate adaptation problem in WiFi broadcast.

Rate adaptation and broadcast never work collaboratively in standard WiFi. On one hand, an accurate adaptation depends on the two-way interaction to measure all channel states. However, the broadcast is a one-to-many and one-way transmission mode without ACKs. On the other hand, existing reactive rate adaptation methods, used in unicast and multicast, cannot directly apply in broadcast. Since there may be numerous clients in broadcast case, one-by-one ACKs result in a heavy overhead.

Inspired by full duplex [1] and concurrent transmission [12], we conceive a concurrent-feedback design to tackle the above dilemma. In addition, we observe two opportunities from the packet structure, which are able to facilitate the

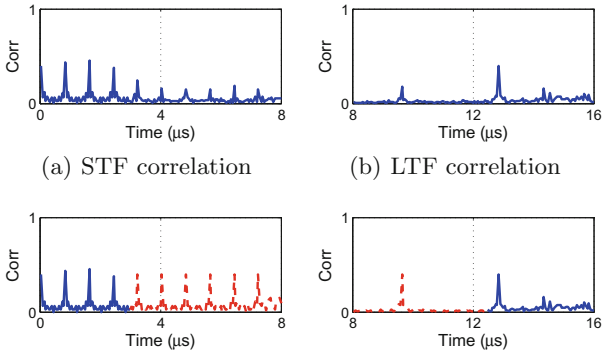


Fig. 3. The correlation and recovered results on the collided signal.

design. First, the essence of the preamble is a training sequence. Partial missing preamble will not lead to any data loss. Moreover, the missing part is potential to be compensated by the known sequence and channel coherence. Second, the reserved bit in signal field can be used to transmit 1-bit information.

3 Design of RAB

Based on the observations, we design the *rate adaptive broadcast* (RAB) to bridge rate adaptation and broadcast in IoT. The block diagram of RAB architecture is shown in Fig. 2, including the designs of client and server. In RAB, every device equips one TX and one RX antennas, which is a usual configuration in commercial WiFi devices.

- At the client side, three modules are added. After receiving a packet, the *preamble compensation* module is ready to compensate the collided preamble of next packet; while the *rate estimation* module estimates the supported data rate by analyzing the received packet. Then, the *feedback generation* module generates the short RAB feedback implying the estimated rate and transmit it to the server during the preamble time.
- At the server side, three modules are added. The server receives an overlapped signal containing the clients' feedbacks and its own preamble. The function of *preamble cancellation* filters the feedbacks out from the preamble. However, the multiple feedbacks are still overlapped. So the *feedback estimation* module recognizes every client's information from the overlapped feedbacks. According to the recognized information, the *rate selection* determines the optimal data rate.

The greatest strength of RAB is to enable the rate adaptation in broadcast without extra time overhead. The server fully exploits the channel in time domain by broadcasting data ceaselessly. On the contrary, all clients receive the data in

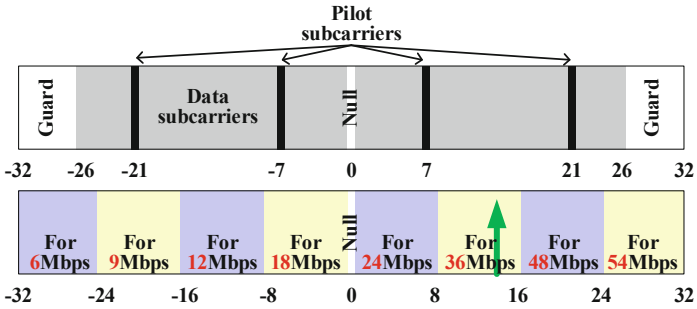


Fig. 4. Subcarrier allocation for OFDM symbol (up) and RAB symbol (down).

most time. They only concurrently transmit their 8 µs feedbacks within the 16 µs preamble time (align center).

It is not easy to realize RAB. The first challenge is the collision resolution. The severe collision caused by the parallel server's preamble and clients' feedbacks. Another challenge is how to design the short feedback so that the server can recognize the information from the parallel feedbacks. Next, we describe the design of every module in details.

