This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE Globecom 2011 proceedings.

A Lightweight Deterministic MAC Protocol Using Low Cross-Correlation Sequences

Dongho Kim Danesh J. Esteki Yih-Chun Hu P. R. Kumar
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign
{dkim99,esteki1,yihchun,prkumar}@illinois.edu

***Abstract*—In traditional wireless networks, two nodes cannot simultaneously transmit their packets to each other with one radio device (having no machinery enabling full duplex). To ensure bidirectional communications, we need a medium access protocol to coordinate neighboring nodes' transmissions. Two conflicting design goals for the medium access protocol are the ease of implementation and performance guarantees. We propose a novel medium access protocol which is easily implementable (not requiring clock synchronization) and guarantees performance (the fraction of available slots). Our basic idea is to exploit a set of binary sequences having provably low cross-correlation. Each node has its own code sequence and determines whether to transmit or receive a packet by sequentially examining each bit of the code sequence. As an example, we consider the application of Gold code sequences and theoretically analyze the the fraction of available slots that a Gold-code-based MAC can provide. Our simulation verifies our analysis and shows that a Gold-code-based MAC guarantees the fraction of available slots even on a short time scale.**

I. Introduction

Since multiple nodes share a common medium in wireless networks, a medium access protocol is necessary to avoid collision resulting from simultaneous medium access. There are two kinds of collision in wireless networks [1]: *primary conflict* and *secondary conflict*. Two neighboring nodes may not be able to simultaneously transmit their packets to other when there is only one radio device on each node. We call this collision primary conflict. Secondary conflict occurs when two nodes transmit their packets to a common receiver node through a shared wireless resource. If the packet is sent with insufficient coding, a node receiving two packets simultaneously will be unable to decode either packet.

Secondary conflicts can be solved using orthogonal signals. For example, CDMA [2]–[4] can be deployed for the simultaneous transmission of multiple nodes with power control [5] or with multiuser detection [6]. However, the signal-to-noise ratio in primary conflicts is too high to be solved by such orthogonal codes. In this paper, we examine how scheduling can help address primary conflicts. There are two kinds of methods to deal with primary conflict: *contention-based scheduling* and *time-slotted scheduling*. Contention-based scheduling relies on a contention-resolving technique such as carrier sense multiple access (CSMA) that is widely used in IEEE 802.11 networks. The main advantage of contention-based scheduling is ease of implementation; in particular, nodes need not be time synchronized. However, the non-deterministic nature of contention-resolving techniques makes it difficult to apply this technique to a situation requiring guaranteed performance (e.g. monitoring natural disasters). In contrast, time-slotted scheduling provides a performance guarantees such as minimum delay or minimum throughput. However, the benefit comes at the expense of additional overhead since nodes in the network need to implement time synchronization in order to share time-slot assignment information.

Hence, when we design a protocol to solve the primary conflict problem, there is trade-off between easy implementation and guaranteed performance. In this paper, we try to narrow the gap between the extreme points on the design space. We present a novel medium access protocol that provides guaranteed performance while retaining ease of implementation; in particular, our approach guarantees the fraction of available slots even on a short time scale without requiring either exchange of coordination information or time synchronization. We define a slot to be available for two nodes if one of those two nodes is transmitting and the other is receiving. Our main approach is to exploit binary sequences having provably low cross-correlation [7], [8]. The provably low cross-correlation between two binary sequences implies that the number of bit positions having different values is provably low. For example, when we compare each bit of two Gold code sequences [7], the number of cases where the bit values of the two sequences at the same position are '1' and '0', or '0' and '1' is proven to be low. In our protocol, each node maintains its own code sequence and maps its time slot to each bit of its code sequence. During a time slot corresponding to a '1' bit, the node transmits, otherwise the node receives. Since the only information a node needs is its own code sequence, nodes in our protocol do not need to exchange any information for coordination. Furthermore, the Gold code's property of low cross-correlation guarantees a time-bound before which a slot will be available.

We summarize our contributions as below.

- **Protocol:** We propose a lightweight Gold-code-based MAC protocol. By lightweight protocol, we mean that our protocol does not require coordination or time synchronization.
- **Analysis:** We analyze the property of our Gold-code-based MAC. We prove that our MAC protocol guarantees the fraction of available slots.

