

Article

Adaptive Strategy to Change Firing Phases of Collided Nodes in Extended-Desync TDMA-Based MANETs

Cheol-Woong Lee ¹, Gyu-Min Lee ² and Byeong-Hee Roh ^{1,2,*}¹ Department of AI Convergence Network, Ajou University, Suwon 16499, Korea; cjfdnd369@ajou.ac.kr² Department of Computer Engineering, Ajou University, Suwon 16499, Korea; mybrand@ajou.ac.kr

* Correspondence: bhroh@ajou.ac.kr; Tel.: +82-31-219-1601

Abstract: As a multi-hop extension of the desynchronization-based TDMA (Desync-TDMA), the extended Desync-TDMA (Ext-Desync) with self-adapting property is proposed to overcome the limitations of existing CSMA/CA and dynamic TDMA-based schemes for Mobile Ad-hoc Networks (MANETs). However, existing studies overlooked the potential problem of firing message collisions caused by node movements, leading to the severe degradation of MANET networking performance. In this paper, we derive a mathematical model to evaluate the problem due to collisions of firing messages for moving nodes. With the derived model, we propose a method for a collided node to determine whether it changes its firing phase or not, adaptively in a distributed manner, by considering both the collision situation and the slot utilization. The comparative analysis between the proposed method and existing representative ones is also presented for various networking features. The performances of the proposed method are compared with CSMA/CA as well as other existing Ext-Desync-based schemes. The numerical results show that the proposed method achieved much faster resolution and higher slot utilization in collision situations than other Ext-Desync-based schemes. In addition, we also show that the proposed method outperformed the comparable methods, including CSMA/CA, in terms of packet delivery ratios and end-to-end delays.

Keywords: bio-inspired; Extended-Desync TDMA; Medium Access Control (MAC); Wireless Sensor Network (WSN); Mobile Ad-hoc Network (MANET)



Citation: Lee, C.-W.; Lee, G.-M.; Roh, B.-H. Adaptive Strategy to Change Firing Phases of Collided Nodes in Extended-Desync TDMA-Based MANETs. *Sensors* **2021**, *21*, 6776. <https://doi.org/10.3390/s21206776>

Academic Editor: Francisco J. Martinez

Received: 24 August 2021

Accepted: 8 October 2021

Published: 12 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mobile Ad-hoc Networks (MANETs) have features to enable dynamic network configuration and to support data delivery between mobile nodes without infrastructure, as in tactical or disaster environments [1,2]. Since multiple mobile nodes share channels, Medium Access Control (MAC) protocols play key roles in managing and operating MANETs. Most studies have adopted the Carrier Sense Multiple Access (CSMA) or the Time Division Multiple Access (TDMA)-based schemes as MAC protocols for MANETs [3].

Schemes based on CSMA employed an RTS/CTS mechanism to solve the hidden node problem in MANETs, which causes high overhead and makes it difficult to provide stable performance in a congested environment with a large amount of traffic [4]. For TDMA-based MAC protocols, which have been widely studied for MANET environments [5], the synchronization of the slot time among nodes is required, but it is still known as one of the most challenging tasks to be solved in the networking environments where nodes are moving [6]. The use of Non-Orthogonal Multiple Access (NOMA) for MANETs was proposed in [7]; however, it requires Base Stations (BSs) to relay communications among nodes.

As an alternative for wireless MAC protocols, biologically inspired (bio-inspired) approaches, which model the collective behavior of various species' ecosystems, have been studied [8,9]. The desynchronization-based TDMA (Desync-TDMA) [10,11], inspired by the firefly's habits, has been proposed as one of the bio-inspired MAC protocols for full-mesh Wireless Sensor Networks (WSNs).

Desync-TDMA does not require any central agent; instead, each node in a Desync-TDMA network sends firing messages periodically and allocates its slot in a distributed manner considering its and other nodes' firing times. Desync-TDMA has been widely adopted and modified in many studies to enhance decentralized reservation-based scheduling and resource management schemes [12–24]. Since the studies related to Desync-TDMA have focused on single-hop wireless network environments with static nodes, they are not suitable to apply directly to multi-hop network environments.

To support multi-hop delivery of information, extended Desync-TDMA (Ext-Desync) [25] has been proposed by extending Desync-TDMA. Each node broadcasts firing messages including its one-hop neighbor list and their firing times. By referring to firing messages from other nodes, all nodes can know their two-hop neighbors and their relative firing times, which can solve the hidden terminal problem in multi-hop wireless networking. There have been also numerous studies to enhance multi-hop wireless networking performances based on Desync-TDMA or Ext-Desync [26–34].

However, since nodes are moving in MANET, collisions among firing messages may occur due to the hidden terminal problem's occurrence by the moving nodes. In [25], nodes that detect collisions change their firing times with a certain fixed probability to solve the problem. As the mobility of nodes increases, the possibility of collisions increases, and nodes may change their firing times frequently. The frequent change of firing times due to such collisions may cause significant degradations of Ext-Desync networking performances [35].

As we will explain in detail in Section 3.1, the collision of firing message affects the degradation of slot utilization performances, resulting in the degradation of the packet delivery performances. Existing studies overlooked this problem in depth in their schemes, although it may seriously impact network performances.

This paper proposes an effective method for a collided node to determine whether to change its firing phase or not in order to resolve the collision situation optimally. The determination is carried out by considering both the collision situation and the slot utilization. The main contributions of this paper are as follows:

- We deal with the potential and critical problem that Ext-Desync-based schemes have when operated in MANETs, which has been overlooked in other studies. The problem definition in detail and its effect on the networking performances are illustrated in Section 3.1.
- We derive an analytical model to evaluate the problem mathematically. Then, we also derive an optimal criterion for the probability that a collided node will change its firing phase in the following next period after it acknowledges the collision.
- With the criterion, a method for a collided node to determine whether it changes its firing phase or not in a distributed manner is proposed.
- The performances of the proposed method are compared with existing Ext-Desync-based TDMA schemes and CSMA/CA.