3.1 Client: Preamble Compensation Module

Every client is only interested in the preamble, but the preamble is hard to be extracted from the collided signal. Fortunately, unlike the payload, the preamble is a training sequence attaching no essential data. Hence, a client just needs to compensate the preamble.

First, for start-of-packet detection and time synchronization tasks, the *preamble compensation* module directly operates the correlation on the collided signal by 16-sample STF sequence and 64-sample LTF sequence. The correlation results are shown in Fig. 3(a) and (b). Although several correlation peaks are distorted, their number and locations are clearly recognized. Hence, the detection and synchronization tasks can be accomplished as usual.

Then, for AGC, this module leverages the feature that the received preamble is partially overlapped. Every feedback signal is 8 µs-duration overlapped at the center of the 16 µs preamble. So the fist 80 samples in STF and last 80 samples in LTF are nearly non-collided. To be conservation, we use the first correlation peak in STF and the last correlation peak in LTF to recover the other peaks in Fig. 3(c) and (d). Therefore, AGC task is accomplished.

3.2 Client: Rate Estimation Module

The *rate estimation* module aims to assess the maximal supported data rate. In literature, plenty of metrics have been adopted to assess the data rate such as constellation [11], collision [5], and SNR [18]. This module is open to any existing

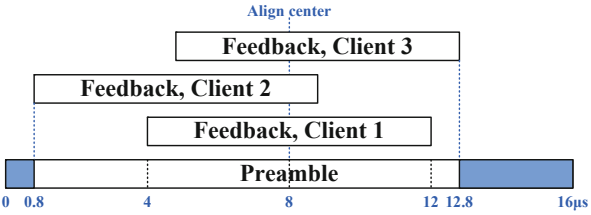


Fig. 5. The synchronization goal is the 'align center' between preamble and RAB feedback. However, it can be tolerant as long as the feedback falls in 0.8–12.8 μs range.

method as long as the data rate could be accurately and quickly estimated. In current RAB, we adopt the classic metric *effective SNR* [4]. Yet SNR is coarse and insufficient, the effective SNR is more accurate to adapt the rate in dynamic channel.

When a client receives the packet, the maximal supported rate is obtained by the following steps: (i) calculate CSI by the received packet (ii) the MMSE expression is used to compute subcarrier SNRs from the CSI; (iii) the effective SNR is computed from the subcarrier SNRs; and (iv) assess the maximal supported data rate through the effective SNR.

Since the effective SNR is a mature technique, we just briefly introduce how to compute the effective SNR. First,

$$\mathcal{B}_{eff,k} = \frac{1}{52} \sum_{s=-26}^{26} f_{\mathcal{S}\to\mathcal{B},k}(\mathcal{S}_s), \tag{1}$$

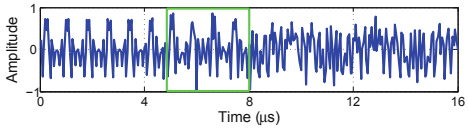
where $\mathcal{B}_{eff,k}$ is the effective BER at the pre-set data rate k, $k \in \{6, 9, \cdots, 54\}$, s is the indicator of subcarriers belonging to $[-26, 26]$ and $s \neq 0$, $f_{\mathcal{S}\to\mathcal{B},k}()$ is the mapping function from SNR to BER, and $\mathcal{S}_s$ is the SNR of the s-th subcarrier. According to different k, $\mathcal{B}_{eff,k}$ needs to be computed respectively because of different mapping function. For example, if $k = 6$ Mbps, $f_{\mathcal{S}\to\mathcal{B},k}(\mathcal{S}_s) = Q(\sqrt{2\mathcal{S}_s})$, where Q is the standard normal CDF; if $k = 48$ Mbps, $f_{\mathcal{S}\to\mathcal{B},k}(\mathcal{S}_s) = \frac{7}{12}Q(\sqrt{\mathcal{S}_s/21})$; the mapping functions for the other data rates can be found in [4]. Then,

$$\mathcal{S}_{eff,k} = f_{\mathcal{B}\to\mathcal{S},k}(\mathcal{B}_{eff,k}), \tag{2}$$

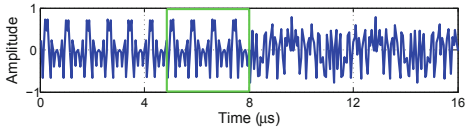
where $\mathcal{S}_{eff,k}$ is the effective SNR at data rate k and $f_{\mathcal{B}\to\mathcal{S},k()}$ is the inverse function of $f_{\mathcal{S}\to\mathcal{B},k}()$.