- **Simulation:** We evaluate the performance of our protocol by simulation. Our simulation shows that Gold-code-based MAC guarantees the fraction of available slots in various environments.

The rest of our paper is organized as follows. We introduce the property of Gold code as background in Section II. We present and analyze our protocol in Section III. In Section IV, we evaluate by simulation the performance of our protocol. We summarize related work in Section V. Section VI concludes our paper.

II. Background

In this section, we explain the basic concept of Gold code, which we use as an example for explaining our medium access protocol.

A. Gold Code

We do not present all details of theory behind Gold code; rather, we introduce important properties of Gold code for our application. A set of Gold codes is a set of binary sequences having a special property about cross-correlation between the sequences in the set. We start with the definition of cross-correlation.

Definition 1 (Cross-correlation). Given two equal-sized binary sequences $\mathbf{a} = (a_1, a_2, \ldots, a_{2^n-1}), \mathbf{b} = (b_1, b_2, \ldots, b_{2^n-1}), (a_i, b_i \in \{0, 1\}, \forall i \in \{1, 2, \ldots, 2^n - 1\})$ cross-correlation θ is defined as

$$\theta(\mathbf{a}, \mathbf{b})(\tau) = \sum_{i=1}^{2^n-1} \mathcal{X}(a_i)\mathcal{X}(b_{i+\tau})$$

$\mathcal{X}$ is a map from 0 and 1 to 1 and -1 respectively.

Hence, the cross-correlation of two binary sequences at relative position τ is the number of bits with same values minus the number of bits with different values. Gold code sequences have the following special cross-correlation property.

Theorem II.1 (Gold's theorem: three cross-correlation values for Gold code sequences). *Given two Gold code sequences* $\mathbf{a}$ *and* $\mathbf{b}$ *with size* $2^n - 1$*, the cross-correlation* $\theta(\mathbf{a}, \mathbf{b})(\tau)$ *(*$\tau \in \{0, 1, \ldots, 2^n - 2\}$*) has the following three values.*

$$\theta(\mathbf{a}, \mathbf{b})(\tau) = \begin{cases} -1,\ -(2^{(n+1)/2} + 1),\ \text{or } 2^{(n+1)/2} - 1 & \text{if } n \text{ is odd,} \\ -1,\ -(2^{(n+2)/2} + 1),\ \text{or } 2^{(n+2)/2} - 1 & \text{if } n \text{ is even.} \end{cases}$$

τ is the amount of cyclical shift of a sequence.

We do not present the proof of the above theorem and how to generate Gold code sequences; we refer the interested reader to Gold's paper [7]. Fig. 1 illustrates an example of two Gold code sequences $\mathbf{a}$ and $\mathbf{b}$ with length 7 (n is 3). We show the shifted sequences of $\mathbf{b}$ and corresponding cross-correlations. Shaded bits represent the bits with values different from sequence $\mathbf{a}$. Gold's theorem says that the cross-correlations of $\mathbf{a}$ and $\mathbf{b}$ are -1, -5 or 3. We can see that this example matches the result of Gold's theorem.

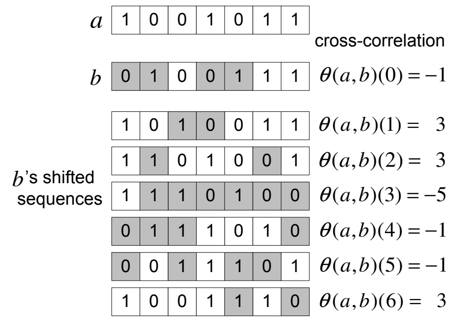


Fig. 1. Gold code example: $\mathbf{a}$ and $\mathbf{b}$ are two Gold code sequences. We present the cross-correlation of $\mathbf{a}$ and shifted versions of $\mathbf{b}$. The shaded bits of $\mathbf{b}$ represent the bits having values different from the corresponding bits of $\mathbf{a}$.

III. Protocol

In this section, we describe our medium access protocol which can be easily implemented and provides performance guarantee. In particular, we present a transmission schedule based on Gold code sequence allocated to each node and analyze the properties of our protocol. We start with our system model.

