

**© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.**

# Performance Characterization and Receiver Design for Random Temporal Multiple Access in Non-Coordinated Networks

**Abstract:** Random access is a well-known multiple access method for uncoordinated communication nodes. Existing work mainly focuses on optimizing iterative access protocols, assuming that packets are corrupted once they are collided, or that feedback is available and can be exploited. In practice, a packet may still be able to be recovered successfully even when collided with other packets. System design and performance analysis under such a situation, particularly when the details of collision are taken into consideration, are less known. In this paper, we provide a framework for analytically evaluating the actual detection performance in a random temporal multiple access system where nodes can only transmit. Explicit expressions are provided for collision probability and signal to interference and noise ratio (SINR) when different numbers of packets are collided. We then discuss and compare two receiver options for the AP, and provide detailed receiver design for the premium one. In particular, we propose a synchronization scheme which can largely reduce the preamble length. We also demonstrate that system performance could be a convex function of preamble length both analytically and via simulation, as well as the forward error correction (FEC) coding rate.

**Keywords:** random temporal multiple access; non-coordination networks; packet collision

## I. INTRODUCTION

In quite a few wireless communication systems, nodes have very limited receiving functionality or even no receiver at all. Some of these systems are designed in this way to minimize circuit power consumption as they do not have internal power sources, such as passive radio frequency identification device (RFID) tags that use backscatter for signal transmission. Other systems, such as the emerging nano-sensors [1], are mainly used for collecting and forwarding information, and will be simpler to implement if receiving modules are completely removed from the sensor nodes. With the wide application of the Internet-of-Things technology [2]-[5], these transmit-only systems may become more popular. For example, the Sigfox IoT solution based on LoRa modulation improves the reliability of transmission by simply repeating the transmission several times without requiring coordination and acknowledgment from the gateway device. This essentially makes it unnecessary to have the reception function in the IoT node. For such systems without receiving functions, it is almost impossible to coordinate the transmission of different nodes using conventional multiple access schemes such as frequency- or time- division multiple access. Instead, a node is allowed to transmit by randomly picking up a timeslot, a frequency channel or a pseudo-random code.

Random access has been extensively studied [2] - [17]. These works have studied various protocols to minimize the collision probability or maximize the overall throughput. However, to implement these protocols, feedback mechanisms are generally needed between nodes, and hence the receiving capability is required for nodes. In addition, these works typically consider the model that a packet is assumed to be corrupted once a collision, i.e., the simultaneous transmission of more than one packet in the same resource slot, happens.

However, it is still possible to recover a collided packet successfully in real systems. In this situation, the *signal to interference and noise ratio (SINR)* is an equally important parameter to consider, to the *collision probability*. The SINR characterizes how severe the desired signal is affected by the multi-user interference due to collisions, and it is closely related to signal modulation and multiple access schemes. In [9], the collision probability is investigated for massive IoT Networks with random access, which is modeled by the Poisson Point Process. Various random access protocols are analyzed and compared based on this model.

The feasibility of recovering collided packets for random access has been investigated in the literature, largely in the work on “*capture effect*” and “*multi-packet reception*” (MPR) [16] - [20]. The capture effect is referred to that the strongest signals may be correctly decoded despite the collision. The term *capture probability*, denoting the probability of the received SINR larger than a threshold, is often used to characterize the MPR performance. The closed-form expression of the capture probability is typically obtained via assuming a continuous distribution of the received SINR, where the details of the collision are ignored.

In this paper, we consider a simple random temporal multiple access (RTMA) protocol, characterize its performance by considering collision details, and investigate receiver designs enabling the use of short preamble for reduced collision, for completely non-coordinated nodes. The system consists of transmitter- only sensor nodes and an access point (AP) with only single packet reception capability. Different to many RTMA systems, we consider a RTMA network where all nodes use the same pseudo-random (PN) spreading code to encode data and send the PN-coded signal to the AP, which can largely simplify the node design and manufacture. Our major contributions in this paper are four folds.

- Firstly, different from existing work such as those reported in [9], we provide a framework for deriving explicit expressions for collision probability and SINR when an arbitrary number of packets collide.
- Secondly, we investigate and compare two receiver options at the AP, differentiating under the situation when packet collision happens. They are named as *first come, first served (FCFS)* and *Switch to the larger (STTL)* schemes. The STTL scheme is shown to have better performance. The STTL scheme is the typical scheme studied in multi-packet reception literature such as those in [16], [17]. Different from the existing work, we specifically consider the impact of collision on the performance of these two receiver options, and systematically compare the two options.
- Thirdly, we propose a novel and detailed receiver design for STTL, which does not require packet detection and can thus significantly reduce the length of the preamble. This is a brand new design that has never been reported before, to our best knowledge. Reduction of the preamble will not only reduce transmission power but also reduce collision probability and hence improve system performance. Optimization of receiver parameters is investigated both analytically and numerically.
- Finally, we demonstrate the design tradeoff in the length of preamble, length of pseudo-random spreading code, and coding rate in such RTMA systems and show that packet error rate could be a convex function of these parameters.

The rest of this paper is organized as follows. In Section II, after introducing the RTMA protocol, we derive analytical expressions for approximated collision probability and average SINR. In Section III-C, we investigate and compare FCFS and STTL, the two receiver options. In Section IV, we propose a detailed receiver design for STTL and evaluate the design tradeoff of the preamble and FEC coding in RTMA. Section V concludes this paper.

## II. RANDOM TEMPORAL MULTIPLE ACCESS

### A. The Protocol

We consider a communication system where each of the  $M$  sensor nodes is transmitting one packet to the AP over a time window with a period equal to  $N$  symbols. The length of every packet is fixed to be  $L$  symbols. Each packet consists of a preamble of length  $L_p$  and data payload of length  $L_d$ . We will study the impact of direct sequence spread spectrum on the system performance, and hence assume a pseudo random (PN) spreading sequence (SS) is applied to the data payload. The length of the PN sequence is  $L_s$ , and its power is normalized to 1. All nodes use the same preamble and spreading code for simplicity, and different nodes' data is separated via their unique addresses that can be embedded in, e.g., the PHY header. The modulation is assumed to be BPSK.

Under the RTMA protocol, every node randomly chooses a starting time to continuously transmit its packet, which consists of a common preamble of length  $L_p$ , the data payload of length  $L_d$ , and the  $(L - L_p - L_d)$  cyclic redundancy check (CRC) samples. The starting time follows a uniform distribution between 0 and  $N - L + 1$ . Note that such randomness can be achieved by using an internal timer plus event driven. Transmission can be activated with a random delay following a new measurement, without the carrier sensing and random backoff mechanism that is applied in conventional random access systems [12]. A node can repeatedly transmit the same signal a few times to increase the probability of success reception. The required transmission time can be worked out for a given desired success probability according to the probability of successful reception as will be derived later. Acknowledgement of successful reception by the AP receiver, if needed, can also be achieved by using, e.g., powering and sending simple message to a passive RFID circuit attached to the node.

Since the transmission time is not synchronized, there is no a formal definition of timeslot here. We let  $N_c = N/L$  be the number of virtual timeslots, and for convenience, we assume that  $N_c$  is an integer.