3.3 Client: Feedback Generation Module

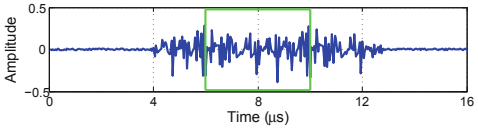
After the data rate is estimated, the client needs to generate the feedback containing this data rate information. Since a server cannot decomposed the collision of conventional ACKs, it is necessary to design a novel pattern of feedback, refer to *RAB feedback*. In RAB, the feedbacks are overlapped in time domain, we conceive to distinguish them from the frequency domain.



(a) Signal with feedbacks and preamble.



(b) Non-collided signal w/ only preamble.



(c) Get the feedback by substraction.

Fig. 6. The process of preamble cancellation.

Let us first understand the OFDM symbol, which is the least transmission unit. Each OFDM symbol consists of 64 orthogonal subcarriers in frequency domain and 80 samples in time domain. The allocation of these subcarriers is illustrated in Fig. 4(up).

Based on the OFDM symbol, we design the *RAB symbol*, a customized symbol for RAB feedback. Each RAB symbol also consists of 64 subcarriers. The allocation of these subcarriers is illustrated in Fig. 4(down), where 64 subcarriers are evenly divided into 8 groups mapping to 6, 9, 12, 18, 24, 36, 48, 54 Mbps, respectively. According to its maximal supported data rate, a client randomly selects one subcarrier within the corresponding group to transmit a peak and all the other subcarriers are set null. Then, operating IFFT on the RAB symbol, the 64 corresponding samples are obtained in time domain. Repeating 2.5 times of these samples, we have the 8 μs RAB feedback (160 samples).

The advantages of RAB feedback include: (i) The RAB symbol is compatible to commercial WiFi devices, because the framework of RAB symbol is the same as OFDM symbol. (ii) All 64 subcarriers are fully exploited to maximize the number of different feedbacks. The guard and pilot subcarriers are unnecessary because the feedback is short and simple. (iii) The length of feedback is minimized in order to reduce the interference on preamble.

Besides generating the RAB symbol, another job of the *feedback generation* module is to synchronize the feedback transmission. The goal is to align center between the preamble and the feedbacks as shown in Fig. 5, i.e., transmitting every feedback at the 4–12 μs position of the preamble. So that the first 4 μs of STF and the last 4 μs LTF can be non-collided for training tasks. This goal is

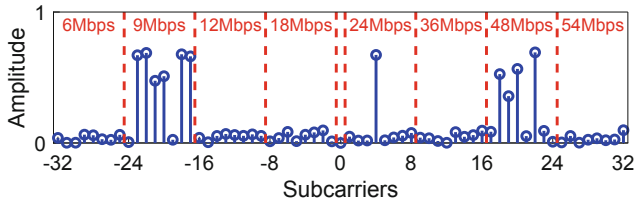


Fig. 7. Transform the feedback signal from time to frequency domain.

approached by two steps. First, since the finish time of current packet reception is known as t and the server broadcasts its packets ceaselessly, the feedback can be transmitted at $t+4\mu s$. Second, the client could detect the time offset between the preamble and its own feedback.

Even utilizing these two steps, the synchronization may be not prefect because of hardware limitation. Fortunately, some offset can be tolerant in our design. We find that the first repetition of STF sequence finishes at 0.8 µs and the last repetition of LTF sequence starts at 12.8 µs. Thus, as long as the RAB feedback is transmitted within the range of 0.8–12.8 µs, the aforementioned two repetitions are still non-collided, which is sufficient to accomplish the training tasks. Compared with the ideal 4–12 µs position, the tolerated offset is up to 4 µs. Such a offset is easily achievable by existing technique [15], which can realize a $<$0.5 µs synchronization for concurrent transmissions.