The rest of the paper is organized as follows. In Section 2, overview Desync-TDMA and Ext-Desync. In Section 3, the problem that Ext-Desync schemes have in MANETs is illustrated. Then, the proposed method is explained. The numerical results are presented in Section 4. Studies that extend Desync-TDMA and Ext-Desync are described in Section 5. We also compare this work and the related work from a functional point of view in this section. Finally, the paper is concluded in Section 6.

2. Background

Here, an overview of the methods underlying the proposed method is illustrated. In Table 1, the main variables used for the illustration are listed.

Table 1. Variables for the illustration of Desync-TDMA and Ext-Desync.

Variable	Description
T	The cyclic period of Desync- and Ext-Desync-based schemes
N	The number of nodes in the network
$\Phi_i(t)$	The firing phase of node i in t -th T cycle
$\Phi_{i,-}(t)$	The firing phase of other node just before $\Phi_i(t)$
$\Phi_{i,+}(t)$	The firing phase of other node just after $\Phi_i(t)$
$S_{i,st}(t)$	The start time of the node i in t -th period
$S_{i,ed}(t)$	The end time of the node i in t -th period
α	constant indicating how $\Phi_i(t+1)$ is calculated from the average of $\Phi_{i,-}(t)$ and $\Phi_{i,+}(t)$
$\mathbf{L}_i^h(t)$	The list of firing phase information of h -hop neighbors managed by node i in the t -th period
$a_{i,j}^h(t)$	Identifier of j -th h -hop neighbor node of node i in t -th period
$\Delta_{i,j}^h(t)$	The relative firing phase with node i of j -th h -hop neighbor node in t -th period
$N_i^h(t)$	The number of h -hop neighbor node of node i in t -th period
$\Phi_{i,j}^h(t)$	The firing phase of node $a_{i,j}^h(t)$
G	The length of the timelag of Ext-Desync-based schemes

2.1. Desync-TDMA

Desync-TDMA [10,11] is a decentralized and distributed TDMA protocol inspired by the firefly's habits for single-hop wireless sensor nodes. In Desync TDMA, the time is managed by a constant cyclic period of T . Each node broadcasts a control packet (*firing message*) once at a predetermined time (*firing phase*) in every T cycle.

Let N and $\Phi_i(t)$ be the number of nodes and the firing phase of node i in the t -th T cycle, respectively, where $i = 1, 2, \dots, N$ and $t = 1, 2, \dots$. Let $\Phi_{i,-}(t)$ and $\Phi_{i,+}(t)$ be the firing phases of other nodes just before and after $\Phi_i(t)$ in the t -th T cycle, respectively. Then, the firing phase of node i at the next $(t+1)$ -th T cycle, $\Phi_i(t+1)$, is calculated as

$$\Phi_i(t+1) = T + (1 - \alpha) \times \Phi_i(t) + \alpha \times (\Phi_{i,-}(t) + \Phi_{i,+}(t))/2, \quad (1)$$

where α ($0 < \alpha \leq 1$) is a constant indicating how $\Phi_i(t+1)$ is closely calculated from the average of $\Phi_{i,-}(t)$ and $\Phi_{i,+}(t)$.

The slot that node i can transmit its data in $(t+1)$ -th T cycle is calculated as follows.

$$S_{i,st}(t+1) = T + (\Phi_{i,-}(t) + \Phi_i(t))/2, \quad (2)$$

$$S_{i,ed}(t+1) = T + (\Phi_i(t) + \Phi_{i,+}(t))/2, \quad (3)$$

where $S_{i,st}(t+1)$ and $S_{i,ed}(t+1)$ denote the start and end times of the slot, respectively.

Figure 1 shows an example of Desync-TDMA operation for $N = 5$. Figure 1a shows the firing phases of all nodes in the t -th period, Φ_i ($i = 1, 2, \dots, 5$). Each node can calculate its firing phases and slot in the following $(t+1)$ -th period with the firing phases received from other nodes. Figure 1b shows the process of adjusting the firing phase in the following $(t+1)$ -th period from the perspective of node 4. In this example, the firing phases immediately before and after $\Phi_4(t)$ are those by nodes 3 and 5, i.e., $\Phi_{4,-}(t) = \Phi_3(t)$ and $\Phi_{4,+}(t) = \Phi_5(t)$.

Then, node 4 calculates its firing phase and slot in the $(t+1)$ -th period, $\Phi_5(t+1)$, $S_{5,st}(t+1)$ and $S_{5,ed}(t+1)$, respectively, using Equations (1)–(3). Similarly, other nodes also calculate their firing phases and slots in the $(t+1)$ -th period, as shown in Figure 1c. As this process is repeated, it reaches the convergence state, as shown in Figure 1d, in which the firing phases and slots of nodes are evenly distributed within every T cycle.

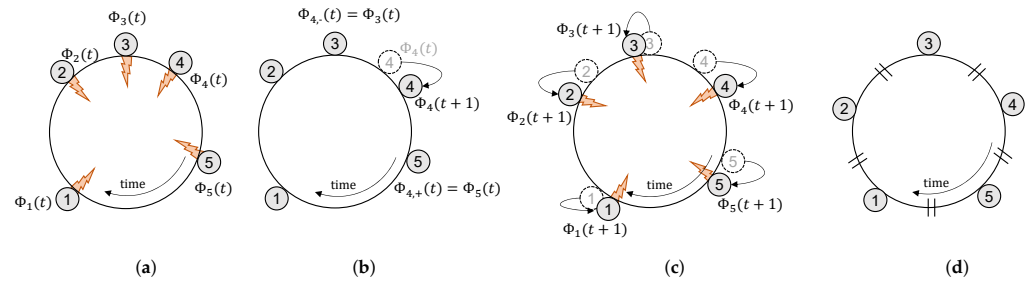


Figure 1. An example of the Desync-TDMA process for $N = 5$: (a) firings in the t -th period (b) node 4's perspective in the t -th period (c) firings in $(t + 1)$ -th period, and (d) converged period.