A. System Model

In our system, each node is equipped with only one radio device. Our system is a slotted system where a node decides whether to transmit or receive a packet at the start of time slots. The size of the time slots is fixed. We do not assume any clock synchronization between network nodes. We model the clock t_n of node n as $t_n = S_n t + o_n$ where S_n is clock skew of node n, o_n is offset of node n, and t is a reference clock. This model is called the affine model [9] and is used for analysis of time synchronization protocols. We do not consider secondary conflicts since they can be solved using orthogonal signals, and are thus orthogonal to our problem.

B. Transmission Schedule

We now present our medium access protocol based on Gold code sequences.

Overview. The basic operation of our medium access protocol is shown in Fig. 2. Each node has its own Gold code sequence. Each node looks up each bit sequentially in each time slot and decide whether it transmits a packet or listens to wireless medium. When a corresponding bit in a given time slot is 1, a node transmits its packet. A node listens with bit 0.

Bootstrapping. Bootstrapping a node in our protocol is very simple. The only thing to do is to assign a Gold code sequence to a node. If a topology of the network is known, a node's Gold code sequence should be unique from those of its neighbors. If the topology is not known, a Gold code of a node should be unique across the entire network.

C. Analysis

We analyze the performance of our MAC protocol based on Gold code sequences. We analyze two cases: the synchronized time slot case and the unsynchronized time slot case. For the

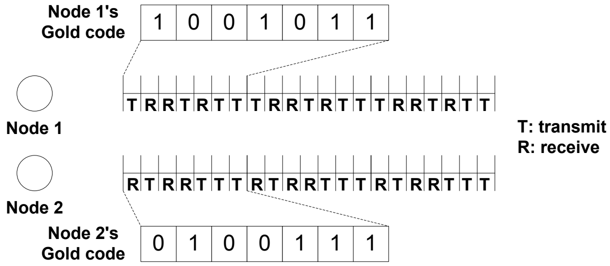


Fig. 2. Transmission schedule based on Gold code sequence allocated to each node. Given a time slot, if corresponding bit is 1, a node transmits. Otherwise, a node listens to wireless medium.

TABLE I
NOTATION

Symbol	Definition (units)
S_i	Clock skew of node i
o_i	Clock offset of node i
$L = 2^n - 1$	Length of Gold code sequence (bit)
t	Reference time (s)
$ts_i(j)$	jth time slot of node i
t_{ts}	Time slot size (s)
$gc_i(j)$	jth bit of Gold code sequence of node i

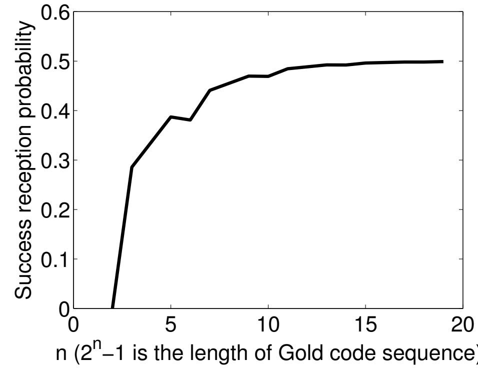


Fig. 3. Minimum success probability of reception in synchronized case

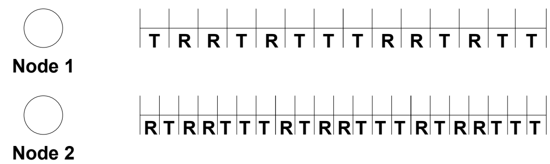


Fig. 4. Unsynchronized case

synchronized time slot case, we obtain the minimum bound of successful reception probability when two nodes have perfectly synchronized slots (not time). For the unsynchronized time slot case, we obtain a sufficient condition for two nodes to overlap time slots corresponding to same bit position of Gold code sequences. Table I summarizes the notations that we use in our analysis.

Theorem III.1 (Guaranteed successful reception probability (synchronized case)). *If two nodes' time slots are perfectly synchronized, the success probability of reception is bounded by* $\frac{2^n - 2^{(n+1)/2}}{2(2^n-1)}$ *(n is odd) or* $\frac{2^n - 2^{(n+2)/2}}{2(2^n-1)}$ *(n is even).*