3.4 Server: Preamble Cancellation Module

As shown in Fig. 6, the server receives a collided signal, which is dominated by its own preamble. The *preamble cancellation* module aims to cancel this preamble and filter the feedbacks out. To operate the cancellation, we introduce the non-collided preamble in the last packet. Due to the channel coherence and two consecutive packets, the last preamble is potential to be the baseline to cancel the preamble in current collision. Nevertheless, since the phase offset may exists in different complex signals, we cannot subtract the non-collided signal directly from the collided signal.

The *preamble cancellation* module operates the cancellation in frequency domain. A 64-sample time window is set to extract the time-domain signal in both the collided and non-collided signals, where the 64-sample is the least window to do FFT on 64 subcarriers. To guarantee that all feedbacks have at least 64-sample overlapped together, the minimal duration of a feedback is 8 µs. And the location of time window is from 4.8 to 8 µs. Hence, the information of all clients are included in this time window.

Operating FFT on both extracted signals and doing the cancellation by magnitude subtraction, the preamble signal is cancelled and we have all clients' feedbacks in frequency domain as shown in Fig. 7.

According to the subcarrier allocation in RAB symbol, the distribution of peaks can be counted in every data rate. Since the noise can be easily measured,

Fig. 8. RAB testbed.

a peak is determined when its amplitude is larger than the average noise. We note a subcarrier with peak as '1' and a subcarrier without peak as '0'.

3.5 Server: Feedback Estimation Module

A client randomly selects one subcarrier in the group of its data rate. Hence, it is possible that multiple clients select the same subcarrier, especially in dense case. Considering this case, the goal of the *feedback estimation* module is to compute the number of clients in every group of data rate. Various existing methods serve for cardinality estimation based on responses of '0' and '1'. For example, UPE [6] and ART [12]. This module is open to diverse such methods as long as the number of clients can be accurately estimated.

In current RAB version, we adopt the classic *unified probabilistic estimation* (UPE) [6] to realize this feedback estimation module. Denote n_0 as the number of '0' s, m as the number of subcarriers in a group, and N as the number of clients in this group. UPE estimates N on account of n_0 and m. The UPE estimator is

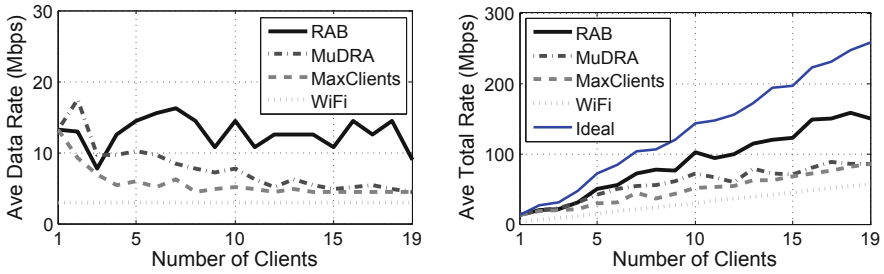
$$\widetilde{N} = -m \times \ln(\frac{n_0}{m}), \tag{3}$$

where $\widetilde{N}$ is the estimate of N. According to Eq. (3) and the setting of $m = 64/8 = 8$, in Fig. 7, there are 6 peaks can be found in the group of 9 Mbps, so $n_0 = m - 6 = 2$. Then, UPE estimates the number of clients $\widetilde{N} = -8 \times \ln(2/8) \approx 11$.

3.6 Server: Rate Selection Module

Using the result of feedback estimation, the *rate selection* module determines the optimal data rate according to the applications. Various policies can be defined such as 'maximize the throughput' or 'maximize the total number of clients'.

To see how rate selection module works, we use a simple example. If the policy is 'maximize the total number of clients', 9 Mbps is the optimal data rate, in which all clients could decode the broadcasting data from server. If the policy is 'maximize the total throughput', 48 Mbps is the optimal rate. Although only 5 clients achieve the successful reception, the total throughput is $48 \times 5 = 240$ Mbps, which is larger than selecting 9 or 24 Mbps.



