

# E-MAC: An evolutionary solution for collision avoidance in wireless ad hoc networks

Haitao Zhao, Jibo Wei, Nurul I Sarkar, Shengchun Huang

**Abstract**— Collision avoidance mechanism is one of the main challenging design issues in distributed multi-hop ad hoc networks. Packet collision occurs more frequently with the increasing number of users in the network and consequently it reduces the overall network performance. To overcome this performance problem we propose an evolutionary scheme (called E-MAC), inspired by the inverse problem of mutual synchronization among fireflies to obtain collision-free schedule for ad hoc networks. In E-MAC, each source node adjusts its next transmission time so that colliding packets can be separated the transmission time of colliding nodes iteratively. And this act of E-MAC will result to that group of nodes converge to a collision-free network. E-MAC is simple and does not require any central coordinator or global time synchronization for transmissions. The performance of E-MAC has been evaluated by extensive simulation. Results obtained show that E-MAC can effectively be used to resolve packet collisions with much less convergence time than the existing protocols. The research findings reported in this paper provide some insight into and solution to collision avoidance in wireless Ad Hoc networks.

**Index Terms**—Ad hoc networks, Collision avoidance, MAC, Evolutionary scheme

## 1. INTRODUCTION

Due to the broadcast nature of wireless communication and the contention for shared wireless medium, packet collision occurs more frequently with the increasing number of users in multi-hop wireless ad hoc networks. When collision occurs, the packets usually cannot be correctly received resulting in wastage of channel bandwidth. Moreover, collisions cause packet retransmissions that consume significant amount of radio resources such as channel bandwidth and energy, which can further reduce the system performance. The request-to-send/Clear-to-send (RTS/CTS) handshake mechanisms are commonly used to alleviate collisions in wireless networks. However, the control packets (e.g., RTS and CTS) and waiting time (e.g., inter-frame sequences and backoff time) significantly decrease the network efficiency [1, 2]. The situation is even worse in high-speed wireless

networks such as IEEE 802.11 ac standards where the PHY rate reaches as high as 1Gbps [3]. The control packets will decrease the network efficiency more severely because they can only be transmitted at a pretty low rate for correct reception. Therefore, avoiding collision in a distributed manner without using low-rate control packets is currently an appealing challenge.

Existing medium access control (MAC) protocols for collision avoidance can be broadly classified into four categories, (i) coordination-based schemes that utilize a central coordination for resource allocation [4, 5]; (ii) multi-frequency assistant schemes that either use out-of-band signaling to avoid colliding transmission [6, 7] or use multiple frequency bands for concurrent transmission [8-10]; (iii) backoff-tuning schemes that use different backoff strategies according to whether the former transmission is successful or not [11-14]; and (iv) slot-assignment schemes that reserve different time slots for different transmissions [15-18]. These MACs either require extra assistance and information exchange, which will introduce some other additional system complexities (including neighbor discovery, local frame exchange, global time synchronization and more occupied frequency), or experience a long convergence time before setting up the collision-free schedule. To overcome the above problems and challenges, we propose an evolutionary MAC (E-MAC) as an alternative solution for collision avoidance in wireless networks. The key concept of E-MAC is inspired by the mutual synchronization phenomena in firefly flashing. When a firefly meets with other fireflies and observes flashes, it reacts to the perceived stimuli and adjusts its internal timer so as to flash at the same rate as the others, resulting in synchronized flashing of the whole group. This process is modeled as a set of pulse-coupled oscillators (PCO) [19], and has been used to study neuron synchronization [20], distributed synchronization in multi-agent networks [21] and sensor networks [22, 23]. The oscillation-based synchrony is believed to be the most energy-efficient physical mechanism for temporal coordination [24]. Inspired by the inverse problem of this PCO process, E-MAC resolves collisions by adjusting its next transmission time and therefore affected nodes iteratively converge to a collision-free network. The main characteristics of E-MAC are highlighted below:

- E-MAC is a distributed protocol where there is no need for a coordination, global time synchronization, out-of-band signaling, or multi-frequency assistance.
- For a correct operation, there is no need for extra control information exchanges because the

required information can be derived from each node's own collided packets. Unlike existing schemes, E-MAC can support variable packet lengths.

- E-MAC is a dynamic adaptive protocol resistant to possible system errors (e.g., inaccurate slot split and slot drift) and scalable to network size.

The rest of the paper is organized as follows: Section 2 reviews relevant literature on collision avoidance protocols. Section 3 introduces the key idea of the proposed E-MAC protocol followed by the evolutionary algorithm that is adaptive to dynamic network environment. The implementation aspect of the proposed protocol is also discussed in this section. The performance of the proposed protocol is evaluated in Section 4, and a brief discussion in Section 5 concludes the paper.

## **2. RELATED WORK**

The collision avoidance protocols for wireless ad hoc networks originated from Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). However, the efficiency of CSMA/CA is low due to its random backoff processes (often use a binary exponential backoff (BEB) process) and the use of low-rate RTS/CTS control packets. Barcelo, et al. proposed L-BEB [11] which is evolved from BEB. It chooses backoff values based on the success or failure of the last transmission. In particular, nodes choose a fixed backoff counter after a successful transmission. The value of the fixed backoff counter is shared by all nodes, and is the cycle length for the network. After a collision, nodes choose the backoff counter uniformly at random, as in the distributed coordination function (DCF). If the number of nodes is less than the cycle length, L-BEB settles into a collision-free state after a set-up period. The similar idea of using a deterministic backoff after successful transmissions has been explored by many network researchers e.g., SRB [12], hashing backoff [13], and L-MAC [16]. L-MAC [16] is actually an update version of L-BEB. It keeps updating selection probability for each slot including all idle and busy slots in the previous cycle. For a successful transmission, a node persists with the same slot in the following cycle by setting the selection probability for that slot to 1. Barcelo, *et al* [14] extend this backoff-tuning idea to multi-hop packet radio networks in the presence of hidden stations. Each wireless node computes the schedule length after gathering information about the number of flows in its neighborhood. Then, a deterministic backoff is used after successful transmissions and a random backoff is used otherwise.

The basic principles of these MAC protocols are similar in the sense that they iteratively change the transmission time of each node to finally converge to a collision-free schedule. Therefore, authors in [25] modeled the convergence process of these schemes by an absorbing Markov chain, and thus derived the expected number of iterations required to reach a collision-free schedule. However, all of these MAC schemes assume that each node's transmission occupies the same amount of slots, and they also have the drawbacks of slow convergence speed to a collision-free state and are vulnerable to new entrants to the wireless network.