### B. Receiver Options

For RTMA, we consider a receiver at AP which only picks up a packet at a time, without invoking complex parallel processing techniques to detect two or more overlapped packets simultaneously. Without loss of generality, let node 0 be the node of interest, and let nodes 1 to  $n$  be those colliding with node 0. The general received signal of node 0, after applying despreading operation, can be represented as

$$y_0 = h_0 g_{0,0} x_0 + \sum_{n'=1}^n h_{n'} g_{0,n'} x_{n'} + z_0, \quad (1)$$

where  $h_{n'}$ ,  $n' \in [1, n]$  denote the channel gains and are complex i.i.d. Gaussian random variables with mean 0 and variance  $\sigma_h^2$ ,  $x_{n'}$  are i.i.d. random discrete variables with equal probability on 1 or -1,  $g_{0,n'}$  is the despreading gain in the correlator output for user  $n'$ -th signal, and  $z_0$  is AWGN with mean zero and variance  $\sigma_z^2$ . At the synchronized point for node 0 in the AP's receiver correlator output,  $g_{0,0} = 1$ . If the  $n'$  node's packet is aligned with node 0, then  $g_{0,n'} = 1$ ; otherwise,  $g_{0,n'}$  shall approach to zero with  $L_s$  increasing. For short code, we assume that  $|g_{0,n'}|^2$  has a mean value of  $1/L_s^2$ .

We study two single-packet receiver options for AP. These two schemes process a packet normally if it is not overlapped with any other packets, but they differ in the actions when packets are overlapped.

1) *First Come, First Served (FCFS)*: For FCFS, the AP starts to process a packet A once it arrives, and it will not look at other packets arriving within the period of packet A. Hence any other packets arriving during this period will be discarded.

2) *Switch to the Larger (STTL)*: For STTL, when packets are overlapped, the AP will always process a packet that has the largest channel magnitude. This means that the AP will shift to a new packet while it is processing an existing one if they are overlapped and the existing packet has lower channel magnitude than the new arrival. A method for implementing such a switching is proposed in Section IV.

### III. PERFORMANCE CHARACTERIZATION

Our focus here is to characterize and evaluate the actual SINR and packet error rate (PER) performance for systems using the protocol, without considering its optimization. The protocol itself can be improved by, e.g., parameter optimization and combining multiple rounds of transmissions, based on the results in this paper.

In this paper, we study the *collision probability*, *processing rate*, *SINR* and packet error rate (PER) for specific number of collided packets, to characterize the performance of the RTMA protocol. The *processing rate* is defined as the ratio between the number of packets, which are being processed by the receiver, and the total packets  $M$ . The processing rate indicates how many packets will be processed by the AP on average, and depends on both the collision probability and the actual receiver scheme.

For analytical characterization, we will apply some model simplification. In the absence of network clock synchronization, transmission can be completely random, and different parts of a packet may be overlapped with different number of other packets. Hence it is very challenging, if not impossible, to derive the analytical expressions for such collision probability and SINR. Instead, we consider a simplified case where transmission is timeslot based and different nodes' timeslots are aligned. This corresponds to the case where each node has its own timer or there is an external trigger to synchronize nodes' timeslots. In this case, the model becomes that  $M$  nodes randomly pick up  $M$  out of  $N_c$  channels. Such an approximation will lead to slightly small collision probability compared to the fully random case. When  $N_c$  is sufficiently large, the difference will be quite small. We will see later that the analytical results from this simplified model are well matched with those simulation results based on the original completely random model.

#### A. Non-Collision Probability

In this subsection, we derive  $\Pr(m)$ , the probability of  $m$  nodes not being collided by any other nodes, using the simplified model of  $M$  nodes sharing  $N_c$  channels. These probabilities can be used to calculate a lower-bound for the processing rate. The probability  $\Pr(m)$  can be computed as follows.

1) Compute  $c_1(m)$ , the number of possible combinations for  $m$  nodes sharing  $N_c$  channels, which is given by

$$c_1(m) = \mathcal{C}_{N_c}^m \mathcal{P}_M^m, \quad (2)$$

where  $\mathcal{P}_M^m = \prod_{k=0}^{m-1} (M - k)$ , and  $\mathcal{C}_{N_c}^m = \mathcal{P}_{N_c}^m / \mathcal{P}_m^m$ ;

2) Compute  $c_2(m)$ , the number of possible combinations for  $(M - m)$  nodes colliding with each other over the remained  $N_c - m$  channels;

3) Obtain the probability  $\Pr(m)$  as

$$\Pr(m) = c_1(m)c_2(m)/N_c^M. \quad (3)$$

Note that  $\Pr(M - 1) = 0$  because the situation that only 1 packet is collided will not happen.

The term  $c_2(m)$  does not have a general and simple expression for different  $m$ , and needs to be computed specifically for every  $m$ . This makes it intractable to obtain exact closed-form performance expressions that involve very large number of nodes. Fortunately, when  $N_c$  is much larger than  $M$ , which is required to achieve acceptable PER performance as to be seen later,  $\Pr(m)$  decreases quickly with  $m$  decreasing. Therefore, we generally only need the exact  $c_2(m)$  expressions for several larger  $m$  values. In particular, we have the following results for  $m = M - 6$  to  $M$  by considering detailed cases of collisions

$$\begin{aligned} c_2(M) &= 1; \\ c_2(M - 2) &= \mathcal{C}_{N_c - M + 2}^1; \\ c_2(M - 3) &= \mathcal{C}_{N_c - M + 3}^1; \\ c_2(M - 4) &= \mathcal{C}_{N_c - M + 4}^1 + \mathcal{C}_{N_c - M + 4}^2 \mathcal{C}_4^2; \\ c_2(M - 5) &= \mathcal{C}_{N_c - M + 5}^1 + \mathcal{C}_{N_c - M + 5}^2 \mathcal{P}_5^2; \\ c_2(M - 6) &= \mathcal{C}_{N_c - M + 6}^1 + 50\mathcal{C}_{N_c - M + 6}^2 + 90\mathcal{C}_{N_c - M + 6}^3. \end{aligned}$$