(a) Data Rate (b) Total Rate

Fig. 9. Rate adaptation results.

4 Performance Evaluation

We implement RAB on USRP and build a 20-node testbed to evaluate its performance.

4.1 Implementation

Platform: We build the prototype of RAB using the GNU Radio toolkit and USRP N210. All the USRP nodes are equipped with SBX daughter boards and two VERT2450 antennas, so that they can simultaneously transmit and receive on 2.4 GHz. We adopt existing *IEEE 802.11 a/g/p transceiver for GNU Radio* [2] as the basic WiFi physical layer (PHY).

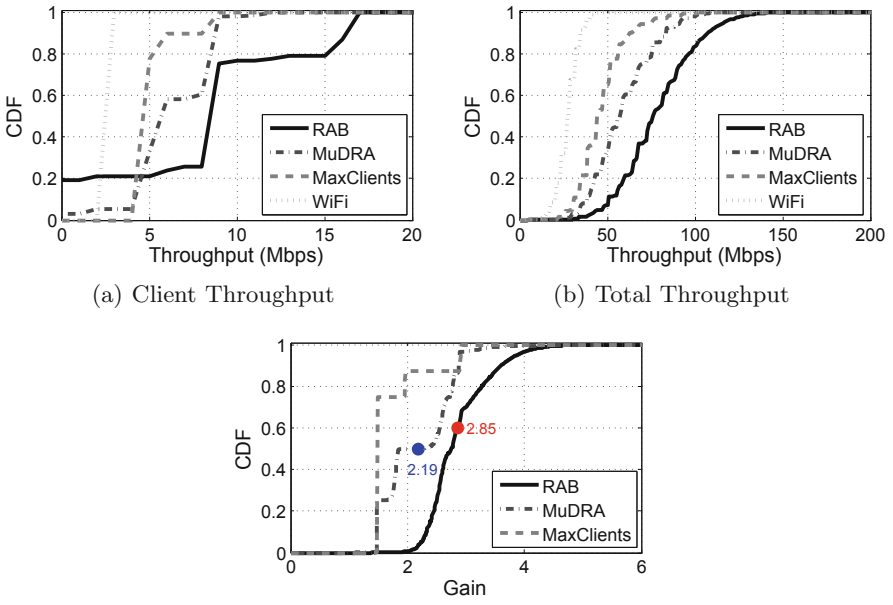
RAB PHY: The design of RAB PHY is shown in Fig. 2. We implement it according to the design. To trigger the rate adaptation, the server sets the *reserved bit* as '1'. The server keeps broadcasting packets while it logs the non-collided preamble for the use of preamble cancellation. Besides, every client estimates the noise level from the received packet and keeps updating the average SNR for each subcarrier in order to calculate the effective SNR.

MAC: Since RAB works at the broadcast mode, the carrier sensing in MAC layer is disabled in our prototype.

SNR: The relationship between the maximal supported rate and the effective SNR is measured using our USRP nodes. Then, every client can empirically determine its maximal supported rate when the effective SNR is obtained.

Testbed: As shown in Fig. 8, our testbed includes 20 USRP nodes as an IoT covering an office, whose area is 32 m^2. One server and 19 clients build a single-hop topology for data broadcasting.

Configuration: All USRP nodes operate at 2.484 GHz. To avoid the odd decimation, we use the bandwidth of 10 M instead of 20 M. Hence, the data rates are halved, *e.g.*, the lowest data rate is 3 Mbps in our evaluation. The size of every packet is 1500 Byte.



(a) Client Throughput

(b) Total Throughput

(c) Gain over WiFi

Fig. 10. Comparison on throughput.

Compared Schemes: We compare RAB with

- MuDRA [3]: is the state-of-the-art rate adaptation design for multicast, which adapts the data rate by lightweight feedbacks from partial clients. MuDRA aims to ensure >85% packet reception ratio (PRR) for >95% clients. The feedback interval is set to be 500 ms as used in [3], and no more than 4 clients will send their feedbacks every 500 ms.
- MaxClient: is a straightforward method that selects the data rate for maximizing the number of clients.
- WiFi: is the standard IEEE 802.11a/g protocol. The server broadcasts with the lowest data rate and the clients do not transmit ACK.