Proof: We call the two nodes node 1 and node 2. In a given time slot, a packet transmitted by a node is successfully received by the other node when the other node listens to wireless medium. If we assume that each of the two nodes starts its clock at the same time, reception is successful when $gc_1(j) \neq gc_2(j), \forall j \in \{1, 2, \ldots, L\}$. However, we do not assume clock synchronization. Hence, we need to consider all possible relative positions (from 1 to $L-1$) between gc_1 and gc_2. Appealing to Gold's theorem, the number of same bit values minus the number of exclusive bit values between two Gold code sequences can take on one of three values. Among those three values, the minimum number of exclusive bit values is $2^{(n+1)/2} - 1$ (n is odd) or $2^{(n+2)/2} - 1$ (n is even). So the number of exclusive bits is $\frac{(2^n - 2^{(n+1)/2})}{2}$ (n is odd) or $\frac{(2^n - 2^{(n+2)/2})}{2}$ (n is even). Consequently, the minimum success probability of reception is $\frac{2^n - 2^{(n+1)/2}}{2(2^n-1)}$ (n is odd) or $\frac{2^n - 2^{(n+2)/2}}{2(2^n-1)}$ (n is even). ∎

An application of Theorem III.1 is to design the length of Gold code sequence given a desired probability of reception. Fig. 3 shows that $n = 20$ gives the maximum success probability among those tested.

So far, we have considered the case where nodes' clocks are perfectly synchronized. We now consider the unsynchronized clock case where the drifts and skews of the clocks of nodes can differ. Fig. 4 illustrates an example where node 2's clock skew is faster than node 1. In this case, we cannot use the minimum success probability of reception that we obtained for the synchronized case. One way to analyze our unsynchronized case might be to exhaustively enumerate the intervals having exclusive bit values for various clock skews. However, this method does not give much insight into the general behavior of the unsynchronized case. Our approach to analyzing the unsynchronized case is to obtain a loose lower bound for guaranteed performance. Gold code sequences guarantee the number of exclusive bits values among two sequences. To use this property, we attempt to find a condition under which time slots corresponding to the same bit position overlap.

To determine the interval between such conditions, we model the overlapping intervals. We consider two nodes 1 and 2. For both nodes, we represent time intervals for a given time slot index with the affine clock model. We then obtain inequalities for when two nodes have overlapping intervals for the same time slot index; these inequalities provide a sufficient condition for the existence of overlapping intervals.

We now introduce our model. Since we use affine clock model, node i's clock reads $S_i t + o_i$ at reference time t for $\forall i \in \{1, 2\}$. We assume that the initial time slot of a node starts at local time 0. We assign time slot index 0 to the initial time slot. The time slot index is assigned to the following time slots sequentially. In the Lth time slot, the index returns to 0. We call the first $L-1$ time slots group 0 and the $(p+1)$th $L-1$ time slots group p. Hence, for node i, time slot $j \in$

$\{0, 1, \ldots, L-1(=2^n-2)\}$ for group $p_i \in \{0, 1, \ldots\}$ spans the time interval from $p_i L t_{ts} + j t_{ts}$ to $p_i L t_{ts} + (j+1) t_{ts}$ at local time scale. We denote the time interval at reference time scale by $I_i(j, p_i)$ as below.

$$I_i(j, p_i) = [(p_i L t_{ts} + j t_{ts} - o_i)/S_i, (p_i L t_{ts} + (j+1) t_{ts} - o_i)/S_i]$$

We denote $(p_i L t_{ts} + j t_{ts} - o_i)/S_i$ and $(p_i L t_{ts} + (j+1) t_{ts} - o_i)/S_i$ by $I_{Si}(j, p_i)$ and $I_{Ei}(j, p_i)$ respectively. For the two nodes to have overlapping time intervals for the same time slot index j, one of the two conditions should be satisfied :

$$I_{S1}(j, p_1) < I_{S2}(j, p_2) < I_{E1}(j, p_1) \quad (1)$$

$$I_{S1}(j, p_1) < I_{E2}(j, p_2) < I_{E1}(j, p_1) \quad (2)$$

For (1), we get the following inequalities:

$$(p_2 L t_{ts} + j t_{ts} - o_2) S_1 < (p_1 L t_{ts} + (j+1) t_{ts} - o_1) S_2 \quad (3)$$

$$(p_1 L t_{ts} + j t_{ts} - o_1) S_2 < (p_2 L t_{ts} + j t_{ts} - o_2) S_1 \quad (4)$$