Z-MAC [26] is a hybrid protocol that combines TDMA and CSMA in wireless multi-hop networks. Z-MAC assigns a slot to each node which can be borrowed by other nodes for contention if the owner of this slot has no data for transmission. However, Z-MAC requires extra information exchange beacons, and it introduces additional system complexity, including neighbor discovery, local frame exchange and global time synchronization. ZC [15] also follows the idea of combining TDMA and CSMA but in a more flexible manner. In ZC, each node records all the idle slots in a cycle. For a transmission collision, the source node uniformly selects one slot for retransmission in its next cycle from a candidate slot set which is composed of all the idle slots and the slot it just used. L-ZC [16] is an evolution of ZC that adopts different probabilities for selecting an idle slot and the slot it has just used. To satisfy users' demands for bandwidth, a randomized MAC scheme for wireless mesh networks is proposed in [17] that supports multiple fixed-length slots in a schedule using one-hop message passing. These methods are effectively allow each node to independently produce a periodic schedule for transmissions (i.e. MAC slots). But these slot-assignment mechanisms are vulnerable to slot drifts (non-ideal clocks) [27]. Table 1 lists some representative collision avoidance schemes reviewed in this section. We have only listed the operation of each scheme after a failed transmission, since they all persist in the same slot as just used in the previous cycle after a successful transmission.

TABLE 1

SUMMARY OF EXISTING COLLISION AVOIDANCE SCHEMES<sup>1</sup>

Scheme	Operation after a collision
ZC	Randomly select a slot from all idle slots and the one it just used
L-BEB	Randomly select a slot throughout the cycle
L-ZC	Select the slot that it has just used with probability of $r$ , while select one from all idle slots with the probability of $(1-r)/n_i$ , where $n_i$ is the total number of idle slots in the previous transmission cycle.
L-MAC	Given the probability that the node select Slot $m$ in the Transmission cycle $n$ as $p_m(n)$ , then $p_m(n+1) = \beta p_m(n)$ $p_j(n+1) = \beta p_j(n) + \frac{1-\beta}{T-1}, \quad (j \neq m)$

In addition to the slow convergence speed, all the existing schemes discussed earlier have three important issues/problems. First, they implicitly assume that each node's transmission occupied the same amount of MAC slots, which is not always practical. Second, these schemes rely on the knowledge of the idle slots in the previous cycle. But to identify whether each slot is idle or busy is a nontrivial problem, especially at the presence of hidden- and exposed-node problems. Third, most of these schemes are actually schedule transmission for the collided nodes in the previously idle slot which is not an efficient way to resolve collisions. For instance, in the scenario illustrated in Fig. 1(a), according to L-BEB or L-ZC, both Nodes 1 and 2 will have higher probability to start their transmissions in the idle period of B, which will inevitably bring collision again. The better solution is that they still start their transmission in the busy period of A with a little adjustment (Fig. 1b).

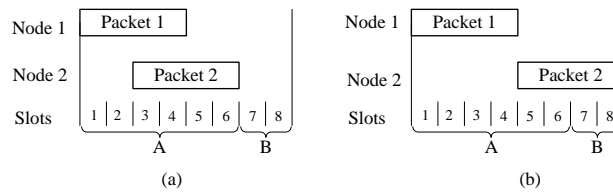


Fig.1. Illustrating the basic principle of packet collision

<sup>1</sup> In Table 1,  $r$  and  $\beta$  are both adjustable parameters that  $r \in (0,1)$  controls the probability that it choose the same slot after a collision, and  $\beta \in (0,1)$  represents the learning strength.  $T$  is the length of the transmission cycle, in terms of MAC slot numbers.

### 3. EVOLUTIONARY SCHEME FOR COLLISION AVOIDANCE

#### 3.1 Collision scenarios in ad hoc networks

The overlap of the transmission time for more than two nodes (i.e. within each other's interference range) will result in transmission collisions. There are two main cases of collision in wireless networks: (i) the synchronized collision, which occurs when no less than two nodes happen to start their transmissions simultaneously (Fig. 2 Case A); and (ii) the hidden node collision. Unlike the synchronized collision, the collided packets start at the same time when hidden nodes exist, the colliding packets can be overlapped at any stages as illustrated in Fig. 2 (cases B to D). In multi-hop ad hoc scenarios, hidden node problem is one of the main causes for collisions. The authors in [28] [29] modeled hidden node collision and analyzed its effect on end-to-end throughput demonstrating that hidden node collision can decrease the throughput by more than 20% in high density networks<sup>2</sup>.

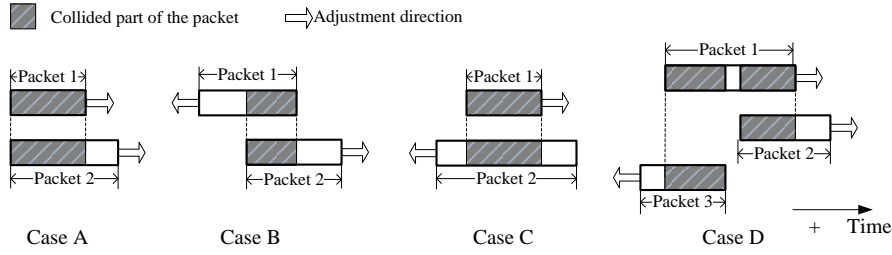


Fig. 2. Typical packet collision scenarios

#### 3.2 Framework to avoid collision

Suppose there are  $n$  nodes that can interfere with each other. Each node is an oscillator with the same fundamental frequency that transmits once in one period. The system model is illustrated in Fig. 3, where we bend the time line into a running circle. The perimeter of this circle equals to the length of one period, which is normalized as 1 in this subsection for better explanation. Let the nodes be labeled in clockwise order. And in the case that  $m$  nodes share the same phase, supposing the node after them is labeled as  $k$ , their labels will be randomly allocated from  $k+1$  to  $k+m$ .

We use the tuple  $\Psi_i = (\phi_i, l_i)$  to denote the state of Node  $i$ . Where  $\phi_i \in [0, 1]$  denotes the start phase of Node  $i$ 's transmission, and  $l_i \in [0, 1]$  denotes the packet length in this transmission. For example, if

<sup>2</sup> Note that each transmission in Fig. 2 and Fig. 3 is actually the combination of DATA packets and the acknowledgements, which can be a single DATA-ACK transmission or a burst transmission with multiple DATA-ACKs. For simplicity, we illustrated them as one virtue packet of variable length.

$\phi_i = 0.75$  and  $l_i = 0.02$ , then Node  $i$  is 75% of the way through its cycle, and its transmission will occupy 2% of the time cycle. The *phase neighbors* of node  $i$  are the nodes  $i \pm 1$ . The *system state* is a column vector  $\bar{\Psi} = [\Psi_i]$ , representing the states of all  $n$  nodes.