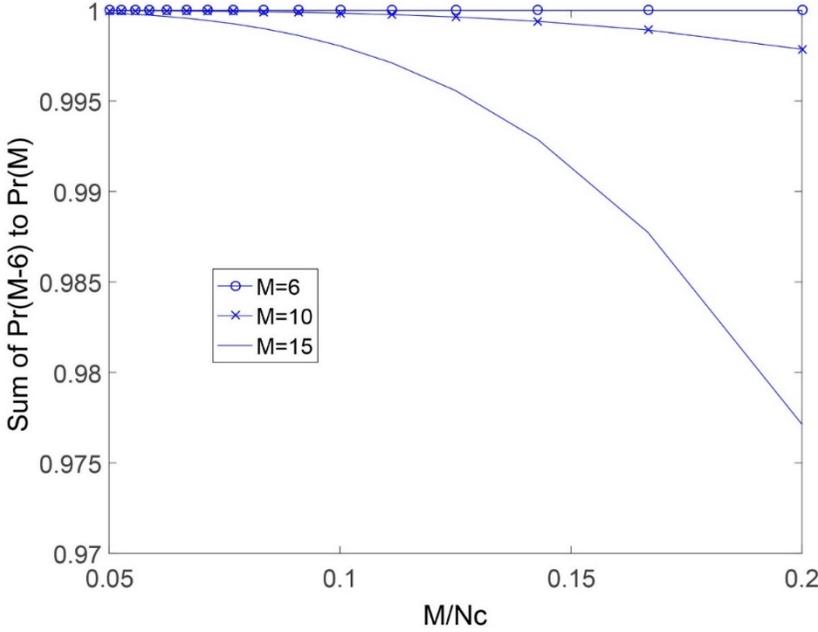
Note that in these equations, we first consider the cases how many nodes collide, and then consider how many combinations of the virtual timeslots in which these collisions could happen. Use  $c_2(M - 5)$  as an example. The first term  $\mathcal{C}_{N_c - M + 5}^1$  accounts for the combinations of timeslots when all 5 nodes collide together. In the second term,  $\mathcal{C}_{N_c - M + 5}^2$  denotes the number of combinations of two timeslots for 2 and 3 nodes colliding within them. There are  $\mathcal{C}_5^2$  combinations for selecting 2 nodes out of 5. Since each 2-node combination could collide in one of the two virtual timeslots, eventually it becomes  $\mathcal{C}_{N_c - M + 5}^2 \mathcal{P}_5^2$ .

Fig.1 demonstrates that  $\sum_{m=M-6}^M \Pr(m)$  approaches 1 for some  $N_c$  and  $M$  values. A close inspection of the values of  $\Pr(m)$  reveals that  $\Pr(m) \gg \Pr(m - 1)$ ,  $m \neq M$  for an even  $m$  as odd number of nodes have less collision combinations. Note that the gap  $1 - \sum_{m=M-6}^M \Pr(m)$  increases with  $M$  and  $M/N_c$ . This indicates that to achieve a higher accuracy for collision probability analysis, more terms need to be evaluated for larger  $M/N_c$  and  $M$ . Hence the analytical results for the probability and then the SINR in this paper could become inaccurate for massive IoT networks where  $M$  can be as large as one thousand [7][8][9]. Nevertheless, the other work in this paper shall still be effective, only if the ratio  $M/N_c$  is sufficiently small.

Given the results above, we can approximate the average number of non-collided packets that arrive at the AP as

$$M_s = \sum_{m=M-6}^M m \Pr(m) + (1 - \sum_{m=M-6}^M \Pr(m))(M - 6)/2 \quad (4)$$

where the second term approximates the impact when more than  $M - 6$  nodes collide. The value  $M_s/M$  serves as the lower bound for the processing rate.



**Fig. 1.** Illustration of  $\sum_{m=M-6}^M \Pr(m)$  approaching 1 for various  $N_c$  and  $M$  values.

### B. Average SINR

For a particular node, we can work out the mean SINR considering all cases when node 0 is collided by different number of packets. Using the model presented in Section III-C, the theoretical mean SINR for node 0 can be represented as

$$\gamma = \sum_{n=0}^{M-1} f(n)E[\gamma(n)],$$

where  $f(n)$  is the probability of  $n$  nodes colliding with node 0,  $\gamma(n)$  is the instantaneous SINR when such collision happens,  $E[x]$  denotes the expectation of  $x$ . Note that when  $n = 0$ , there is no collision and  $\gamma(0)$  becomes the SNR.

The probability  $f(n)$  can be computed as

$$f(n) = \frac{C_{M-1}^n (N_c - 1)^{M-n-1}}{N_c^{M-1}}$$

$$\approx \frac{C_{M-1}^n}{(N_c - 1)^n}, \text{ when } N_c \text{ is large}$$

(6)

In (6),  $C_{M-1}^n$  represents the combinations of selecting any  $n$  out of  $(M-1)$  nodes that collide with node 0 at a specific timeslot, and  $(N_c - 1)^{M-n-1}$  represents the combinations of the rest  $M - n - 1$  nodes occupying the rest  $N_c - 1$  timeslots, and  $N_c^{M-1}$  is the total combinations of  $M - 1$  nodes occupying  $N_c$  timeslots.

The term  $E[\gamma(n)]$  can be computed according to the distribution of signal and interference links. Since different nodes use the same spreading code for preamble and data payload, we need to differentiate between the cases when packets from different nodes are fully aligned or not. The interference will be quite different

between the two cases. Based on the length of the direct sequence, the probability of the  $n'$ -th node aligning and non-aligning with node 0 can be approximated as  $1/L_s$  and  $(L_s - 1)/L_s$ . Therefore we can obtain

$$\begin{aligned} E[\gamma(n)] &= \frac{\sigma_h^2}{\sum_{n'=1}^n \left( \frac{\sigma_h^2}{L_s} + \frac{(L_s - 1)\sigma_h^2}{L_s^3} \right) + \sigma_z^2} \\ &= \frac{L_s^3 \sigma_h^2}{n(L_s^2 \sigma_h^2 + (L_s - 1)\sigma_h^2) + L_s^3 \sigma_z^2}. \end{aligned}$$

By applying (6) and (7) to (5), the mean SINR  $\gamma$  can be calculated as

$$\begin{aligned} \gamma &= \sum_{n=0}^{M-1} \frac{C_{M-1}^n (N_c - 1)^{M-n-1} L_s^3 \sigma_h^2}{N_c^{M-1} (n(L_s^2 \sigma_h^2 + (L_s - 1)\sigma_h^2) + L_s^3 \sigma_z^2)} \\ &\approx \sum_{n=0}^{M-1} \frac{C_{M-1}^n \sigma_h^2}{(N_c - 1)^n \left( n(L_s^{-1} \sigma_h^2 + L_s^{-3} (L_s - 1)) \sigma_h^2 \right) + \sigma_z^2}. \end{aligned}$$

When the length of the transmission window  $N$  is fixed, the product  $N_c L_s$  can be regarded as fixed. In this case, the longer direct sequence will cause smaller  $N_c$  and more collision, but it can lead to higher spreading gain. The combined impact on the averaged SINR  $\gamma$  can be assessed below.

Let  $\lambda(n)$  be the denominator in (8), and let  $a = L_s N_c$  be a constant. Then we can rewrite  $\lambda$  as

$$\lambda(n) \approx (N_c - 1)^{n+1} n \sigma_h^2 / a + (N_c - 1)^{n+2} \sigma_h^2 / a^2 + (N_c - 1)^n \sigma_z^2. \quad (9)$$

Equation (9) suggests that for any  $n \geq 0$ ,  $\lambda(n)$  is always a monotonically increasing function of  $N_c$ .

The mean SINR is closely linked to the BER and PER. It can also be used to guide the preamble and coding design, as will be shown later.

### C. Performance of Receivers

We define the *processing rate* as the ratio between the number of packets, which are being processed by the receiver without stopping to move to another packet during the processing, and the total packets  $M$ . The processing rate indicates how many packets will be processed by the AP on average, and depends on both the collision probability and the receiver options. Together with the mean SINR, they determine the overall throughput performance of the proposed RTMA scheme and systems.