For (2), we get the following inequalities:

$$(p_2 L t_{ts} + (j+1) t_{ts} - o_2) S_1 < (p_1 L t_{ts} + (j+1) t_{ts} - o_1) S_2 \quad (5)$$

$$(p_1 L t_{ts} + j t_{ts} - o_1) S_2 < (p_2 L t_{ts} + (j+1) t_{ts} - o_2) S_1 \quad (6)$$

We call these inequalities *characteristic inequalities*. When there exist integers p_1 and p_2 satisfying these inequalities, there are overlapping intervals for time slots with the same index. To get an intuition, we geometrically illustrate these inequality in Fig. 5(a). Without loss of generality, we assume $S_1 > S_2$ in this example and later theorems. The shaded region in this figure represents the region satisfying the inequalities (3),(4),(5) and (6). Since the inequalities translates into straight lines with the same slope, we can see that either there is no overlapping intervals or there are periodically overlapping intervals. Using this model, we introduce conditions for the existence of overlapping intervals.

Theorem III.2 (Absence of overlapping intervals of the same index bits of two Gold code sequences). *If* $S_1 > S_2$ *and* $S_1 = k S_2 (k \in \{1, 2, \ldots\}$*, there is no overlapping interval if* $\frac{((j+1)t_{ts} - o_2)S_1 - (j t_{ts} - o_1) S_2}{L t_{ts} S_2} < \lfloor \frac{(j t_{ts} - o_2) S_1 - ((j+1) t_{ts} - o_1) S_2}{L t_{ts} S_2} \rfloor + 1$

Proof: **Intuition.** As shown in Fig. 5(b), there exist cases having no integer solutions for (p_1, p_2) with $S_1 = k S_2$. Given $p_2 = l(l \in \{0, 1, \ldots\})$, there is no integer p_1 when the region represented by our characteristic inequalities is within two adjacent integers. We identify p_1-intercepts of lines having such regions and obtain a condition for having no integer solution (p_1, p_2).

To obtain the condition such that the p_1 interval represented by characteristic inequalities is between two adjacent integers, we use p_1-intercepts of the lines corresponding inequalities (3) and (6). p_1-intercepts of the line for (3) is

$$\frac{(j t_{ts} - o_2) S_1 - ((j+1) t_{ts} - o_1) S_2}{L t_{ts} S_2} \quad (7)$$

p_1-intercepts of the line for (6) is

$$\frac{((j+1) t_{ts} - o_2) S_1 - (j t_{ts} - o_1) S_2}{L t_{ts} S_2} \quad (8)$$

Hence, when the region between (7) and (8) is between two adjacent integers, the following inequality holds: $\frac{((j+1)t_{ts} - o_2)S_1 - (j t_{ts} - o_1) S_2}{L t_{ts} S_2} < \lfloor \frac{(j t_{ts} - o_2) S_1 - ((j+1) t_{ts} - o_1) S_2}{L t_{ts} S_2} \rfloor + 1$ ■

Theorem III.2 is a sufficient condition for which there is no one-to-one mapping between the same index bits of Gold code sequences of two nodes. It does not mean that there is no chance for such two nodes to communicate with each other since bits with different index can have exclusive values (e.g. '1' and '0').

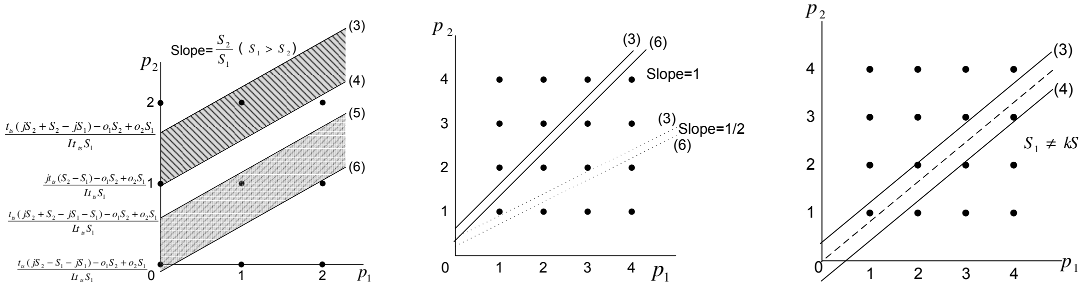
Theorem III.3 (Existence of overlapping intervals). *If* $S_1 \neq k S_2 (k \in \{1, 2, \ldots\})$*, there is integer solution* (p_1, p_2) *satisfying our characteristic inequalities.*

Proof: **Intuition.** Fourier-Motzkin elimination [10] is a general method used to obtain integer solution given a system of linear inequalities. However, we may not use this method since the lines' slopes are equal. Instead, we check if the region specified by the characteristic inequalities can meet an integer solution (p_1, p_2) as shown in Fig. 5(c).