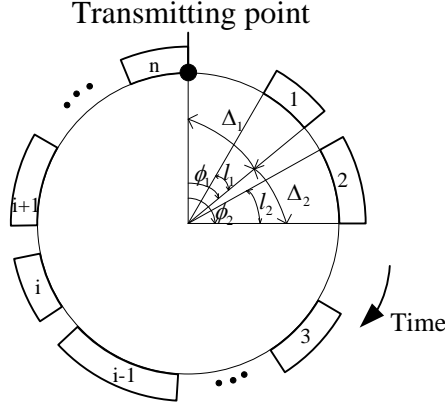


Fig. 3 The system model

We first define the collision-free state as any system state in which no transmissions overlap (Fig. 3). It is easy to understand that in a collision-free state, the phase gap between two phase neighbors will be more than the length of the leading transmission. Formally, Node  $i$  will experience no collision with its following neighbor once Equation (1) is satisfied,

$$\Delta_i \equiv \phi_i - \phi_{i-1} > l_i, \quad i = 1, 2, \dots, n, \text{ and } \phi_0 = \phi_n - 1 \quad (1)$$

And if Equation (1) is satisfied to every node, these  $n$  nodes will start their transmission in sequence without a collision and the network achieves a collision-free state. Note that in this framework, when a node just transmitted its packet ( $\phi = 1$ ), it will reset its phase to  $\phi = 0$ . The following constraints are satisfied,

$$\begin{cases} \sum_{i=1}^n \Delta_i = 1 \\ \sum_{i=1}^n l_i \leq 1 \end{cases} \quad (2)$$

Supposing the packet length of each node will keep constant in a long term, now the problem is that once collisions occur, how the collided packets adjust their phase to avoid collision in the next cycles. And in a distributed ad hoc network, the adjustment would better to depend on its only status/information, namely,

$$\phi_i' = f(\Psi_i; \alpha_i) = f(\phi_i; l_i; \alpha_i) \quad (3)$$

where  $\phi_i'$  is the new phase of Node  $i$  after the adjustment,  $\alpha_i > 0$  is the adjustment control parameter that controls how far a packet adjusts. And  $f$  is the adjust function.

### 3.3 The Proposed E-MAC Algorithm

Assuming that Node  $i$  is the next node to transmit. It will adjust its phase from  $\phi_i$  to  $\phi_i'$  as in (4), where  $\alpha_i \in (0, 1/2]$  is a random variable.

$$\phi_i' = f(\Psi_i; \alpha_i) = \begin{cases} \phi_i, & \text{when Node } i \text{ experiences no collision} \\ \phi_i - \alpha_i l_i, & \text{when Node } i \text{ experiences a collision,} \\ & \text{but the front part of Node } i \text{'s packet is not collided} \\ \phi_i + \alpha_i l_i, & \text{otherwise} \end{cases} \quad (4)$$

Equation (4) indicates that if a collision occurs, the collided nodes will adjust their phases, namely the transmission time, according to which part of its transmitted packet suffers from the collision. When no collision occurs, Node  $i$  will stick to its former phase. When a node experience a collision but the front part of its packet is not collided, namely the collision happens on the middle or the back part of the packet, it should adjust forward. Otherwise, once the front part of the packet is collided, either only the front part is collided or other parts of the packet are also collided, it should adjust backward. In Fig. 2, we have used hollow arrow to illustrate the adjustment direction of the collided packets.

Note that for Case A in Fig 2, both packets adjust towards the same direction in the first run. However, given that their adjustment step is not always the same, guaranteed by randomly selected adjustment control ( $\alpha$ ) and the possible different packet length ( $l$ ), Case A will soon turn to Case B if  $\alpha_1 l_1 < \alpha_2 l_2$  or to Case C if  $\alpha_1 l_1 > \alpha_2 l_2$  and then they will adjust towards different directions to finally avoid collision.

**Theorem 1** (E-MAC Convergence): Given the constraints defined by Equation (2), for all initial conditions,  $n$  nodes whose dynamics are governed by E-MAC algorithm will converge to a collision-free state in  $O(n)$  rounds. The proof of theorem can be found in the Appendix.

The main advantage of E-MAC is that it can shorten the overlapped part of two collided packets by adjusting the phase (i.e., transmission time) of each collided packets. Comparing to the schemes that randomly select the transmission slot, it can speed up the convergence significantly, especially under high traffic loads.



Another point of clarification is that the transmission cycle of each node in E-MAC is not necessarily to be aligned (see Fig. 4). This means that only if each node is informed about the specific cycle ( $T$ ) before deployment, no global time is needed to keep the transmission cycle of each node aligned.

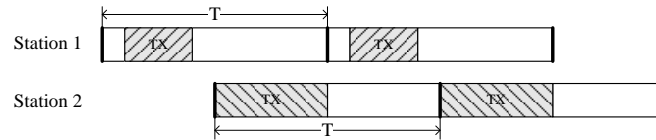


Fig. 4. Two nodes can still avoid collision even the transmission cycles are not aligned

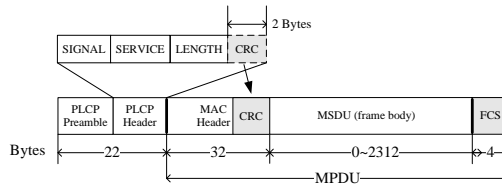
Now the main challenge is to identify the collided packets. To accomplish this, we need to revise the format of PLCP Protocol Data Unit (PPDU) in IEEE 802.11 standards (Fig. 5a). In the revised format, we move the CRC that originally at the end of PLCP Header to the end of MAC Header, which enables CRC to protect both PLCP Header and MAC Header. By considering PLCP preamble is a known parameter, we can now determine whether PLCP preamble, PLCP header and MAC header, or MPDU is successfully received. Therefore, we can easily identify which part of the packet suffers collision for a collided packet (see Supposition 1 below).

**Supposition 1:** For a collided packet, if not all of the PLCP preamble, the PLCP header and the MAC header are correctly decoded, and then this packet is collided in its front part<sup>3</sup>.

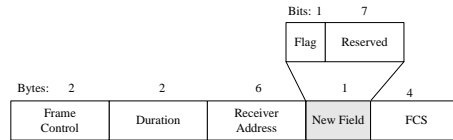
After receiving a data packet, the receiver can determine the status according to Supposition 1, and we adopt the “NACK” for the receiver to feedback the status that collision occurs but not on the front part. And the “NACK” is an extension of the traditional “ACK” with an extra flag bit to differentiate it from “ACK” (see Fig. 5b).