4.2 Evaluation Result

Micro Benchmark Rate Adaptation. The server broadcasts 1500Byte packets to a client with a 50 ms interval using RAB, MuDRA, MaxClient, and WiFi. The aim of such real-time experiments is to validate the rate adaptation in RAB. In the experiments, the USRP nodes are kept stationary, leading to relatively stable effective SNR.

Figure 9 shows the average data rate when the number of clients increases from 1 to 19. We observe that RAB always selects the highest data rate, because

it does not sacrifice the overall throughput for poor clients. To be specific, when the number of clients increases, MaxClients tends to be stuck in the data rate of 4.5 Mbps, which is the lowest data rate for all 19 clients. Meanwhile, since MuDRA is allowed to ignore the worst client, it is slightly better than MaxClients, but still fall behind RAB. In Fig. 10, we adopt the sum of each client's data rates, i.e., the total rate, to indicate the throughput. The high total rate implies that RAB scales well with increasing clients. These results verify the promising advantage of RAB in broadcast.

System-Level Gain. Based on the experiment results, we collect the maximal supported data rates for all the 19 clients for system-level simulation. For comparison, we assume MaxClients utilize the feedbacks without extra overhead, while the standard WiFi directly broadcasts packets using the lowest data rate. Moreover, as [3], each feedback in MuDRA is 1 ms, and we let all the feedback nodes transmit successively.

We iterate all the possible combinations of the 19 clients, and calculate the cumulative distribution of the throughput shown in Fig. 10. From Fig. 10(a), we find that in WiFi, all clients have the same throughput because they all receive packets from the server at the lowest data rate. Besides, both RAB and MaxClients have partial zero throughput. The reason is that the server "abandons" some clients with poor link quality. However, those zero-throughput clients (less than 20%) does not impact the network-wide throughput. As shown in Fig. 10(b), the overall throughput gain of RAB is tremendous, *i.e.*, the median throughput increments compared with MuDRA and MaxClients are as high as 31.03% and 72.73%, respectively.

Specifically, we evaluate the total throughput gains of RAB, MaxClients, and MuDRA, compared with the standard WiFi broadcast in Fig. 10(c). The median gain of RAB and MuDRA are 2.76x and 1.87x respectively. Moreover, RAB can achieve an average throughput gain of 2.85x compared with WiFi, which is also 30.51% higher than that of MuDRA.

5 Related Work

We classify existing rate adaptation techniques into two categories.

Rate Adaptation in Unicast. Adapting the suitable data rate to the dynamic channel is valuable for wireless communication, where a too high data rate leads to decoding error and a too low rate wastes the channel resources. In unicast, the data rate is adjusted by various metrics. RRAA [16] measures the packet loss rate. SoftRate [14] utilizes the SoftPHY hints to obtain the per packet BER. FARA is a frequency awareness rate adaptation [9]. AccuRate [11] investigates the constellation state. CARA analyzes the collisions [5]. These methods perform well in unicast. However, collecting the metrics from all clients is not practical in broadcast due to heavy overhead.

Rate Adaptation in Multicast. To improve the throughput in multicast, extensive low-overhead rate adaptation methods are developed. REMP [8]

requires the non-reception clients to send NACKs. ARSM [13] reduces the overhead by selecting partial clients to send feedbacks, typically the clients with poor channel quality. Peercast [17] improves the link layer multicast through collaborative relays and batch ACKs. MuDRA [3] dynamically adjusts the data rate relying on collecting representative feedbacks via a light-weight protocol. However, overhead of above methods cannot be completely removed.

6 Conclusion

This paper presents the novel RAB to enable the rate adaptation in IoT broadcast. Leveraging the preamble and orthogonal subcarriers, RAB collects parallel feedbacks from all clients during the preamble time, which has no affect on packet reception and requires no extra overhead. Through implementation and testbed based performance evaluation, we show that RAB embraces the parallel acknowledgements, separates feedbacks from preamble signals, and increases the throughput in broadcast.