We consider a middle line residing in the region specified by inequalities. For example, the dashed line in Fig. 5(c) represents the middle line for the region specified inequalities (3) and (4). The same logical argument apply for inequalities (5) and (6). Hence, we consider only inequalities (3) and (4). As the first step, we fix the y-intercept of the middle line as (p_1, p_2)=(0,0). We relax this constraint later. We map the slope of the middle line to the real number space. When the slope (S_2/S_1) of the middle line is a rational number, it is clear that there are integer solutions (p_1, p_2). When S_2/S_1 is an irrational number, the middle line will not meet an integer solution (p_1, p_2). However, we use a well known theorem about density of rational numbers: Given any two real numbers $\alpha, \beta \in R$, $\alpha < \beta$, there is a rational number r in Q such that $\alpha < r < \beta$. It implies that for any $\epsilon > 0$, there exists rational number r such that $r = S_2/S_1$(irrational number) + ϵ. As the lines representing inequalities (3) and (4) are not the same, we can say that there are integer solutions (p_1, p_2) satisfying the inequalities (3) and (4). When y-intercept is (y($\neq$ 0,0), our mapping the slope of the middle line to real number space is just $y-shifted$ version of the original mapping with y-intercept=0. Hence, due to the density of rational numbers, there are integer solutions (p_1, p_2) satisfying inequalities (3) and (4). ■

IV. Simulation

In this section, we evaluate the performance of our scheme by simulation. We developed our simulation tool using SMPL [11], a discrete-event-based simulation framework. We show that our scheme provides a guaranteed performance even in short-term.



(a) With assumption $S_1 > S_2$, the shaded region is the region represented by our characteristic inequalities. Each line maps to the inequalities (3),(4),(5) and (6).

(b) $S_1 = kS_2 (k \in 1, 2, \ldots)$

(c) $S_1 \neq kS_2 (k \in 1, 2, \ldots)$

Fig. 5. Illustration of conditions for the existence of integer solutions (p_1,p_2)

A. Methodology

In our simulations, we made some assumptions. First, each node decides its action to transmit or receive a packet at each time slot having fixed size. Second, each node decides its action without any knowledge about other nodes' action. In other words, there is no explicit coordination for transmitting or receiving between nodes. Third, we assume that nodes in the network always have packets to send and they are able to utilize all available time intervals for communication. This assumption implies that we are concerned about the maximum performance that a network can provide. Our baseline protocol is a random MAC where a node randomly decides whether to transmit or receive. In particular, a node chooses to transmit a packet with probability 0.5 at each time slot. The random MAC does not require the coordination of nodes' transmitting or receiving. We compare the performance of our scheme with the random MAC scheme. We perform our evaluation in the case of synchronized and unsynchronized time clocks.

B. Synchronized time clock

In this section, we evaluate our scheme in the case that all nodes are synchronized.

Long-term results. To verify the theoretical result in Theorem III.1 and our simulation implementation, we compare the fraction of available slots that we get from our analysis and simulation. In our simulated network, there are two nodes that communicate between each other. We measure the fraction of available slots for each user. We performed our simulation for our Gold-code-based MAC and random MAC. For Gold-code-based MAC, we vary the order of Gold code sequence generator from 3 to 11. Each simulation spends 100000 time slots which is enough time slots for system to stabilize. We ran each simulation 10000 times. We plot the measured average fraction of available slots with theoretical results in Fig. 6(a). In random MAC, the fraction of available slots on a long time scale should be 0.25 when there are two nodes. Our simulation result verifies the theoretical value for random MAC. Our Gold-code-based MAC does not provide a much better fraction of available slots on a long time scale. Depending on the phase difference between two nodes, the fraction of available slots of the Gold-code-based MAC can be lower than random MAC. The results for short Gold code sequence is worse than random MAC as expected by Theorem III.1.