---

<sup>3</sup> Note that in Supposition 1, only if a packet collides in its front part, whether its other part is collided or not, it will be taken as one that collided in its front part. Though this supposition does not accurately differentiate all the cases that a packet may collide, this easy-realized idea can correctly differentiate the cases whether a collided packet should jump forward or backward in the next transmission cycle to avoid collision.



(a) DATA



(b) ACK

Fig. 5. Revision packet format

The type of virtue packet incases of successful transmission, collision occurs in the front part and otherwise as illustrated in Fig. 6. T, R and H are the transmitter, dedicated receiver and hidden node, respectively. Note that in Fig. 6b, to avoid NACK colliding with the DATA packet from H, R will wait (via carrier sensing) until H finishing its transmission to feedback the NACK. The timeout in Fig. 6c is set as long as the cycle length minus the length of the transmitted packet, i.e.,  $T-l$ .

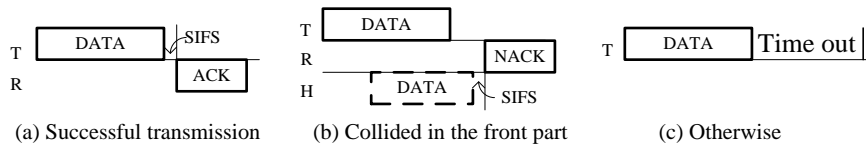


Fig. 6. Three types of the virtue packet

### 3.4 Evolutionary feature of E-MAC

There are three more problems need considering while adopt E-MAC in practice. First, all the transmissions admitted have to be finished in a single period, and they cannot cross two adjacent cycles. Second, we have used unsaturated assumption in our model (see Eq. ()), however, practically there is saturated status that one cycle cannot admit all packets to transmit. Under saturated status, collision-free state cannot be achieved whatever method we use to adjust the nodes' phases. Therefore, each node should have the ability to recognize this situation, and probabilistic backoff when it happens. And the third problem is the selection of adjustment steps. It is straightforward to understand that when the network is far from saturation, a big step will helpful to the convergence speed since it can take the

collided packets apart very soon. However, in crowded network that close to saturation, big step will be harmful to the convergence since that too far away from one packet may cause collision to another one, as illustrated in Fig. 7. Thus, in crowded situation a smaller step would be preferable. Therefore, proper selections of adjustment steps, especially the fine-grained adjustments that can adapt to dynamic network sceneries would be desired.

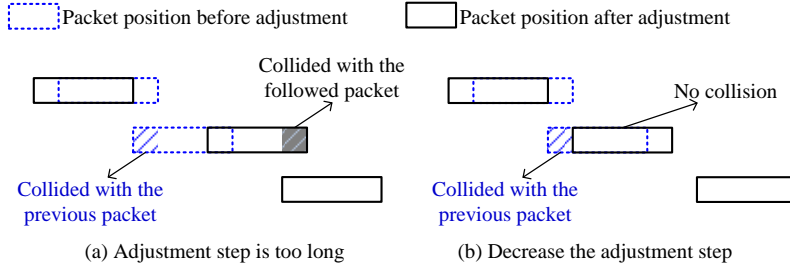


Fig. 7. Adaptive fine-grained adjustment step

For practical considerations, we enhance E-MAC by including the following evolutionary algorithms.

- If a packet moves to the start time (or end time) of the transmission period  $T$ , it will not moves forward (or backward) anymore in the following adjustment cycles.
- The iteration starts with a relative long adjustment step, for instance  $\alpha$  is randomly selected from  $9/20$  to  $1/2$  and when “moving forward” and “moving backward” appear alternatively for several times, which means the adjustment step might be too long and the adjustment step should be halved. If after a specific time of decreasing the adjustment step collision still occurs, the collided packets will respectively have the probability of  $p$  (for instance,  $p=0.5$ ) to backoff in the following  $b$  cycles, where  $b$  is an integer that randomly selected from 1 to 5.
- After the backoff process, a node will compete for transmission again as a new entrant.

TABLE 2

PSEUDO CODE OF THE EVOLUTIONARY PROCESS

---



---

//Given: Length of transmission cycle ( $T$ ), packet length ( $l$ ), the transmission time in previous cycle ( $t$ );
//Initialization (Default): Initial adjustment ( $\alpha = l/2$ ), adjustment decreasing threshold ( $\xi = 5$ ), backoff probability

---

---

---

$(p=0.5), de\_adjust\_flag=0;$

---

```
1  If (the collision occurs in the front part)
2      Move_flag (now)=backward;
3      if ((Move_flag (last time)=forward) AND (Move_flag (the time before last time)= backward))
4           $\alpha = \alpha / 2;$ 
5          de_adjust_flag ++;
6      End
7      If (de_adjust_flag >  $\xi$  )
8          Backoff with probability of  $p$ ;
9      Else
10         if ( $t+l+\alpha < T$ ) // moves towards backward
11             Packet transmission time in next cycle =  $t+\alpha$  ;
12         Else
13             Packet transmission time in next cycle =  $T-l$ ;
14         End
15     End
16     Move_flag (the time before last time)=Move_flag (last time); Move_flag (last time)= Move_flag (now)
17 End
```

---

---

The pseudo code of the evolutionary process for the case that the packet from Node  $i$  is collided in the front part (for brevity we omit the subscript of  $i$  here) is listed in Table 2. The process for the otherwise collided packets is similar to it except that the packets jump forward.

Although we use a fixed transmission period  $T$  in our proposal, it is worth mentioning that the transmission period  $T$  can be adaptive based on traffic load using a MIMD scheme as proposed in [16], where if a node finds that the schedule length is too short to accommodate all nodes it doubles the value of  $T$  being used, on the other hand if the schedule length is too large then  $T$  is halved. We leave this extension to E-MAC as possible future work.

#### 4. PERFORMANCE EVALUATION AND COMPARISON

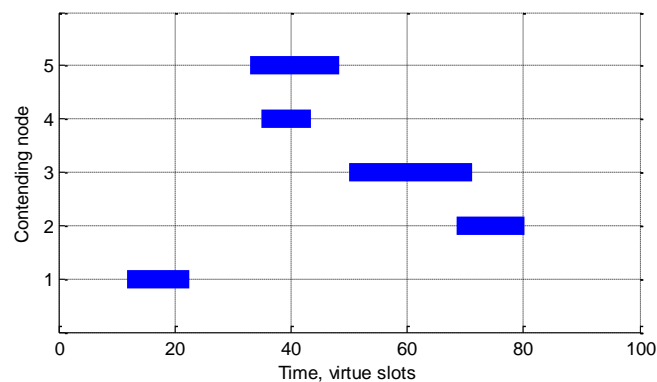
The performance of the proposed E-MAC is evaluated by extensive simulation experiments using

ns-2 simulator. We compare the performance of E-MAC with the existing four distributed collision avoidance protocols such as L-BEB, ZC, L-ZC and L-MAC. We have implemented E-MAC in NS2.32 platform. Without loss of generality, the cycle length ( $T$ ) in the simulation is set as the time needed to continuously transmit 20 packets of 1K Byte at the rate of 11Mbps. And for easier explanation, we define one *virtue slot* as  $T/100$ , and in this section we will use the *virtue slot* as the time unit.