2.2. Ext-Desync

Ext-Desync [25] is a multihop extension of Desync-TDMA. Ext-Desync solves the hidden terminal problem by making all nodes simply know all their two-hop neighbors and their relative firing times as follows: Node i in an Ext-Desync network has its own identifier (a_i) and maintains the set of its h -hop neighbors acknowledged in the t -th period as follows:

$$L_i^h(t) = \{(a_{i,j}^h(t), \Delta_{i,j}^h(t)) | j = 1, 2, \dots, N_i^h(t)\}, \quad (4)$$

where $h = 1, 2$, $N_i^h(t)$ denotes the number of h -hop neighbor nodes, and $a_{i,j}^h(t)$ and $\Delta_{i,j}^h(t)$ are the identifier and the relative firing phase with node i of j -th h -hop neighbor node, respectively. Node i can obtain $\Delta_{i,j}^h(t)$ as

$$\Delta_{i,j}^h(t) = \Phi_{i,j}^h(t) - \Phi_i(t), \quad (5)$$

where $\Phi_{i,j}^h(t)$ denotes the firing phase of node $a_{i,j}^h(t)$. It is noted that $(a_i, 0) \in L_i^1(t)$ and $L_i^1(t) \subseteq L_i^2(t)$.

During t -th period, node i broadcasts a firing message containing $L_i^1(t-1)$ at its firing phase and receives firing messages from its 1-hop neighbor nodes. Then, $L_i^1(t)$ is updated by referring to the nodes' firing messages that node i received. The nodes listed in the 1-hop neighbors' firing messages, but not included in $L_i^1(t)$, become node i 's 2-hop neighbors, and they are updated in $L_i^2(t)$.

Let $\Delta_{i,-}(t)$ and $\Delta_{i,+}(t)$ be the relative firing phases just before and after 0 in $L_i^2(t)$. Then, the firing phase and the slot of node i at $(t + 1)$ -th period are determined by

$$\Phi_i(t+1) = T + \Phi_i(t) + \alpha \times (\Delta_{i,-}(t) + \Delta_{i,+}(t))/2, \quad (6)$$

$$S_{i,st}(t+1) = T + \Phi_i(t) + (\Delta_{i,-}(t) + G)/2, \quad (7)$$

$$S_{i,ed}(t+1) = T + \Phi_i(t) + (\Delta_{i,+}(t) - G)/2, \quad (8)$$

where G denotes the time-lag between adjacent slots for the firing message of a new node or a node changing its firing phase after collision not to violate slots of other nodes.

Due to the hidden terminal problem and the nodes' movements, firing messages may collide with others. If a node does not know that the collision has occurred, the node continues to transmit packets in the next period, and it results in packet losses or retransmissions due to the losses, which degrades the packet delivery performances.

To solve the problem, Ext-Desync provides the method for node i to acknowledge the collision by monitoring the firing message from other nodes j within its one-hop range whether its identifier is not included in L_j^1 ($j \neq i$). When the node recognizes the collision of its firing message, it may decide whether it changes its firing phase in the next period or not according to a certain probability (e.g., 0.5). The fixed probability-based decision policy may cause the degradation of networking performances, which are discussed in Section 3.1.

3. Extended-Desync TDMA with Optimal Criterion to Change the Firing Phase

3.1. Problem Definition

As mentioned in Section 2.2, when Ext-Desync is operated in MANET environments, collisions of firing messages may occur due to node mobility. Node i (a_i) can detect the collision of its firing message by monitoring the firing message from other node j within its one-hop range during the next periods whether its identifier is included in L_j^1 ($j \neq i$).

Figure 2 shows an example scenario to illustrate this situation more clearly. There are 12 nodes denoted a_i ($i = 1, 2, \dots, 12$), and a_8 is moving while other nodes are not. Figure 2a,b show the network topologies in $(t-1)$ -th and t -th periods before and after a_8 moves, respectively.

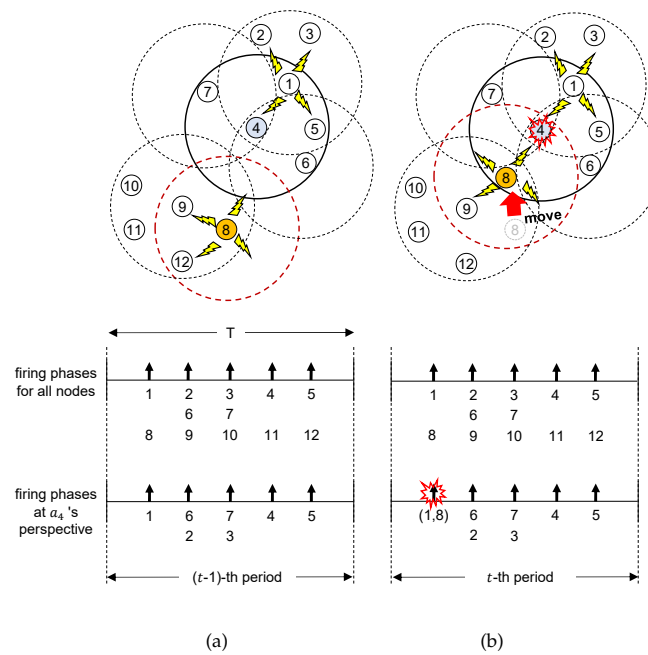


Figure 2. Example topology and timelines for firing messages: (a) before and (b) after a_8 moves.

In Figure 2a, example topology and timelines of the firing messages from both all nodes and a_4 's perspective at $(t-1)$ -th period before a_8 moves are shown. Since nodes a_2 , a_6 , and a_9 are three-hop distances from each other, collision does not occur though they have the same firing phase [36,37]. Node a_4 has $L_4^1(t-1) = \{a_1, a_4, a_5, a_6, a_7\}$ and $L_4^2(t-1) = \{L_4^1(t-1), a_2, a_3\}$.

When a_8 moves into the one-hop range of a_4 at t -th period as shown in Figure 2b, since a_8 's firing phase overlap with a_1 's one, firing messages from a_8 and a_1 are collided at a_4 . a_4 makes its firing message $L_4^1(t) \neq \{a_1, a_8\}$. After a_8 and a_1 receive the firing message from a_4 , they can acknowledge the collision of their firing messages.