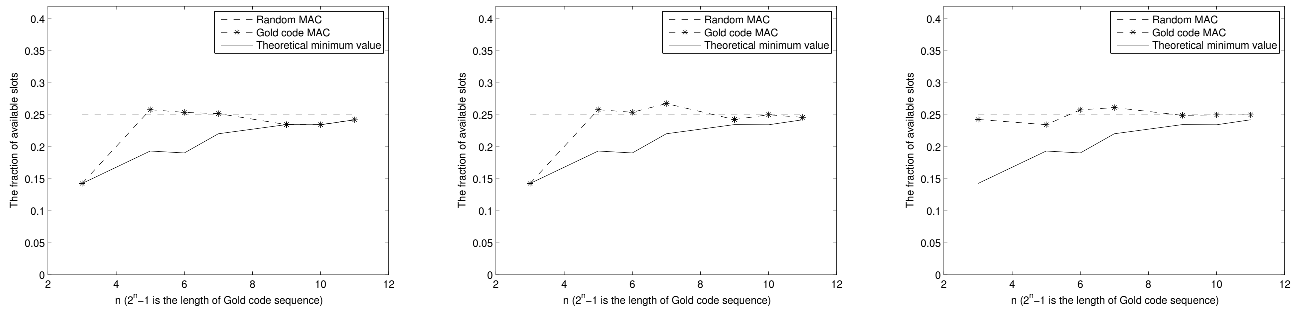
Guaranteed performance. The design goal of the Gold-code-based MAC is not to improve the fraction of available slots on a long time scale but to guarantee a level of performance (the fraction of available slots) even on a short time scale. Our second simulation shows that our Gold-code-based MAC achieves the design goal.

In our simulation, there are two nodes again. We measure a cumulative distribution for the fraction of available slots from 10000 simulation runs. The length of Gold code sequence that we use is 31(=$2^5 - 1$). This length is the smallest length for which Gold-code-based MAC can provide performance comparable to random MAC. We vary the number of time slots simulated to understand short-term performance. Fig. 7(a) compares the cumulative distributions of the fraction of available slots for random MAC and Gold-code-based MAC. Our results show that the distribution of random MAC heavily depends on the number of time slots. Random MAC does not guarantee the fraction of available slots with 100% assurance. It guarantees the fraction of available slots with some probability. For example, random MAC provides the fraction of available slots as 0.23 with probability 0.63, 0.92 and 1 when the number of slots is 100, 1000 and 10000 respectively. In contrast, Gold-code-based MAC guarantees the fraction of available slots as 0.23 with 100% assurance for all cases.

C. Unsynchronized time clock

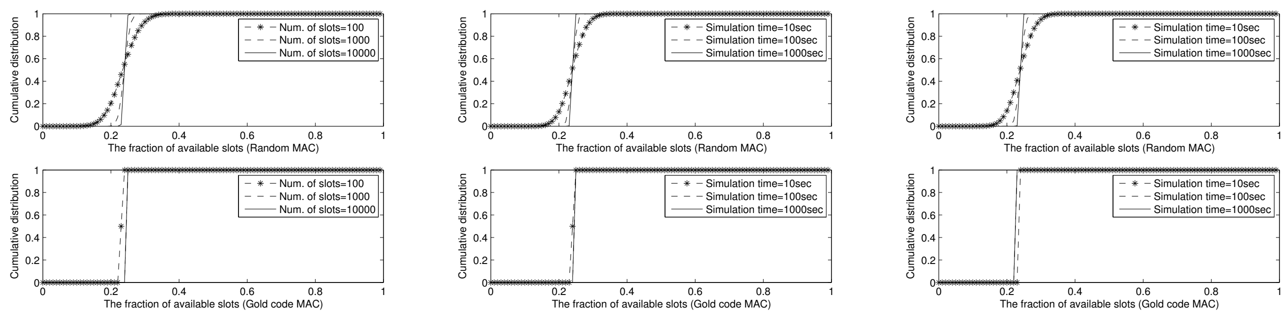
So far, our simulation assumes the synchronization of clocks of nodes. In this section, we perform a simulation study for the unsynchronized cases to understand the effect of unsynchronized time slots on system performance.

We simulated two different unsynchronized cases: $S_1 = 2S_2$ and $S_1 = 1.6S_2$. We used the same values for other parameters. Fig. 6(b) and Fig. 7(b) are for $S_1 = 2S_2$. Fig. 6(c) and Fig. 7(c) are for $S_1 = 1.6S_2$. We can see that general trend in these figures are similar to the synchronized case. Gold-code-based MAC still guarantees the fraction of available slots with few number of time slots.