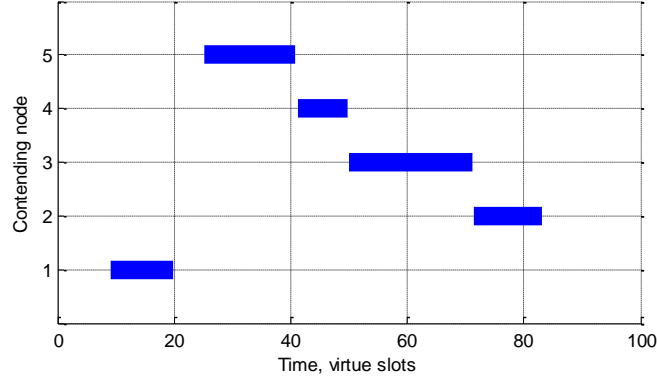
After achieving the collision-free schedule, all packets will be transmitted without collision and therefore the network performance, for instance the average throughput and transmission delay, would be the same for all collision avoidance schemes. Thus, what most matters is the performance in the set-up process, which is the focus of this evaluation section. Results presented in the next section are based on average values of 5000 simulation runs.

#### 4.1 Validation of E-MAC

We first give a simple example to illustrate the collision free feature of E-MAC . Figure 8a shows the initial transmission state of five contending nodes in the transmission cycle for 100 virtue slots, where the transmissions between Node 2 and Node 3, between Node 4 and Node 5 are collided. With the effect of E-MAC, all collided packets will adjust their transmission time accordingly. For instance, the packet from Node 4 will move forward, while packets from Node 5 will move backward in the first iteration. After only five iterations, the transmissions are scheduled collision-free as shown in Fig. 8b.



(a) The initial transmissions with collision



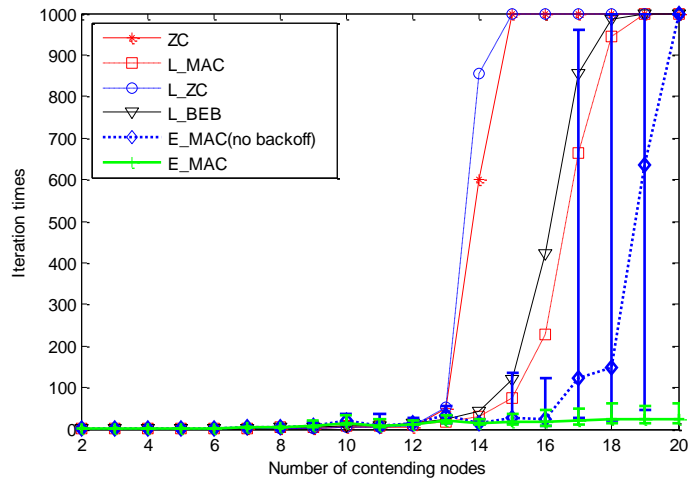
(b) The achieved collision-free state after 5 iterations

Fig. 8. Illustrating the effectiveness of E-MAC

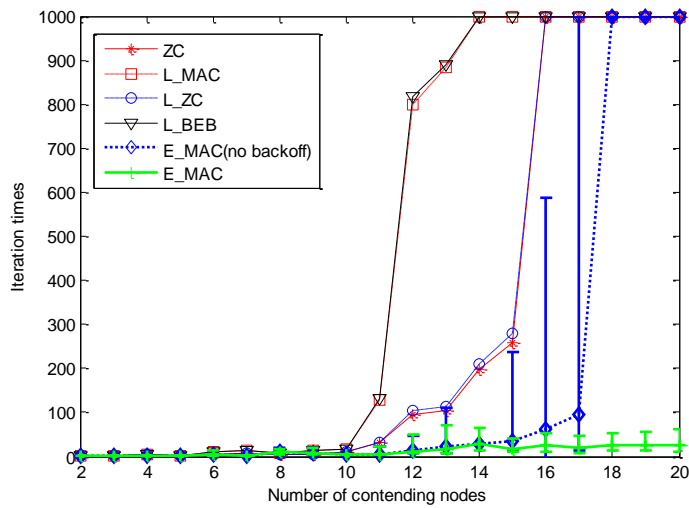
## 4.2 Speed of Convergence

To evaluate the convergence performance under different contending conditions, we range the number of contending nodes in one transmission cycle from 2 to 20. And for L-ZC and L-MAC we use the optimal parameters of  $\gamma$  and  $\beta$  as deduced in [16], i.e.,  $\gamma = 1/(T - n + 2)$  and  $\beta = 0.95$ , where  $T$  is the cycle length and  $n$  is the number of contending nodes. We will compare two cases: (i) the fixed-packet-length case that each packet has the same length of 1 virtue slots as assumed in ZC, L-BEB, L-ZC and L-MAC, and the  $T$  is set 20 virtue slots; (ii) the variable-packet-length case that the packet length of each packet is randomly selected from 1 to 10 virtue slots, and  $T$  is set 100 virtue slots.

Figure 9 plots the average simulation results for both cases. It should be noted that for ease of presentation and readability we only show the error bars of E-MAC on the graph (the randomness of other schemes makes the error bar too large to be shown clearly).



(a) The fixed-packet-length case



(b) The variable-packet-length case

Fig. 9. Convergence speed of the proposal

From Fig. 9, we observe that the randomness nature of existing schemes (i.e., ZC, L-BEB, L-ZC and L-MAC) has significant drawback of low convergence. It clearly shows that the convergence time increases very fast with the increasing number of contending nodes.

Under the scenarios with constant length of packets as existing schemes supposed, in terms of convergence speed, L-MAC is better than L-BEB, ZC and L-ZC. This is because the learning process in L-MAC can increase the convergence speed when all packets occupy the same transmission time. However, under the scenarios with different lengths of packets, the performance of ZC and L-ZC are better than L-BEB and L-MAC. This is due to L-MAC retreats close to L-BEB when packets occupy

different transmission time, and both idle slots and busy slots have almost the same probability to be selected. While in ZC and L-ZC, the idle slots have more probability to be selected, which makes them perform better in the variable-packet-length case.