1) *First Come, First Served (FCFS)*: For FCFS, the AP starts to process a packet A once it arrives, and it will not look at other packets arriving within the period of packet A. Hence any other packets arriving during this period will be discarded.

The mean SINR for FCFS  $\gamma_{\text{FCFS}}$  can be approximated by  $\gamma$  in (8) as the distribution of the processed signals is unchanged.

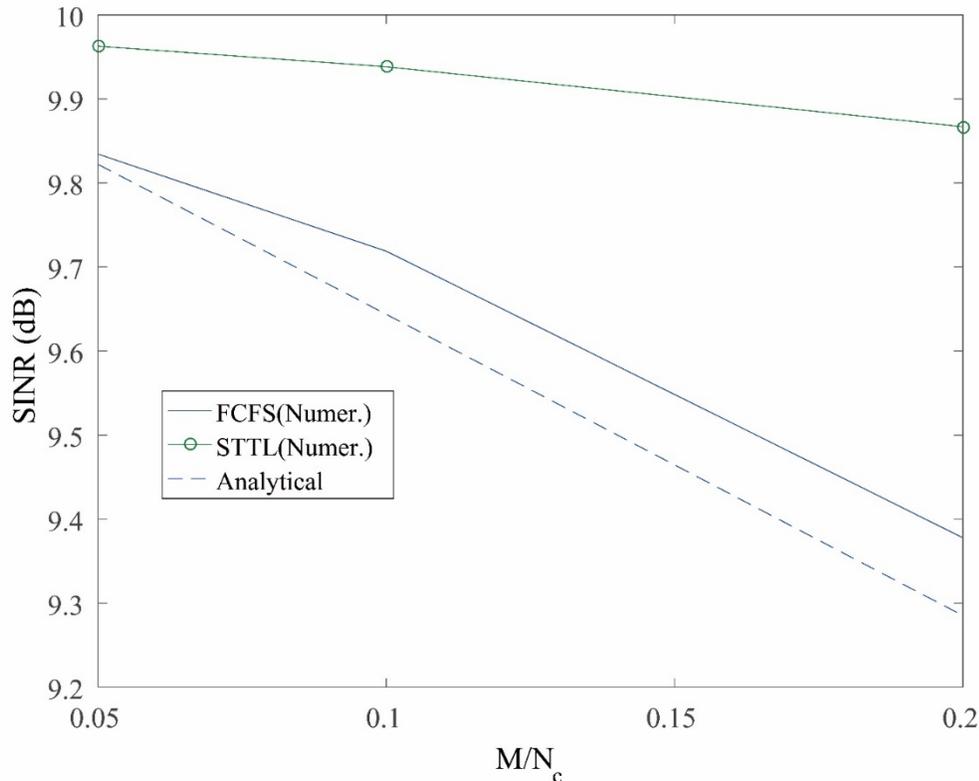
The processing rate is larger than the non-collision ratio  $M_s/M$ , because one packet will still be processed among multiple collided packets.

2) *Switch to the Larger (STTL)*: For STTL, when packets are overlapped, the AP will always process a packet that has the largest channel magnitude. This means that the AP will shift to a new packet when it is processing a first one, if the two packets are overlapped and the first packet has a lower channel magnitude

than the new arrival.

Apparently, this will lead to a lower processing rate compared to FCFS. In the worst case, only one packet will be processed if all packets are mutually overlapped, even if they span a much longer period than the packet period  $L$ .

However, STTL assures a higher mean SINR than  $\gamma$  in (8), which can lead to better BER.



**Fig. 2.** Measured SINR in the simulation and analytical SINR from (8) for different ratios of  $M/N_c$ , where  $M = 10$  and  $1/\sigma_z^2 = 10\text{dB}$ .

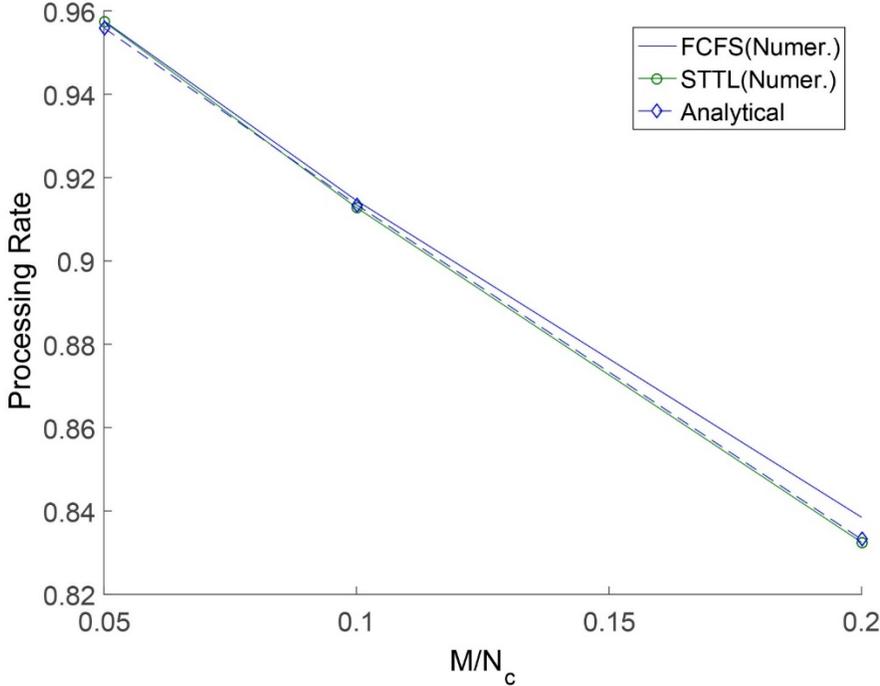
3) *Comparison of the Two Options:* Comparing the SINR of the two receiver options, we can see that  $\gamma_{\text{STTL}} > \gamma_{\text{FCFS}}$  and FCFS has larger processing rate than STTL. Such two analytical relationships cannot tell us directly which option is better. Hence in this subsection, we refer to some simulation results to show that 1) the analytical processing rate and mean SINR results match the simulation results well, and 2) in terms of the overall throughput performance, STTL is better than FCFS. In these simulations, the arrival time of all the packets are assumed to be perfectly known, the modulation is BPSK, and  $L_p = 15, L_d = 33$ .

Fig. 2 shows how the numerical SINR for FCFS and STTL, and the analytical SINR from (8) vary with  $M/N_c$ . The simulation is conducted for a system with 10 nodes at an original SNR of 10dB. The simulated channel is Gaussian distributed with mean zero and variance 1. The figure validates the intuitive conclusion that STTL achieves better SINR than FCFS. It also indicates that the analytical SINR is a good approximation to the actual SINR for FCFS.

Fig. 3 demonstrates the processing rates for FCFS and STTL, as well as the analytical non-collision probability given by  $M_s/M$ , where  $M_s$  is calculated from (4). The figure shows that the analytical result

provides a good approximation for the processing rates of both FCFS and STTL.