The future work includes the RAB transplant from WiFi to other IoT wireless protocols such as ZigBee and LoRa, the RAB extension on subcarrier allocation to obtain more accurate estimation, and the RAB generalization from broadcast to multicast.

Acknowledgment. This work was supported in part by the National Key R&D Program of China 2018YFB1004703, National Natural Science Foundation of China grant 61972253, 61672349, 61672353.

References

1. Bharadia, D., McMilin, E., Katti, S.: Full duplex radios. In: ACM SIGCOMM (2013)
2. Bloessl, B., Segata, M., Sommer, C., Dressler, F.: An IEEE 802.11 a/g/p OFDM receiver for GNU radio. In: ACM SIGCOMM Workshop on Software Radio Implementation Forum (2013)
3. Gupta, V., Gutterman, C., Bejerano, Y., Zussman, G.: Experimental evaluation of large scale WiFi multicast rate control. In: IEEE INFOCOM (2016)
4. Halperin, D., Hu, W., Sheth, A., Wetherall, D.: Predictable 802.11 packet delivery from wireless channel measurements. In: ACM SIGCOMM (2011)
5. Kim, J., Kim, S., Choi, S., Qiao, D.: CARA: collision-aware rate adaptation for IEEE 802.11 WLANs. In: IEEE INFOCOM (2006)
6. Kodialam, M., Nandagopal, T.: Fast and reliable estimation schemes in RFID systems. In: ACM MobiCom (2006)
7. Kong, L., et al.: Adasharing: adaptive data sharing in collaborative robots. IEEE Trans. Ind. Electron. **64**(12), 9569–9579 (2017)
8. Lim, W.-S., Kim, D.-W., Suh, Y.-J.: Design of efficient multicast protocol for IEEE 802.11n WLANs and cross-layer optimization for scalable video streaming. IEEE Trans. Mob. Comput. (TMC) **11**(5), 780–792 (2012)
9. Rahul, H., Edalat, F., Katabi, D., Sodini, C.: Frequency-aware rate adaptation and MAC protocols. In: ACM MobiCom (2009)

10. Sen, S., Madabhushi, N.K., Banerjee, S.: Scalable WiFi media delivery through adaptive broadcasts. In: NSDI (2010)
11. Sen, S., Santhapuri, N., Choudhury, R.R., Nelakuditi, S.: AccuRate: constellation based rate estimation in wireless networks. In: NSDI (2010)
12. Shahzad, M., Liu, A.X.: Every bit counts: fast and scalable RFID estimation. In: ACM MobiCom (2012)
13. Villalón, J., Cuenca, P., Orozco-Barbosa, L., Seok, Y., Turletti, T.: Cross-layer architecture for adaptive video multicast streaming over multirate wireless lans. IEEE J. Sel. Areas Commun. (JSAC) **25**(4), 699–711 (2007)
14. Vutukuru, M., Balakrishnan, H., Jamieson, K.: Cross-layer wireless bit rate adaptation. In: ACM SIGCOMM (2009)
15. Wang, Y., He, Y., Mao, X., Liu, Y., Li, X.-Y.: Exploiting constructive interference for scalable flooding in wireless networks. IEEE/ACM Trans. Netw. (ToN) **21**(6), 1880–1889 (2013)
16. Wong, S.H., Yang, H., Lu, S., Bharghavan, V.: Robust rate adaptation for 802.11 wireless networks. In: ACM MobiCom (2006)
17. Xiong, J., Choudhury, R.R.: PeerCast: improving link layer multicast through cooperative relaying. In: IEEE INFOCOM (2011)
18. Zhang, J., Tan, K., Zhao, J., Wu, H., Zhang, Y.: A practical SNR-guided rate adaptation. In: IEEE INFOCOM (2008)
19. Zhou, Z., Gao, C., Xu, C., Zhang, Y., Mumtaz, S., Rodriguez, J.: Social big-data-based content dissemination in internet of vehicles. IEEE Trans. Ind. Inform. **14**(2), 768–777 (2017)