After the collision is detected by a_1 and a_8 during t -th period, the possible cases where they can resolve the collision by changing their firing phases are shown in Figure 3. Figure 3a shows one case where both a_1 and a_8 change their firing phases. Nodes a_1 and a_8 do not transmit any messages, but listen to the firing messages from neighbors during $(t+1)$ -th period. No node can utilize the slot allocated for a_1 and a_8 previously, which are denoted by the dotted line, and thus the slot is wasted. They calculate timelags, randomly choose one among them, and proceed with the join process to determine their slots in the network during $(t+2)$ -th period as described in Section 2.2. Then, they can transmit data on their determined slots from $(t+3)$ -th period.

Figure 3b shows the case where a_1 changes its firing phase while a_8 does not. In this case, a_8 can continuously utilize its slot previously assigned, no slot is wasted, unlike Figure 3a. However, after the listen and join processes, a_1 can transmit its data from $(t+3)$ -

th period. The case where only a_8 changes its firing phase is shown in Figure 3c. In this case, the opposite situation appears for a_1 and a_8 as in Figure 3b.

Figure 3d shows the case where both a_1 and a_8 do not change their firing phases. Then, since their firing messages collide again during $(t + 1)$ -th period, they should repeat one of Figure 3a–d from $(t + 2)$ -th period, and the slot is wasted.

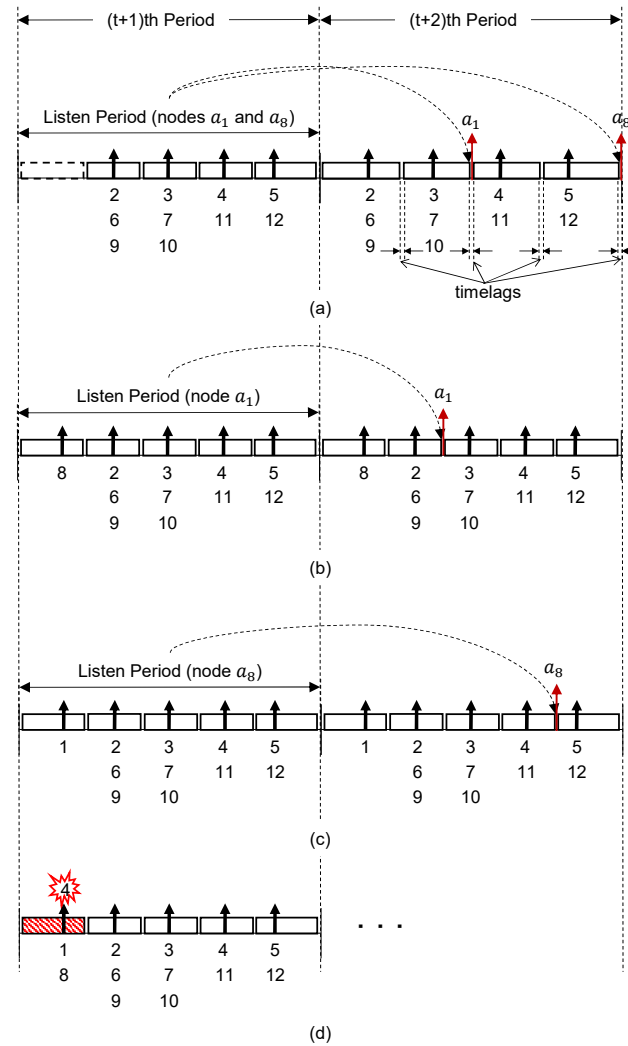


Figure 3. Example process to change the firing phases just after Figure 2b by (a) both a_1 and a_8 , (b) a_1 only, (c) a_8 only, and (d) none.

3.2. Criterion of Firing Phase Change to Maximize Slot Utilization

The frequent change of firing phases due to collisions may cause significant degradations of Ext-Desync networking performances, especially from the viewpoints of the slot utilization as described in Section 3.1. The authors in [25] suggested that nodes that detect the collision may change their firing phases with a certain probability of 50% as a trade-off between reliability and latency. However, when the probability is fixed like that, there always exists the possibility that collided nodes change their firing phases simultaneously to equal phases, which results in collisions again and the waste of slots as shown in Figure 3.

In this Section, to overcome the problem by the fixed probability model as in [25], we derive an effective criterion model for collided nodes to determine the probabilities that they can change their firing phases dynamically by considering collision situations in order to maximize the slot utilization. The variables to explain the model are illustrated in Table 2.

For the convenience of the model derivation, it is assumed that a_i acknowledges its firing message collision in the 0-th period, i.e., $t = 0$. As shown in Figure 3a, when a_i changes its firing phase, since it can obtain the slot for data transmission in the following second period after the change of the firing phase, the slot is wasted for the periods. On the other hand, if a_i does not change its firing phase, it can utilize the slot assigned to it continuously, as shown in Figure 3b,c.

Table 2. Description of variables.

Variable	Description
a_i	Node that acknowledged the collision of its firing message in the 0-th period
$N_i^1(n)$	Number of one-hop neighbors of a_i in the n -th period ($n \geq 0$)
$\hat{C}_i(n)$	Number of hidden nodes that cause the collision to a_i 's firing message in the n -th period ($n \geq 0$)
$c_{i,j}(n)$	Nodes of hidden to a_i in the n -th period ($j = 1, 2, \dots, \hat{C}_i(n)$)
$p_{c,i}(n)$	Probability that a_i changes its firing phase in the n -th period ($n \geq 0$)
$p_{nc,i}(n)$	Probability that a_i does not change its firing phase until the n -th period ($p_{nc,i}(n) = 1 - p_{c,i}(n)$) ($n \geq 0$)
$\tilde{s}_{c,i}^*(n)$	Expected slot size for a_i when it changes its firing phase before n -th period ($n \geq 1$)
$\tilde{s}_{nc,i}^*(n)$	Expected slot size for a_i when it does not change its firing phase until the n -th period ($n \geq 1$)
$\hat{p}_{c,j}(n)$	Probability estimated by a_i that a node $c_{i,j}$, caused the collision of a_i 's firing message in the n -th period, will change its firing phase in the n -th period, where $j = 1, \dots, \hat{C}_i(n)$ ($n \geq 0$)
$s_i(n)$	Amount of slot size available to a_i up to the n -th period ($n \geq 0$)
$s_{i,col}(n)$	Amount of slot size that a_i fails to transmit data due to the collision up to the n -th period ($n \geq 0$)
$s_{i,suc}(n)$	Amount of slot size that a_i succeeds to transmit data up to the n -th period, $s_{i,suc}(n) = s_i(n) - s_{i,col}(n)$ ($n \geq 0$)