Finally, Fig. 4 plots the PER performance for these two receivers. STTL consistently shows better PER than FCFS, which indicates that switching to a larger signal improves the overall detection performance although it may miss more packets than FCFS.



**Fig. 3.** Measured processing rates in the simulation and analytical  $M_S/M$  for different  $M/N_c$ , where  $M = 10$ .

#### IV. DETAILED RECEIVER DESIGN FOR STTL

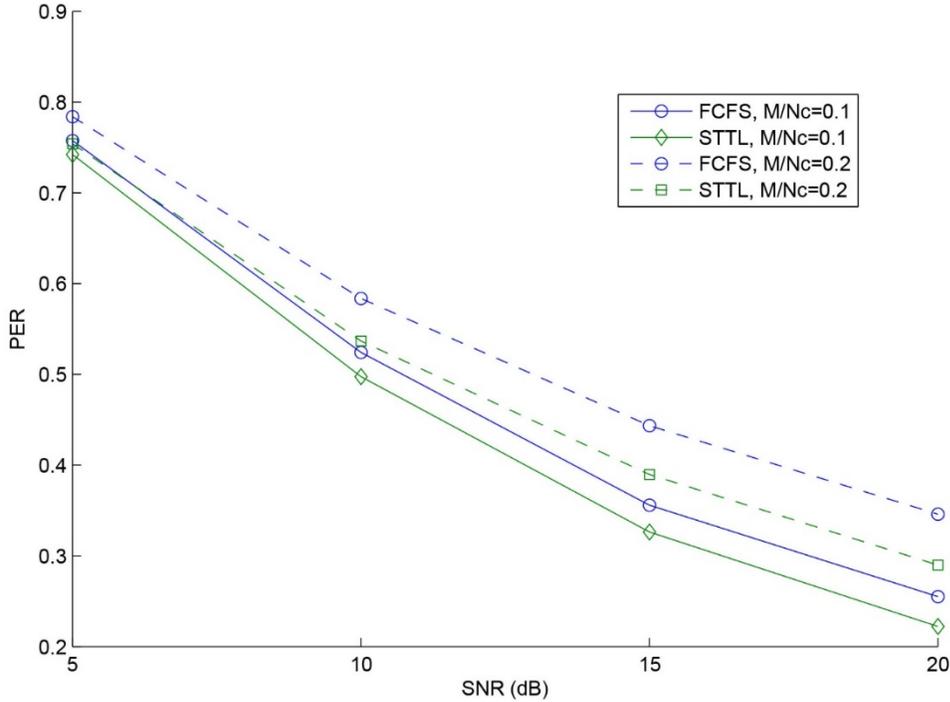
For RTMA, the probability of collision and hence the system performance are closely related to the packet length. This has a direct impact on the preamble design and the choice of modulation and coding schemes. For example, in conventional design, to achieve better packet synchronization at the receiver of AP, the longer preamble is preferred; and lower coding rate could also lead to better coding gain and system performance. However, in RTMA, longer packets can cause more collision, and hence such designs may lead to worse performance. We hence propose a special receiver which allows the use of short preambles and also investigate such tradeoffs in Section IV.

We study detailed receiver design for STTL and propose a packet structure suitable for its implementation in this section.

In a typical receiver, the synchronization module includes both packet detection and fine timing functions. Conventional receivers need to know whether a packet is arriving before invoking the fine timing module. Packet detection typically uses energy or autocorrelation detectors, such as the one proposed in [21]. To achieve good autocorrelation performance, the preamble is required to be long. Unfortunately, for the RTMA protocol, longer preambles can significantly increase the collision probability. The auto-correlation detector is also very likely to fail for overlapped packets.

For STTL, we propose a receiver where such packet detection module can be avoided. Fig. 5 shows the

detailed block diagram for this proposed receiver. The blocks outlined in black show the main components of the receiver, similar to a conventional receiver, except for the lack of a packet detection module. The receiver uses the fine timing module and the auxiliary modules plotted in blue to control the output of a decoded packet, which consists of both data payload and CRC samples. The CRC check function (not shown in the figure) in the MAC layer will then determine whether this is a correctly recovered packet.



**Fig. 4.** PER for FCFS and STTL versus SNR ( $1/\sigma_z^2$ ), where  $M = 10$ .

The fine timing module is based on cross-correlating the received signal with the template signal. The absolute value of the cross-correlation output at time  $m$ ,  $|r_m|$ , is compared with a stored value  $r_{\max}$ . If  $|r_m| > r_{\max}$ , it means this is possibly the start of a new packet with larger signal power. Then the current  $r_{\max}$  is updated with  $c|r_m|$ , where  $c$  is a scalar slightly larger than 1, e.g.,  $c = 1.2$ , to prevent frequent switching due to small fluctuation. At the same time, the pointer of the output buffer of length  $(L - L_p)$  is reset to 1. The channel estimation module uses the value of  $r_m$  for channel estimation, and the equalization module starts to work on the next samples using the estimated channel. Alternatively, if  $|r_m| \leq r_{\max}$ , the receiver will assume the new sample is part of the current packet and continue the processing of the current packet. It will also increase the buffer pointer by 1, such that the decoded bit for the current sample will be written to the next position in the buffer. Once the buffer is full, the processed packet will be exported to MAC layer for CRC check, and the receiver will reset  $p = 1$  and  $r_{\max} = 0$ . This processing flow well matches the STTL principal.

Such a scheme allows the preamble design to be focused on the cross-correlation property, which typically leads to much lower missed and false detection probabilities, compared to packet detection.

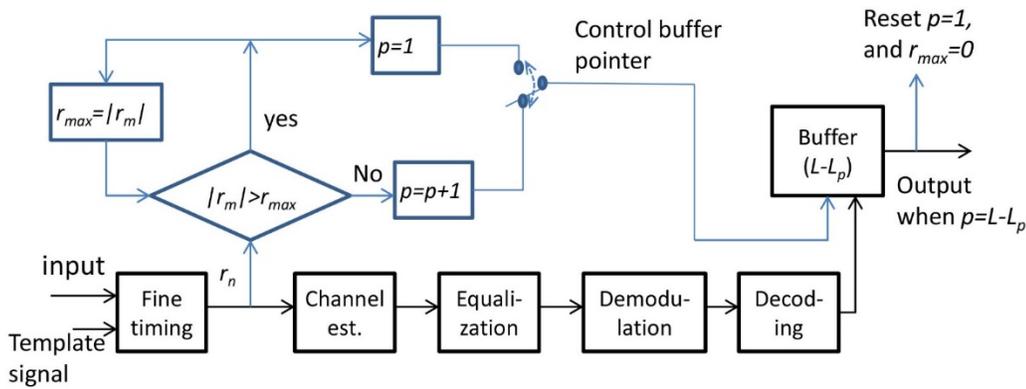


Fig. 5. Receiver block diagram for STTL.