On the other hand, the collided nodes adjust their transmission time according to their specific collision status in E-MAC. This adjustment is more efficient for both cases. It is clear that the convergence time of E-MAC is much less than L-BEB, ZC, L-ZC and L-MAC. This advantage becomes evident in heavy competition status (e.g. when the number of contending nodes is more than 12). More specifically, with increasing number of the contending nodes, E-MAC can decrease the convergence times more than 10 times compared to the existing schemes.

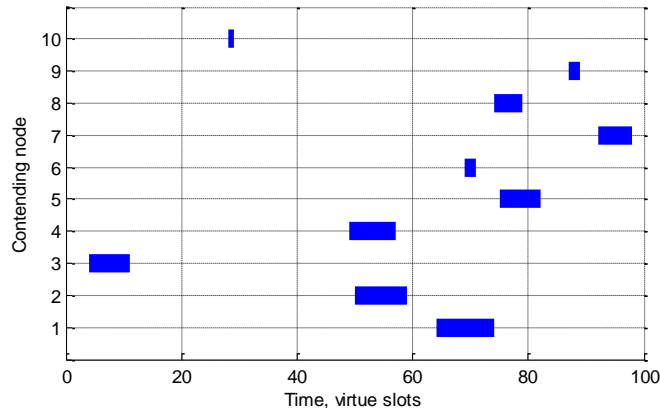
We clarify that the fast convergence characteristics of E-MAC comes in part from its ability to backoff collided packets under saturated status, which is not included in existing proposals. For a more fair comparison to check just the idea advantage of E-MAC (i.e., adjusting based on the specific collision situations), we exclude the backoff-collided-packets function of E-MAC, and also plotted its convergence performance in Fig. 9 (marked as 'E-MAC (no backoff)'). It clearly shows that even without the backoff function, E-MAC still has better convergence performance than the existing proposals.

### **4.3 Stability of iteration times**

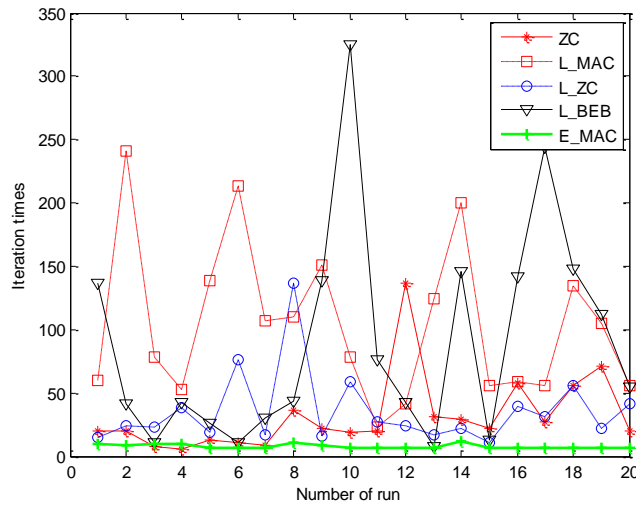
Recall that in L-BEB, ZC, L-ZC and L-MAC, the time slot a packet will transmit in the next period is decided with a certain probability. This means that even in the same network scenario different runs of L-BEB, ZC, L-ZC or L-MAC may select different time slots for next transmission and this may lead to different convergence time. Therefore, in all the existing schemes studied, the convergence time is a variable even though the given network scenario is fixed. E-MAC on the other hand can provide a much more stable convergence time and it changes a little with different runs. Figure 10b illustrates the iteration variations at the given collision status of Fig. 10a with 10 contending nodes, where we only show 20 runs using monte carlo simulation. It clearly shows that the E-MAC can provide constant convergence time, while in existing schemes this convergence time varies significantly among different runs. And if we change the given collision status, similar conclusion can be derived.



This is another significant advantage of E-MAC over existing schemes, since a stable convergence time can guarantee a stable access delay for all nodes in the network. And therefore, this virtue is very helpful to provide users with stable quality of service, and the network operator with a specific setting-up time of the network when some changes occur.



(a) The given network scenario with collision



(b) The convergence time of each run

Fig. 10. E-MAC can provide constant convergence time

#### 4.4 Channel utilization efficiency

Another important issue in collision avoidance schemes is that in the saturated status if the collision-free schedule cannot be obtained after a predefined period, then either we need to increase  $T$  (for instance, double it as in [16]), or to make some collided nodes backoff. The later solution would be useful when we are to decrease the active node's access delay. Here, we evaluate the channel

utilization efficiency, which is the ratio of the aggregated collision-free transmission time to the transmission cycle  $T$ , under saturated scenario with access delay constraints. In particular, to simulate the saturated scenario, we setting 20 nodes with the average packet length of 5 virtue slots contend to transmit in a period of 100 virtue slots cycle, and we check the channel utilization efficiency after 200 times of iterations.

Figure 11 shows the simulation results. It shows that E-MAC can averagely utilize more than 96% of the channel, L-BEB and L-MAC can achieve the channel utilization ratio around 70%, while ZC and L-ZC only achieve 60%. The higher channel utilization of E-MAC comes from two main reasons, first it adjusts the transmission time according to which part of its packets suffering from the collision, and also it enables adaptive fine-grained adjustment step. Second, it backoffs some collided packets under saturation scenario, which is helpful to have other packets successfully transmitted.

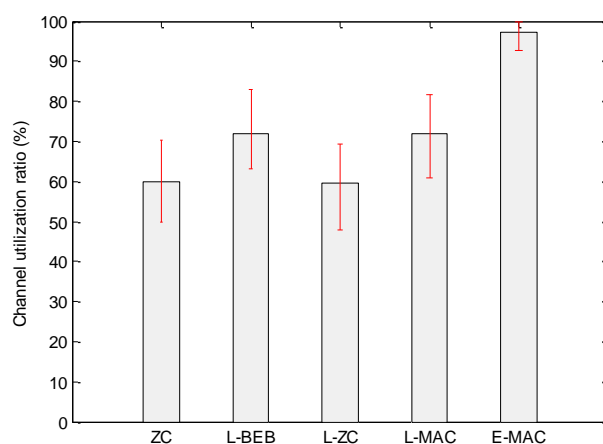


Fig. 11. Channel utilization ratio under saturation scenario

#### 4.5 Robustness to new entrants

We now consider the effect of new entrants to the performance of E-MAC. When a new node contends to enter the network, it may cause collision to the already established collision-free schedule. And then there will be a re-establishment of collision-free schedule after a few times of adjustment. The time for re-establishment reflects the robustness of the proposal against new entrants. It is clear that this time slot depends on the number of existing contending nodes and the length of packets.