Let $p_{c,i}(k)$ be the probability that a_i changes its firing phase in the k -th period. $p_{nc,i}(k) = 1 - p_{c,i}(k)$. Then, the expected slot size for a_i when it changes its firing phase before n -th period ($n \geq 1$) is written as

$$\tilde{s}_{c,i}^*(n) = \sum_{k=1}^n \{(n-k-2)s_{c,i}(k)p_{c,i}(k) \prod_{m=1}^k p_{nc,i}(m-1)\}, \quad (9)$$

where $s_{c,i}(k)$ denotes the slot size for a_i , when it changes its firing phase in the k -th period and $p_{c,i}(0)=0$.

We also have the expected slot size for a_i when it does not change its firing phase until the n -th period as

$$\tilde{s}_{nc,i}^*(n) = ns_{c,i}(0) \prod_{k=1}^n p_{nc,i}(k). \quad (10)$$

From Equations (9) and (10), the expected value of $s_i(n)$ is written as

$$E[s_i(n)] = \tilde{s}_{c,i}^*(n) + \tilde{s}_{nc,i}^*(n). \quad (11)$$

The collision may occur by the hidden terminal problem, for example, between a_1 and a_8 shown in Figure 2b. Since firing messages are broadcast, the collision of other nodes' firing messages can be recognized by monitoring L_*^1 from others in the next period. Let $\hat{C}_i(n)$ be the number of those hidden nodes that cause the collision to a_i 's firing message in the n -th period. For example, nodes a_1 and a_8 are hidden terminals each other in Figure 2b.

Accordingly, for a_1 , we have $\hat{C}_1(0)=1$ and $c_{1,1} = a_8$. On the other hand, $\hat{C}_8(0) = 1$ and $c_{8,1} = a_1$ for a_8 . With $\hat{C}_i(n)$, the expected value of $s_{i,col}(n)$ can be calculated by

$$E[s_{i,col}(n)] = \frac{\hat{C}_i(n)}{N_i^1(n)} \tilde{p}_{nc,i}^*(n). \quad (12)$$

$\hat{C}_i(n)$ s ($n \geq 1$) can be estimated consecutively using $\hat{C}_i(0)$ and the firing messages received from its one-hop neighbors in the n -th period as follows:

$$\hat{C}_i(n) = \sum_{j=0}^{\hat{C}_i(n-1)} \left\{ \binom{\hat{C}_i(n-1)}{j} \cdot j \cdot \{1 - \hat{p}_{c,i}(n)\}^j \cdot \hat{p}_{c,i}(n)^{\hat{C}_i(n-1)-j} \right\}. \quad (13)$$

Using Equations (11) and (12), the expected value of $s_{i,suc}(n)$ is calculated as

$$E[s_{i,suc}(n)] = E[s_i(n)] - E[s_{i,col}(n)]. \quad (14)$$

Assuming that a_i acknowledges the collision of its firing message in the 0-th period and changes the firing phase with probability p ($0 \leq p \leq 1$) in the next 1-st period, let define $\overline{ES}_i(p, n)$ as the expected amount of total data successfully transmitted up to the n -th period. With Equations (11) to (14), it can be written as

$$\begin{aligned} \overline{ES}_i(p, n) &= \sum_{k=1}^n E[s_{i,suc}(k)] \\ &= (n-1) + (1-p)^n - \sum_{k=1}^n CN_i(k)(1-p)^k, \end{aligned} \quad (15)$$

where $p = p_{c,i}(1)$ and $CN_i(k) = \hat{C}_i(k)/N_i^1(k)$.

Then, we have the criterion to determine the probability that a_i changes its firing phase to maximize $\overline{ES}_i(p, n)$ as following:

Proposition 1. When a_i acknowledges first the collision of its firing message in a period, $\overline{ES}_i(p, n)$ is maximized when it changes its firing phase in the next period with the probability of $p_{c,i}(1) = 0$ or 1.

Proof of Proposition 1. From Equation (15), we have $\overline{ES}_i(0, n)$ and $\overline{ES}_i(1, n)$ when $p_{c,i}(1) = 0$ and 1, respectively, as follows

$$\overline{ES}_i(0, n) = n - \sum_{k=1}^n CN_i(k), \quad (16)$$

$$\overline{ES}_i(1, n) = n - 1. \quad (17)$$

To validate the Proposition that $\overline{ES}_i(0, n)$ or $\overline{ES}_i(1, n)$ are always greater than $\overline{ES}_i(p', n)$ for $0 < p' < 1$, we consider the following two cases.