### A. Preamble Length

Extending the signal model in (1) to represent a column vector of  $L_p$  received samples, we get

$$\mathbf{y} = h_0 \mathbf{x}_0 + \sum_{n'=1}^n h_{n'} \mathbf{x}_{n'} + \mathbf{z}_0.$$

Given the locally known template signal  $\mathbf{x}_t = \mathbf{x}_p$ , where  $\mathbf{x}_p$  denotes the preamble, the  $m$ -th output of cross-correlator, as a function of  $n$ , is given by

$$r_m(n) = \mathbf{x}_p^H \mathbf{y}, \quad (11)$$

where the superscript  $H$  denotes conjugate transpose.

When  $\mathbf{y}$  corresponds to the received preamble,  $r_m(n)$  becomes

$$\rho(n) \triangleq h_0 L_p + i_m(n) + \mathbf{x}_p^H \mathbf{z}_0, \quad (12)$$

where  $i_m(n) = \sum_{n'=1}^n h_{n'} \mathbf{x}_p^H \mathbf{x}_{n'}$ . The random variable  $\rho(n)$  can be regarded as the sum of an effective signal with mean zero and variance  $\sigma_h^2 L_p^2$  and a perturbation term with mean zero and variance  $L_p (n \sigma_h^2 + \sigma_z^2)$ .

When  $\mathbf{x}_t$  is not aligned with the preamble, the correlation output  $\eta(n)$  can be regarded as a random variable with mean zero and variance  $L_p ((n+1) \sigma_h^2 + \sigma_z^2)$ .

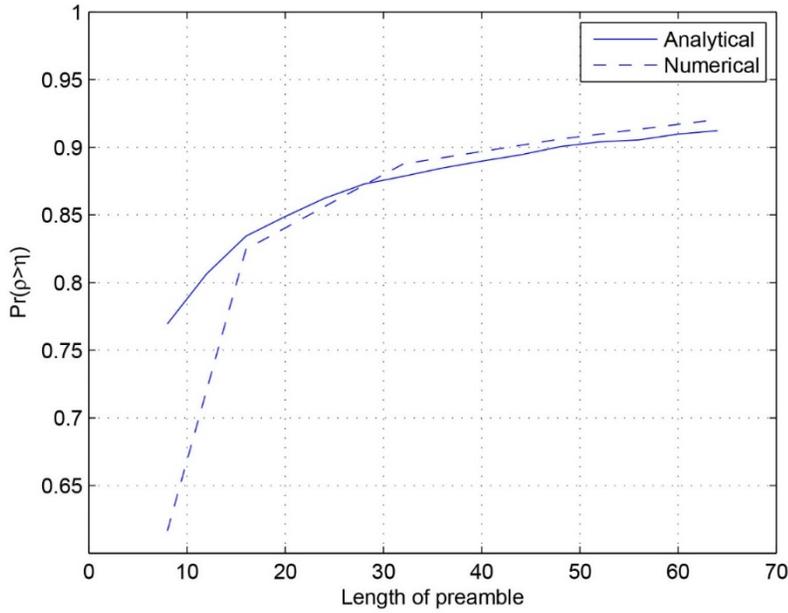
To achieve good timing performance,  $|\rho(n)|$  needs to be larger than  $|\eta(n)|$  with a high probability. The average probability of  $|\rho(n)| > |\eta(n)|$  can be represented as

$$\Pr(\rho > \eta) = \sum_{n=0}^{M-1} f(n) \Pr(|\rho(n)| > |\eta(n)|).$$

From (13), we can see that longer preamble may not always lead to larger  $\Pr(\rho > \eta)$ . This is because  $\Pr(|\rho(n)| > |\eta(n)|)$  and  $f(n)$  change in the reverse direction with respect to  $L_p$ . At the same time, longer preamble causes more collision to data payload. Hence longer preamble may not lead to lower PER, either.

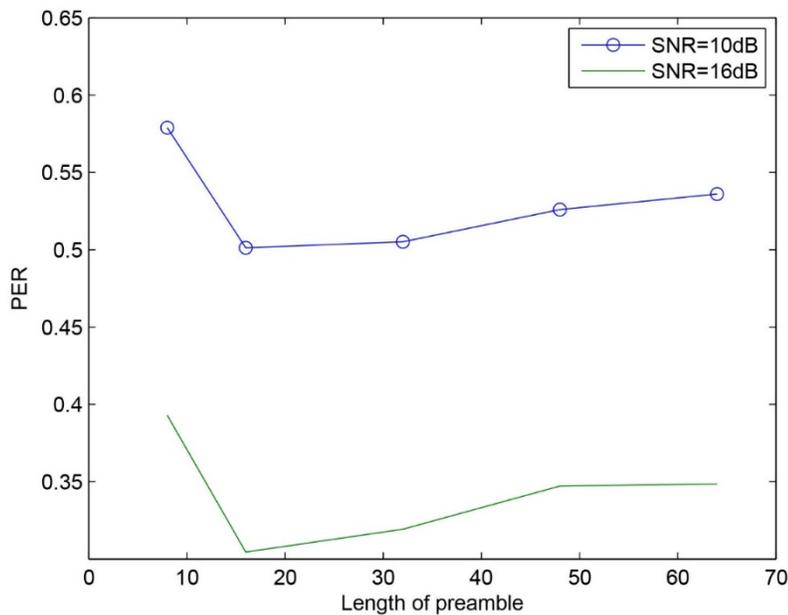
Assume that both  $\rho(n)$  and  $\eta(n)$  are Gaussian distributed. In Fig. 6, we plot the analytical results from (13) and compare it with simulation results. The length of the transmission window is fixed as 4800 samples throughout this paper, unless stated otherwise. The figure shows a good match between the analytical and

simulation results when the preamble is longer than 16. Hence the analytical result can be used for determining the preamble length for a given probability of successful detection. Inaccuracy at shorter preamble is likely due to the inaccuracy of Gaussian assumption of  $\rho(n)$  and  $\eta(n)$  in this case.



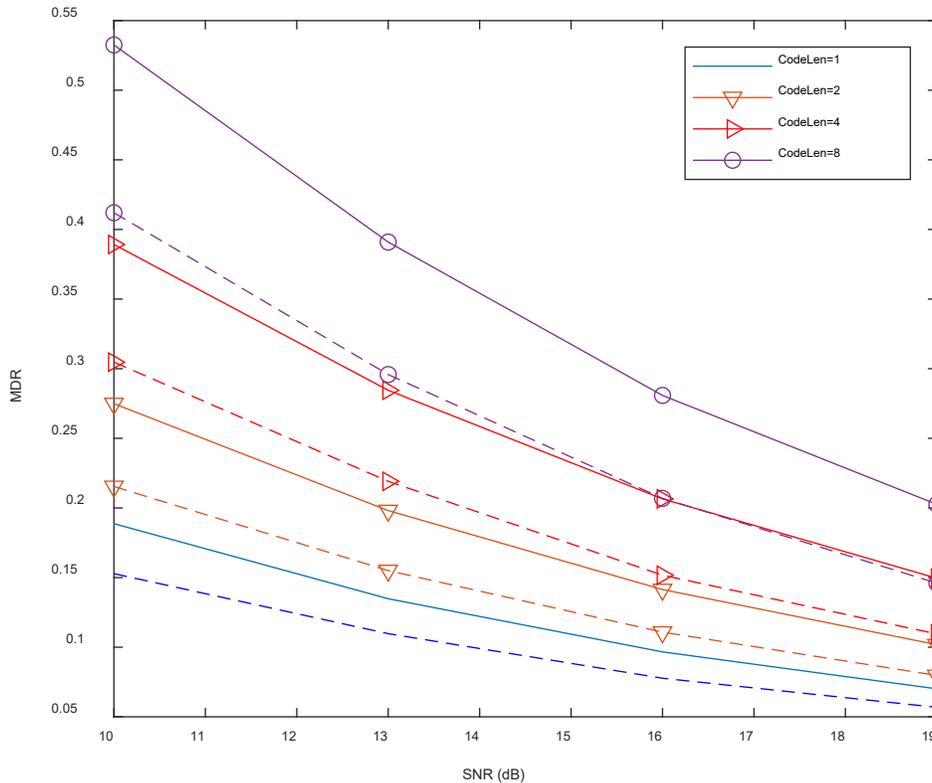
**Fig. 6.** The length of preamble versus  $\Pr(\rho > \eta)$  for fixed parameters  $N = 4800$ ,  $L_d = 32$  and  $\text{SNR} = 10\text{dB}$ .