(a) Synchronized case ($S_1 = S_2$). (b) Unsynchronized case ($S_1 = 2S_2$). (c) Unsynchronized case ($S_1 = 1.6S_2$).

Fig. 6. Comparison of the fraction of available slots of random MAC and Gold-code-based MAC on a long time scale. Random MAC can be better than Gold-code-based MAC as expected by Theorem III.1. However, our design goal is to achieve guaranteed performance not to enhance long-term performance.



(a) Synchronized case ($S_1 = S_2$). (b) Unsynchronized case ($S_1 = 2S_2$). (c) Unsynchronized case ($S_1 = 1.6S_2$).

Fig. 7. Cumulative distributions of the fraction of available slots in terms of the number of time slots that nodes spend. While the distribution of random MAC heavily depends on the number of time slots, the distribution of Gold-code-based MAC does not. Gold-code-based MAC can guarantee the fraction of available slots.

V. Related Work

Rental and Kunz applied Reed-Solomon and Hermitian to MAC protocol [12]. Their work is similar to our work in the sense that they apply a code sequence with special property to MAC protocol. However, we apply different type of code sequence and our analysis considers unsynchronized case as well as synchronized case.

VI. Conclusion

In this paper, we have presented the application of Gold code sequence to MAC protocol. We theoretically analyzed the guaranteed performance of Gold-code-based MAC in synchronized and unsynchronized cases. Our analysis shows that Gold-code-based MAC guarantees the fraction of available slots. By simulation, we verified our analysis and performed comparative studies between Gold-code-based MAC and random MAC.

Acknowledgements

This material is based upon work partially supported by NSF under Contract CNS-0953600 and USARO under Contract No. W-911-NF-0710287.

References

[1] I. Chlamtac and A. Faragó, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Trans. Netw.*, vol. 2, no. 1, pp. 23–29, 1994.

[2] E. Sousa and J. Silvester, "Spreading code protocols for distributed spread-spectrum packet radio networks," *Communications, IEEE Transactions on*, vol. 36, no. 3, pp. 272 –281, mar 1988.

[3] B. H. Liu, N. Bulusu, H. Pham, and S. Jha, "Csmac: A novel ds-cdma based mac protocol for wireless sensor networks," in *Global Telecommunications Conference Workshops, 2004. GlobeCom Workshops 2004. IEEE*, 29 2004, pp. 33 – 38.

[4] J. G. Andrews, S. Weber, and M. Haenggi, "Ad Hoc Networks: To Spread or not to Spread?" *IEEE Communications Magazine*, vol. 45, no. 12, pp. 84–91, Dec. 2007.

[5] A. Muqattash and M. Krunz, "Cdma-based mac protocol for wireless ad hoc networks," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*. New York, NY, USA: ACM, 2003, pp. 153–164.

[6] K. Kusume, R. Vilzmann, A. Mller, C. Hartmann, and G. Bauch, "Medium access in spread-spectrum ad hoc networks with multiuser detection," *EURASIP Journal on Advances in Signal Processing*, 2009.

[7] R. Gold, "Optimal binary sequences for spread spectrum multiplexing," *Information Theory, IEEE Transactions on*, vol. 13, no. 4, pp. 619 – 621, oct 1967.

[8] T. Kasami, "Weight distribution formula for some class of cyclic codes," Coordinated Sci. Lab., Univ. Illinois, Urabana-Champaign, IL, Tech. Rep. R-285(AD 635274), 1966.

[9] N. M. Freris and P. R. Kumar, "Fundamental limits on synchronization of affine clocks in networks," in *the 46th Conference on Decision and Control, IEEE*, 2007, pp. 921 – 926.

[10] A. Schrijver, *Theory of linear and integer programming*. New York, NY, USA: John Wiley & Sons, Inc., 1986.

[11] M. H. MacDougall, *Simulating computer systems: techniques and tools*. Cambridge, MA, USA: MIT Press, 1987.

[12] C. H. Rentel and T. Kunz, "Mac coding for qos guarantees in multi-hop mobile wireless networks," in *Q2SWinet '05: Proceedings of the 1st ACM international workshop on Quality of service & security in wireless and mobile networks*. New York, NY, USA: ACM, 2005, pp. 39–46.