Case 1. $\overline{ES}_i(0, n) > \overline{ES}_i(1, n)$:

In this case, from Equations (16) and (17), we have $\sum_{k=1}^n CN_i(k) < 1$. For $0 < p' < 1$, the inequity of $\sum_{k=1}^n CN_i(k)\{1 - (1-p')^k\} < \{1 - (1-p')^n\}$ can be obtained. With the

inequalities and Equation (15), we have the relationship between $\overline{ES}_i(0, n)$ and $\overline{ES}_i(p', n)$ as follows

$$\begin{aligned} \overline{ES}_i(0, n) - \overline{ES}_i(p', n) &= \{n - \sum_{k=1}^n CN_i(k)\} - \{(n-1) + (1-p')^n - \sum_{k=1}^n CN_i(k)(1-p')^k\} \\ &= 1 - (1-p')^n - \sum_{k=1}^n CN_i(k)\{1 - (1-p')^k\} > 0 \end{aligned} \quad (18)$$

Case 2. $\overline{ES}_i(0, n) \leq \overline{ES}_i(1, n)$:

This case is equivalent to $\sum_{k=1}^n CN_i(k) \geq 1$. For $0 < p' < 1$, we have $\sum_{k=1}^n CN_i(k)(1-p')^k > (1-p')^n$. With the inequalities and Equation (15), the following relationship between $\overline{ES}_i(1, n)$ and $\overline{ES}_i(p', n)$ holds for this case.

$$\begin{aligned} \overline{ES}_i(1, n) - \overline{ES}_i(p', n) &= (n-1) - \{(n-1) + (1-p')^n - \sum_{k=1}^n CN_i(t, k)(1-p')^k\} \\ &= -(1-p')^n + \sum_{k=1}^n CN_i(t, k)\{1 - (1-p')^k\} > 0 \end{aligned} \quad (19)$$

Cases 1 and 2 indicate that either of $\overline{ES}_i(0, n)$ or $\overline{ES}_i(1, n)$ is always larger than $\overline{ES}_i(p', n)$ for $0 < p' < 1$. In other words, $\overline{ES}_i(p, n)$ is maximized when a_i changes its firing phase with the probability of 0 or 1, not determining probabilistically between 0 and 1. \square

To show the effectiveness of the *Proposition*, the values of $\overline{ES}_i(p, n)$ for various $p_{c,i}(1)$ and $\hat{p}_{c,j}(1)$ when $n = 5$ and $N_i^1(0) = 10$ are shown in Figure 4. For the convenience, it is assumed that the slot size of each node is set to 1, i.e., $s_{c,i}(k) = 1$, for $\forall i$ and $\forall k$. As shown in Figure 4, $\overline{ES}_i(p, n)$ shows different aspects for each $p_{c,i}(1)$ and $\hat{p}_{c,j}(1)$ and is maximized when $p_{c,i}(1)$ is 1 or 0, and it appears differently depending on $\hat{p}_{c,j}(1)$.

That is, when $\hat{p}_{c,j}(1)$ s are lower than around 0.2, $\overline{ES}_i(p, n)$ s show the increasing pattern as $p_{c,i}(1)$ increases, and thus $\overline{ES}_i(p, n)$ s become maximum at $p_{c,i}(1) = 1$. On the other hand, when $\hat{p}_{c,j}(1)$ s are greater than around 0.2, $\overline{ES}_i(p, n)$ s show a decreasing pattern as $\hat{p}_{c,j}(1)$ increases, which results in the maximums of $\hat{p}_{c,j}(t, 1)$ s at $p_{c,i}(1) = 0$. Likewise, ES becomes its maximum at $p_{c,i}(t, 1) = 0$ or 1 depending on $\hat{p}_{c,j}(1)$. As mentioned earlier, it is not possible for a_i to know $\hat{p}_{c,j}(1)$. The method to estimate $\hat{p}_{c,j}(1)$ is explained in Section 3.3 as one of our proposed methods in this paper.

With the *Proposition*, we have the optimal criterion to determine the probability for changing its firing phase as follows

$$p_{c,i}(1) = \begin{cases} 0, & \overline{ES}_i(0, n) > \overline{ES}_i(1, n) \\ 1, & \text{otherwise} \end{cases} \quad (20)$$

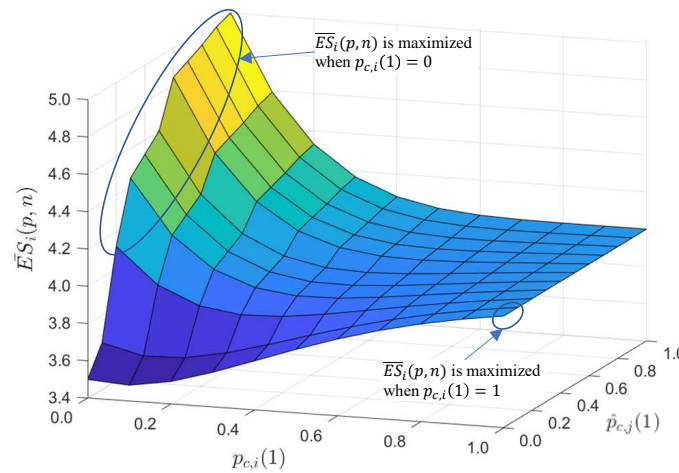


Figure 4. An example results of $\overline{ES}_i(p, n)$ with varying $p_{c,i}(1)$ and $\hat{p}_{c,j}(1)$ (when $n = 5$, $N_i^1(0) = 10$, and $\hat{C}_i(0) = 3$).

3.3. Ext-Desync TDMA with Optimal Criterion of Firing Phase Change

Here, algorithms for determining the optimal criterion of firing phase change illustrated in the previous section are explained. For better understanding, the relationships among the functions used in the algorithms and their operational processes concerning cycle periods are shown in Figure 5. Since the proposed method operates asynchronously, like Ext-Desync, the algorithms also operate asynchronously. A detailed explanation of the operation is explained later. ListenMode(T) in Algorithm 1 is executed every period, and The algorithm's operation starts from ListenMode(T) in Algorithm 1 every period by receiving firing messages from nodes and detecting whether its firing message collided or not.

The period executing ListenMode(T) is denoted as the 0-th period for convenience. The blue line and text in Figure 5 are the main functions used in Algorithm 1 showing their interrelationship on operations according to subsequent periods following the 0-th period. Among Algorithm 1 functions, OnSlotStart() operation depends on the result of DeterminePhaseChange() function performed in Algorithm 2. The red line and text indicate the process of Algorithm 1 according to the result of Algorithm 2.