Fig. 7 demonstrates how PER is affected by the different length of preambles. It can be seen that there exists an optimal preamble length for a given  $N$  and  $L_d$ .



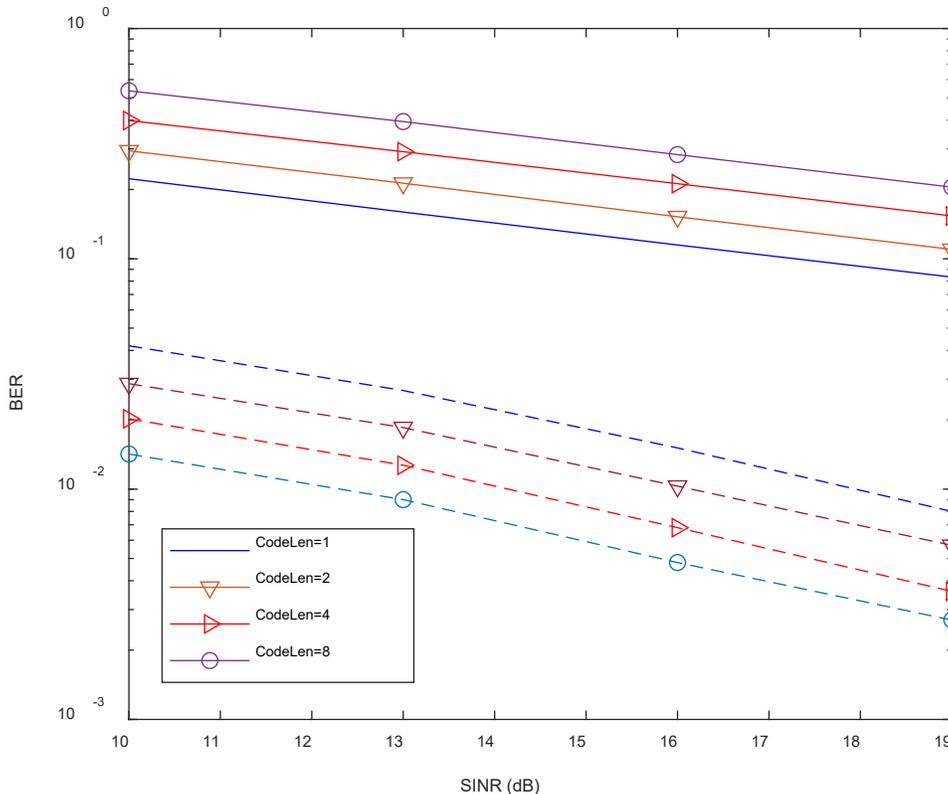
**Fig. 7.** The length of preamble versus PER for fixed parameters  $N = 4800$  and  $L_d = 32$ .

Fig. 8 presents the missed detection rate (MDR) for different PN code length (CodeLen) and different preamble lengths at different SNRs. The MDR is defined as the ratio between the missed packets by the proposed cross-correlation based processing and the total packets. The values of  $M/N_c$  for CodeLen=1, 2, 4, 8 are approximately 0.07, 0.12, 0.21 and 0.4078. With preamble length increasing from 16 to 24, the MDR is reduced since the increased gain is larger than the multiuser interference due to increased collision. With the length of the PN code increasing, more collisions happen, and hence the MDR increases too. It can also be observed that to have a practically working system with a low MDR, using smaller  $M/N_c$ , larger SNR or longer preamble could both could all be potential solutions.



**Fig. 8** Variation of missed detection rate versus SNR for different length of PN codes, Preamble length is 16 for solid curves and 24 for dashed curves.  $N = 9200$  and  $L_d = 44$ .

In Fig. 9, we plot the BER separately for the detected packets and for the total transmitted packets assuming that the detection of all received bits in a packet is wrong if it is not detected. PN spreading codes with different lengths of 1, 2, 4 and 8 are tested. The time window is fixed to be  $N=9200$ . As expected, longer PN codes lead to lower BER for successfully detected packets thanks to improved multiuser suppression capability. However, longer packets cause more collision, leading to higher missed detection ratio, as being demonstrated in Fig. 8. Hence, shorter spreading codes lead to lower overall BER when considering the bit errors due to missed packets.

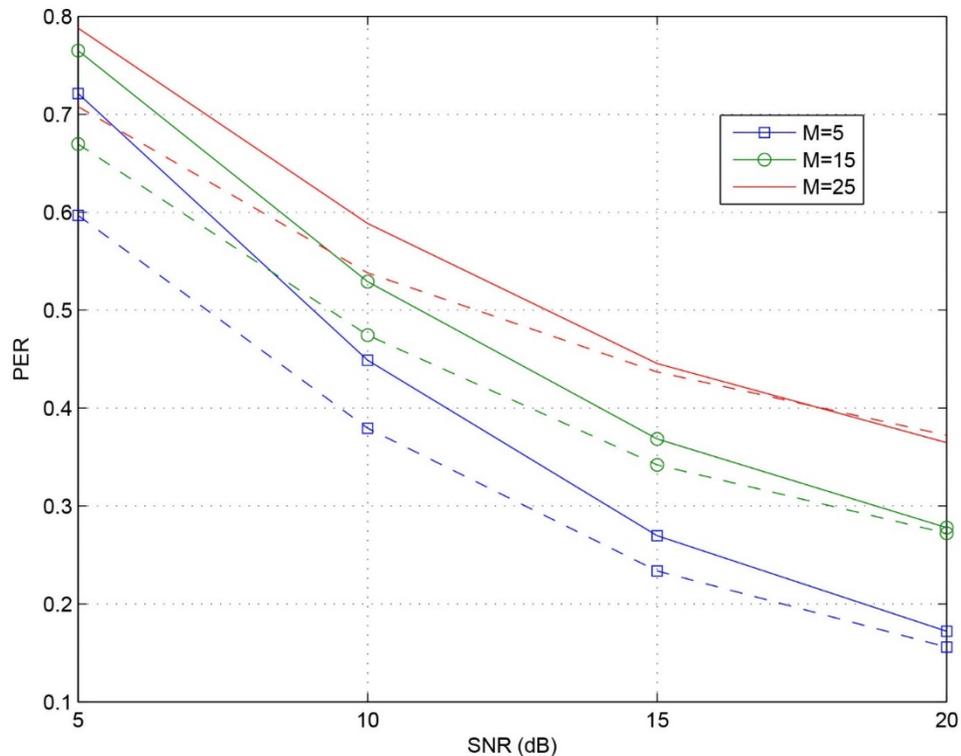


**Fig. 9** BER for detected packets (dashed curves) and total transmitted packets (solid curves). Spreading codes with different lengths are used for encoding the information bits. Length of preamble is fixed to 16,  $N = 9200$ ,  $M = 10$ , and  $L_d = 44$ .