We assume  $n$  contending nodes already exist in one cycle with  $T=100$  virtue slots, and still set the

average packet length as 5 virtue slots. The packets transmitted by the  $n$  nodes are collision-free and evenly distributed throughout  $T$  and then a new contending node will send a packet with the length of 5 virtue slots via a randomly selected time in  $T$ . In Fig. 12, we check the re-establishment time when a new node contends to access the channel for different values of  $n$  via using different collision-avoidance schemes.

The main advantage of E-MAC over the existing schemes is that a new entrant comes the re-establishment time for E-MAC always keeps at a very low level (below 10 times of iterations), while the re-establishment time for existing schemes increase remarkably with the number of existing nodes.

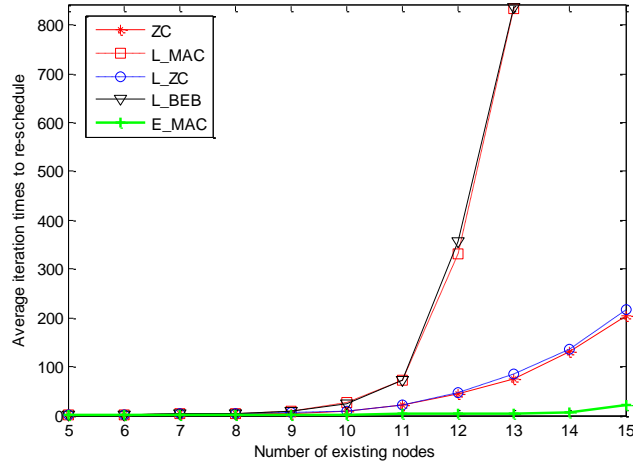


Fig. 12. Re-establishment speed after a new entrant

#### 4.6 Performance study in the presence of slot drifts

If a node miscounts the time slots we refer to this as slot drift, which may result from clock errors, hardware/software miscounting and so on. Gong and Malone [27] have pointed out that the slot drift may heavily affect the performance of ZC and L-MAC indicating that ZC and L-MAC can even be outperformed by DCF in the presence of 2% slot drift. The performance of L-MAC and ZC can be decreased by more than 40% in the presence of 10% slot drift.

To evaluate the performance of E-MAC in the presence of slot drift, we adopt the same model of slot drift as in [27]. We simulated a probability  $\theta$  of slot drift by introducing a probability of  $\theta/2$  that each node leads the idealized slot count by one slot and also a probability of  $\theta/2$  that it lags the

idealized slot count by one slot. These probabilities are applied once in each idealized slot. The average packet length is also set to be 5 virtue slots, we vary the number of contending nodes in the network in a period of 100 virtue slots. Figure 13 shows the simulation results of E-MAC in the presence of slot drift, varying from 0~100% slot drift. It clearly shows that E-MAC is only a little affected by the presence of slot drift in terms of convergence speed, although this affect can increase with the increasing number of contending nodes. Even with 12 contending nodes and 100% slot drift in the network, the performance is decreased by only around 14%.

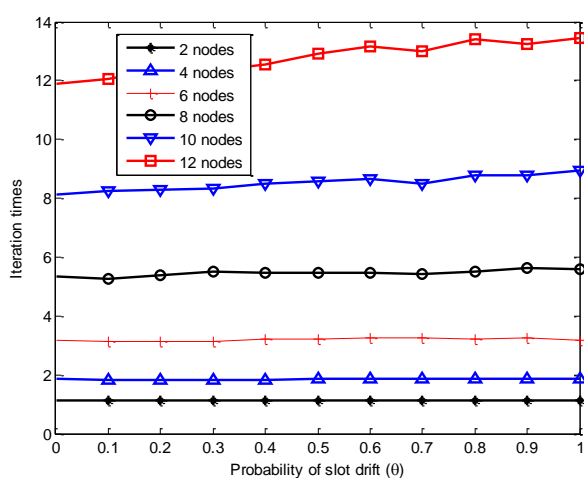


Fig. 13. Resistance to slot drift

## 5. CONCLUSION

In this paper, we proposed an evolutionary MAC scheme (called E-MAC) for collision avoidance in ad hoc wireless networks. The design principle of E-MAC lies on the observation that the collision-free transmission is actually an asynchronization process for each node that they try not to transmit packets simultaneously. To achieve this goal, we modeled the inverse problem of firefly flashing synchronization process and adopted in E-MAC. E-MAC works well in full distributed manner without requiring any central coordinator or global time synchronization, and bring no extra load to the network. Results obtained show that the proposed E-MAC can be used to provide both fast and stable convergence speed with high channel utilization. The system is also robust to both new entrants and slot drifts. We believe that E-MAC can increase the efficiency of wireless ad hoc networks. The bio-inspired features of E-MAC could also be used to solve other open challenging problems in wireless networks more effectively.

## ACKNOWLEDGMENT

This work was supported by the 863 project (Grant No. 2014AA01A701) and National Natural Science Foundation (Grant No. 61002032).