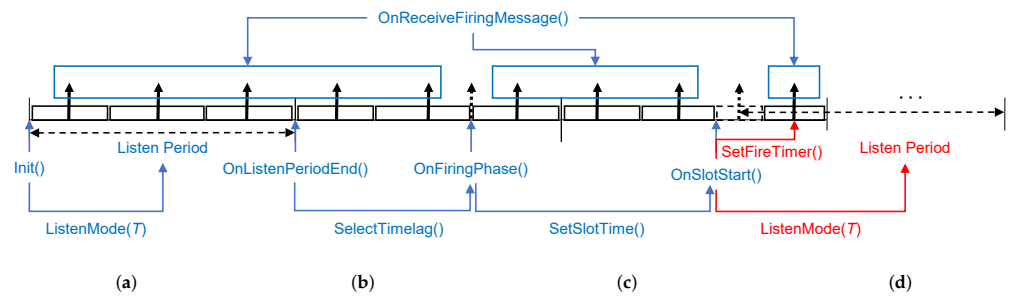


Figure 5. An example of the Proposed Ext-Desync process from the standpoint of the main functions: (a) 0th period (b) 1st period (c) 2nd period (d) after 2nd period.

The overall procedure of Ext-Desync TDMA with the adaptive change of the firing phase from the viewpoints of a node a_i ($i = 1, 2, \dots$) in the network is shown in Algorithm 1 with five main functions such as Init(), OnListenPeriodEnd(), OnReceiveFiringMessage(), OnFiringPhase(), and OnSlotStart(). Since Algorithm 1 operates repeatedly, the period index is expressed to 0. Notations of $\bullet(0)$ and $\bullet(1)$ refer to the parameters of the current and the following periods, respectively.

Init() is done when a_i is initially joining the network and starts to execute the adaptive firing phase change mode.

OnListenPeriodEnd() is called at the end of the listen period, i.e., at the end of ListenMode() in Init(), which is mentioned in Section 3.1. $\hat{p}_{c,i}(1)$ is initialized by $\hat{p}_{c,0}$, which is one of the system parameters and used in DeterminePhaseChange() of Algorithm 2. The firing phase in the following period, $\Phi_i(1)$, is randomly selected by SelectTimelag() as illustrated in Figure 3.

Algorithm 1: Proposed Ext-Desync TDMA Procedure with Optimal Criterion of Firing Phase Change.

Input: $T, \alpha, G, \hat{p}_{c,0}, \hat{p}_{nc,0}$

```

1 Procedure Init ()
2   ListenMode( $T$ );
3   return;
4 Procedure OnListenPeriodEnd ()
5    $\hat{p}_{c,i}(1) \leftarrow \hat{p}_{c,0}$ ;
6    $\Phi_i(1) \leftarrow \text{SelectTimelag}(L_i^1(0), L_i^2(0))$ ;
7   SetFireTimer( $\Phi_i(1)$ );
8   return;
9 Procedure OnReceiveFiringMessage ()
10   $N_i^1(0) \leftarrow N_i^1(0) + 1$ ;
11  UpdatePhaseLists( $L_i^1(t), L_i^2(0), a_f, L_f^1(0)$ );
12  DetectCollision( $C_i(0), a_i, L_f^1(0)$ );
13  return;
14 Procedure DetectCollision ( $C_i(0), a_i, L_f^1(0)$ )
15  if  $a_i \notin L_f^1(0)$  then
16     $C_i(0) \leftarrow C_i(0) + 1$ ;
17  end
18  return;
19 Procedure OnFiringPhase ()
20  SendFiringMessage( $a_i, L_i^1(0)$ );
21   $N_i^1(1) \leftarrow 0, N_{i,col}(1) \leftarrow 0$ ;
22   $S_{i,st}(1) \leftarrow T + \Phi_i(0) + (\Delta_{i,-}(0) + G)/2$ ;
23   $S_{i,ed}(1) \leftarrow T + \Phi_i(0) + (\Delta_{i,+}(0) - G)/2$ ;
24  SetSlotTime( $S_{i,st}(1), S_{i,ed}(1)$ );
25  return;
26 Procedure OnSlotStart ()
27   $p_{c,i}(1) \leftarrow \text{DeterminePhaseChange}(N_i^1(0), C_i(0), \hat{p}_{c,i}(1))$ ;
28  if  $p_{c,i}(1) = 1$  then
29    Init ();
30  else
31     $\Phi_i(1) \leftarrow T + \Phi_i(0) +$ 
32       $\alpha \times (\Delta_{i,-}(0) + \Delta_{i,+}(0))/2$ ;
33    SetFireTimer( $\Phi_i(1)$ );
34  end
35  return;

```

OnReceiveFiringMessage() is called whenever a_i receives a firing message from an one-hop neighbor denoted by a_f with $L_f^1(0)$. Then, $N_i^1(0)$ is increased by one, and $L_i^1(0)$ and $L_i^2(0)$ are updated using a_f and $L_f^1(0)$ by UpdatePhaseLists(). To update $C_i(0)$, it calls DetectCollision(), in which it is increased by one when a_i is not included in $L_f^1(0)$.

In OnFiringPhase(), a_i sends its firing message, initializes N_i^1 and C_i to 0. Then, it calculates and allocates its slot in the following period using Equations (7) and (8).

OnSlotStart() is invoked at the slot start time set by SetSlotTime(). According to the decision on $p_{c,i}(1)$ by DeterminePhaseChange(), which is explained in Algorithm 2, it enters the listen period to reconnect with other devices in the network if $p_{c,i}(1)$ is 1, as shown in Figure 3 in Section 3.1. On the other hand, it sets the next firing phase.

Algorithm 2: Determination of Firing Phase Change with Optimal Criterion.