### B. FEC Coding and its Impact

Similar to that the PER is a convex function of the preamble length, PER can also be a convex function of the FEC coding rate, and lower coding rate does not necessarily lead to better performance for a fixed  $N$  and information rate. Codes with lower rate have better error correction probably, but they also lead to more coded bits and hence increase collision. Such effects can also be analyzed using the SINR formulation in Section III-B. Here we only show some simulation results to demonstrate the effects.

In Fig. 10, the PER obtained for coded and uncoded systems are plotted. An 11/15 BCH coding scheme is applied. The figure shows that with increasing number of nodes, the gap between coded and uncoded systems is reduced because the coding gain is counteracted by increased multi-user interference due to the collision. Coding gain is more prominent at lower SNR.



**Fig. 10.** PER for coded (dashed curves) and uncoded (solid curves) systems with fixed parameters  $N = 4800$  and  $L_d = 32$ .

## V. CONCLUSIONS

We have provided a framework for analysing the collision probability and SINR when different number of packets collide in a random temporal multiple access system. We show that the provided analytical results are well matched with simulation results. Two receivers, applying either the first come, first serve or keeping the larger principle, are investigated. It is shown that the latter, which switches to process a new packet with larger power when packet collision happens, always achieves better PER performance. Detailed designs for this better receiver option are investigated. Optimization of the preamble is studied both analytically and by simulation. It is shown that the PER is a convex function of the preamble length for a fixed access window, and so is the FEC coding rate.

## REFERENCES

- [1]. I. F. Akyildiz and J. M. Jornet, "Electromagnetic wireless nanosensor networks," *Nano Communication Networks*, vol. 1, no. 1, pp. 3–19, 2010.
- [2]. Yang Yang, Guannan Song, Wuxiong Zhang, Xiaohu Ge, "Neighbor-Aware Multiple Access Protocol for 5G mMTC Applications," *China Communications*, vol. 13, no. 2z, pp. 80-88, 2016.
- [3]. Yali Wu, Guixia Kang, Ningbo Zhang, "Random Access and Resource Allocation for the Coexistence of NOMA-Based and OMA-Based M2M Communications," *China Communications*, vol. 14, no. 6, pp. 43–53, 2017.

- [4]. W. Zhan and L. Dai, "Massive Random Access of Machine-to-Machine Communications in LTE Networks: Modeling and Throughput Optimization," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2771-2785, April 2018.
- [5]. L. Li, X. Wen, Z. Lu, Q. Pan and W. Jing, "Pre-Backoff Based Random Access with Priority for 5G Machine-Type Communication," *2017 IEEE Globecom Workshops (GC Wkshps)*, Singapore, 2017, pp. 1-6.
- [6]. B. Han and H. D. Schotten, "Grouping-Based Random Access Collision Control for Massive Machine-Type Communication," *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, Singapore, 2017, pp. 1-7.
- [7]. L. Liu, and W. Yu, "Massive Connectivity with Massive MIMO-Part I: Device Activity Detection and Channel Estimation," Jun. 2017, [Online] Available: <https://arxiv.org/abs/1706.06438v1>.
- [8]. R. G. Cheng, Z. Becvar and P. H. Yang, "Modeling of Distributed Queueing-Based Random Access for Machine Type Communications in Mobile Networks," in *IEEE Communications Letters*, vol. 22, no. 1, pp. 129-132, Jan. 2018.
- [9]. N. Jiang, Y. Deng, A. Nallanathan, X. Kang and T. Q. S. Quek, "Analyzing Random Access Collisions in Massive IoT Networks," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 10, pp. 6853-6870, Oct. 2018.
- [10]. G. Fayolle, P. Flajolet, M. Hofri, and P. Jacquet, "Analysis of a stack algorithm for random multiple-access communication," *Information Theory, IEEE Transactions on*, vol. 31, no. 2, pp. 244-254, 1985.
- [11]. B. Tsybakov, "Survey of USSR contributions to random multiple-access communications," *Information Theory, IEEE Transactions on*, vol. 31, no. 2, pp. 143-165, 1985.
- [12]. G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 3, pp. 535-547, March 2000.
- [13]. C. H. Wei, G. Bianchi, and R. G. Cheng, "Modeling and analysis of random access channels with bursty arrivals in OFDMA wireless networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 1940-1953, April 2015.
- [14]. J. Luo and A. Ephremides, "A New Approach to Random Access: Reliable Communication and Reliable Collision Detection," in *IEEE Transactions on Information Theory*, vol. 58, no. 2, pp. 989-1002, Feb. 2012.
- [15]. L. Zhen, H. Qin, B. Song, R. Ding, X. Du and M. Guizani, "Random Access Preamble Design and Detection for Mobile Satellite Communication Systems," in *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 280-291, Feb. 2018.
- [16]. N. Mahravari, "Random-access communication with multiple reception," *IEEE Transactions on Information Theory*, vol. 36, no. 3, pp. 614-622, May 1990.
- [17]. L. Tong, Q. Zhao, and G. Mergen, "Multipacket reception in random access wireless networks: from signal processing to optimal medium access control," *Communications Magazine, IEEE*, vol. 39, no. 11, pp. 108-112, 2001.
- [18]. Q. Zhao and L. Tong, "A multi-queue service room MAC protocol for wireless networks with multipacket reception," *Networking, IEEE/ACM Transactions on*, vol. 11, no. 1, pp. 125-137, 2003.
- [19]. S. Ghez, S. Verdu', and S. Schwartz, "Stability properties of slotted aloha with multipacket reception capability," *Automatic Control, IEEE Transactions on*, vol. 33, n9. 7, pp. 640-649, Jul 1988.
- [20]. A. Zanella and M. Zorzi, "Theoretical analysis of the capture probability in wireless systems with multiple packet reception capabilities," *Communications, IEEE Transactions on*, vol. 60, no. 4, pp. 1058-1071, April 2012.
- [21]. T. M. Schmidl and D. C. Cox, "Robust frequency and timing synchronization for OFDM," *IEEE Trans. Commun.*, vol. 45(12), pp. 1613-1621, Dec. 1997.