## REFERENCES

- [1] P. Chatzimisios, *et al.*, "Effectiveness of RTS/CTS handshake in IEEE 802.11 a wireless LANs," *Electronics Letters*, vol. 40, pp. 915-916, 2004.
- [2] K. Xu, *et al.*, "Effectiveness of RTS/CTS handshake in IEEE 802.11 based ad hoc networks," *Ad Hoc Networks*, vol. 1, pp. 107-123, 2003.
- [3] R. Van Nee, "Breaking the gigabit-per-second barrier with 802.11 ac," *IEEE Wireless Communications*, vol. 18, p. 4, 2011.
- [4] P. Wang and W. Zhuang, "A collision-free MAC scheme for multimedia wireless mesh backbone," in *IEEE ICC*, 2008.
- [5] D. Panigrahi and B. Raman, "TDMA scheduling in long-distance WIFI networks," in *IEEE INFOCOM*, 2009, pp. 2931-2935.
- [6] Z. J. Haas and J. Deng, "Dual busy tone multiple access (DBTMA)-a multiple access control scheme for ad hoc networks," *IEEE Transactions on Communications*, vol. 50, pp. 975-985, 2002.
- [7] P. Wang and W. Zhuang, "An Improved Busy-Tone Solution for Collision Avoidance in Wireless Ad Hoc Networks," in *IEEE ICC*, Istanbul, Turkey, 2006.
- [8] S. Zhang, *et al.*, "Flexible Multi-channel Access and Channel Width Adaptation in Wireless Networks," in *Student poster session, IEEE INFOCOM*, 2013.
- [9] T. Kim, *et al.*, "Distributed semi-synchronous channel coordination for multi-channel wireless networks," *Ad Hoc Networks*, vol. 11, pp. 368-382, 2013.
- [10] C. Chen, *et al.*, "Self-learning collision avoidance for wireless networks," in *IEEE INFOCOM*, 2006.
- [11] J. Barcelo, *et al.*, "Learning-BEB: Avoiding Collisions in WLAN," Eunice Summer School 2008.
- [12] H. Yong, *et al.*, "Semi-Random Backoff: Towards Resource Reservation for Channel Access in Wireless LANs," in *ICNP*, 2009.
- [13] P. Starzetz, *et al.*, "Hashing Backoff: A Collision-Free Wireless Access Method," in *Proceedings of IFIP Networking*, 2009, pp. 11-15.
- [14] J. Barcelo, *et al.*, "On the Distributed Construction of a Collision-Free Schedule in Multi-Hop Packet Radio Networks," *Springer Telecommunication Systems*, pp. 1-14, 2013.
- [15] J. Lee and J. Walrand, "Design and analysis of an asynchronous zero collision MAC protocol," vol. Tech. Rep. UCB/EECS-2007-63, 2007.
- [16] M. Fang, *et al.*, "Decentralised learning MACs for collision-free access in WLANs," *Springer Wireless Networks*, vol. 19, pp. 83-98, May 2013.
- [17] Y. Yi, *et al.*, "MAC scheduling with low overheads by learning neighborhood contention patterns," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1637-1650, 2010.
- [18] P. Djukic and P. Mohapatra, "Soft-TDMAC: A software TDMA-based MAC over commodity

- 802.11 hardware," in *IEEE INFOCOM*, 2009, pp. 1836-1844.
- [19] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM Journal on Applied Mathematics*, vol. 25, pp. 1645-1662, 1990.
- [20] I. Belykh, *et al.*, "Synchronization of bursting neurons: what matters in the network topology," *Physical review letters*, vol. 94, p. 188101, 2005.
- [21] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*: Princeton University Press, 2010.
- [22] Y.-W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE Journal on Selected Areas in Communications*, pp. 1085-1099, May 2005.
- [23] J. Yick, *et al.*, "Wireless sensor network survey," *Computer networks*, vol. 52, pp. 2292-2330, 2008.
- [24] G. Buzsáki and A. Draguhn, "Neuronal oscillations in cortical networks," *Science*, vol. 304, pp. 1926-1929, 2004.
- [25] J. Barcelo, *et al.*, "Modelling a Decentralized Constraint Satisfaction Solver for Collision-Free Channel Access," *CoRR*, vol. abs/1210.3598, 2012.
- [26] I. Rhee, *et al.*, "Z-MAC: a hybrid MAC for wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 16, pp. 511-524, 2008.
- [27] W. Gong and D. Malone, "Addressing slot drift in decentralized collision free access schemes for w lans," *Multiple Access Communications*, pp. 146-157, 2012.
- [28] H. Zhao, *et al.*, "Modeling Intra-flow Contention Problem in Wireless Multi-hop Networks," *IEEE Communications Letters*, vol. 14, pp. 18-20, 2010.
- [29] H. Zhao, *et al.*, "Evaluating the Impact of Network Density, Hidden Nodes and Capture Effect for Throughput Guarantee in Multi-Hop Wireless Networks," *Ad hoc networks*, vol. 11, pp. 54-69, 2013.

## APPENDIX: PROOF OF THEOREM 1

Scenarios with more than two contending nodes can be obtained by combining cases with two contending nodes. For instance, in Fig. 2, Case D can be first seen as Case B when we take Packet 1 and Packet 2 as one super-packet, and after this super-packet avoiding collision to Packet 3, the relationship between Packet 1 and Packet 2 will be either Case A, Case B or Case C. And the cases with two contending nodes can transform to each other within limited iteration times as illustrated in Fig A1, where the denotations along the arrow represent the iteration times needed to complete the transition.

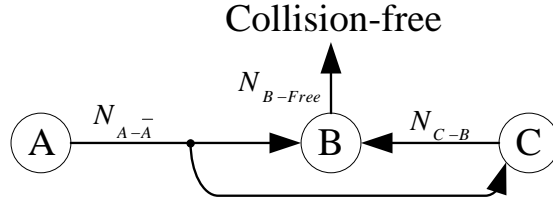


Fig. A1. Transitions of the collision cases

According to our proposal, in most cases (only if Packet 1 and Packet 2 have different adjustment steps, i.e.,  $\alpha_1 l_1 \neq \alpha_2 l_2$ ), Case A will transform to Case B or Case C in just one iteration. Assuming that after  $N_{A-A}$  times of continuous iteration, Case A will leave this state and jump to Case B or Case C, we have  $N_{A-A} \square O(1)$ .

Case C will first transform to Case B after  $N_{C-B}$  iterations, and we have

$$N_{C-B} \leq \left\lceil \frac{|l_2 - l_1|}{\alpha_1 l_1 + \alpha_2 l_2} \right\rceil \leq \frac{|l_2 - l_1|}{\alpha_1 l_1 + \alpha_2 l_2} + 1,$$

where  $\lceil \cdot \rceil$  is the ceil function. Similarly, denoting  $\delta$  as the original phase difference between the two packets in Case B, we have the iteration times from Case B to collision-free state as

$$N_{B-Free} = \left\lceil \frac{\delta}{\alpha_1 l_1 + \alpha_2 l_2} \right\rceil \leq \frac{\min(l_1, l_2)}{\alpha_1 l_1 + \alpha_2 l_2} + 1$$

Therefore, in two-packet scenarios, the iteration times to achieve collision-free state are at most:

$$N_2 = N_{A-A} + N_{C-B} + N_{B-Free} = N_{A-A} + \frac{\max(l_1, l_2)}{\alpha_1 l_1 + \alpha_2 l_2} + 2 \square O(1) \quad (\text{a1})$$

And if there are  $n$  nodes in the network, the worst case is that the collided packets will get apart one by one, thus the total number is up bounded by  $N_n$ ,

$$N_n = (n - 1)(N_2 + N_r) \square O(n) \quad (\text{a2})$$

where  $N_r$  represents the possible repeating actions that two collision-free packets become collided because of other packets' adjustments, and  $N_r \approx N_2 \square O(1)$ .

Therefore,  $n$  nodes whose dynamics are governed by E-MAC algorithm will converge to a collision-free state in  $O(n)$  iterations. ■