Input : $N_i^1(t)$, $C_i(0)$, and $\hat{p}_{c,i}(1)$
Output: $p_{c,i}(1)$

```

1 Procedure DeterminePhaseChange ( $N_i^1(0)$ ,  $C_i(0)$ ,  $\hat{p}_{c,i}(1)$ )
2    $\hat{C}_i(1) \leftarrow \sum_{l=0}^{C_i(0)} (C_i^{(0)}) \cdot l \cdot (1 - \hat{p}_{c,i}(1))^l$ 
3      $\cdot \hat{p}_{c,i}(1)^{C_i(0)-l};$ 
4   for  $m \leftarrow 2$  to the  $n$  do
5      $\hat{C}_i(m) \leftarrow \sum_{l=0}^{[\hat{C}_i(m-1)]} ([\hat{C}_i(m-1)]) \cdot l$ 
6        $\cdot (1 - \hat{p}_{c,i}(1))^l \cdot \hat{p}_{c,i}(1)^{[\hat{C}_i(m-1)]-l};$ 
7   end
8   if  $\overline{ES}_i(0, n) > \overline{ES}_i(1, n)$  then
9      $p_{c,i}(1) \leftarrow 0;$ 
10  else
11     $p_{c,i}(1) \leftarrow 1;$ 
12  end
13  NeighborChangeProb( $p_{c,i}(1)$ ,  $\hat{p}_{c,i}(1)$ ,  $C_i(0)$ );
14  return  $p_{c,i}(1);$ 
15 Procedure NeighborChangeProb ( $p_{c,i}(1)$ ,  $\hat{p}_{c,i}(1)$ ,  $C_i(0)$ )
16  if  $p_{c,i}(1) = 0$  then
17    if  $C_i(0) > 0$  then
18       $\hat{p}_{c,i}(2) \leftarrow \hat{p}_{c,i}(1) - \hat{p}_{nc,0};$ 
19    else
20       $\hat{p}_{c,i}(2) \leftarrow \hat{p}_{c,0};$ 
21    end
22  end
23  return;

```

Algorithm 2 shows the procedures to determine whether to change the firing phase in the following period or not. It utilizes $N_i^1(0)$, $C_i(0)$, and $\hat{p}_{c,i}(1)$ obtained from OnSlotStart() in Algorithm 1. The purpose of Algorithm 2 is to derive the probability for changing the firing phase, $p_{c,i}(1)$. Accordingly, we can have the probability of $p_{c,i}(1)$ as the result of Algorithm 2.

In DeterminePhaseChange(), it computes $\hat{C}_i(n)$ ($n \geq 1$) sequentially using Equation (13). From the Proposition and Equation (20), $p_{c,i}(1)$ is determined. Then, it calls NeighborChangeProb() to update $\hat{p}_{c,i}(2)$ using $\hat{p}_{c,i}(1)$.

In NeighborChangeProb(), a_i calculates $\hat{p}_{c,i}(2)$ based on $\hat{p}_{c,i}(1)$. When $p_{c,i}(1) = 1$, a_i will enter the listen mode in the following period by setting $\hat{p}_{c,i}(2)$ to $\hat{p}_{c,0}$. Thus, no further action for this case is needed in NeighborChangeProb(). The case where $p_{c,i}(1) = 0$ and $C_i(0) > 0$ means that the neighbors caused the collision will not change their firing phases. Then, $\hat{p}_{c,0}(2)$ is set to $\hat{p}_{c,0}(1) - \hat{p}_{nc,0}$, where $\hat{p}_{nc,0}$ denotes the initialization parameter of $\hat{p}_{nc,i}(1)$, which are determined as the system operational parameters by the administrator. On the other hand, i.e., if $C_i(0) = 0$, $\hat{p}_{c,0}(t+1)$ is set to $\hat{p}_{c,0}$ since no collision occurred. It is noted that $\hat{p}_{c,i}(n)$ s, for $n = 3, \dots$, are calculated at the next period in the same way.

Since the proposed method follows the same process of Ext-Desync, the complexity for running time of the proposed method is the same as that for Ext-Desync. Therefore, it becomes $O((\bar{N}_1 + \bar{N}_2)^2)$ according to [10], where \bar{N}_1 and \bar{N}_2 are the average number of one-hop and two-hop neighbors, respectively.

4. Numerical Results

The performances of the proposed method (*Proposed*) are compared to those of the following existing schemes: CSMA/CA, *Ext-Desync* with the phase changing probability of 50% fixed as illustrated in Section 2.2 [25] and the adjustment of the *Ext-Desync* scheme proposed in [35] (*Adjustment*) in which the firing phase change probability is adjusted according to the network environment. We also compared the method to support multi-hop communications by extending *Desync-TDMA*, called Multi-hop *Desync-TDMA* (*MH-Desync*) [29]. The numbers of control and data time slots for *MH-Desync* were set to 40 and 80, respectively. We implemented the network simulator for the performance comparisons using the Riverbed Modeler (formerly, OPNET Modeler) [38].

We consider a MANET environment with 50 nodes moving in a 1000×1000 m region. The wireless link bandwidth and each node's transmission range are set to 2 Mbps and 250 m, respectively. The random waypoint model is considered to model the mobile behaviors of the nodes. The mobility model was implemented in the Riverbed Modeler. The maximum movement speed of the node is set to 10, 20, and 30 m/s, which reflects the low, medium, and high-moving speeds, respectively.

Each simulation is carried out during 500 s for a given condition and repeatedly done over 10 times with different random seeds. As parameters for the proposed algorithm, \hat{P}_i is set to 0.5. In addition, $\hat{p}_{nc,0}$ is set to 0.1.

4.1. Collision Resolution Performances

As mentioned in Section 2.2, in *Ext-Desync*, each node determines its slot to be used for data transmission every period considering the firing phases of 1- and 2-hop neighbors. The neighbor relationship may change according to nodes' movements, and hence the slot size for a node may vary. The slots may be wasted due to collisions of firing messages, as illustrated in Figure 3.

Figure 6 shows the total slots wasted by collisions of firing messages due to nodes' mobility during a 500s simulation time. As the nodes' maximum speeds increase, the possibility of firing messages' collisions also increases, which increases wasted slots. As we can see from Figure 6, the *Proposed* and *Adjustment* methods have very few wasted slots and very small increments with increasing the node's maximum speed, while, for *Ext-Desync*, this increases significantly. *MH-Desync* shows much higher wasted slots than others. Since the control and data channels are divided into fixed slots in *MH-Desync*, the control-slot and firing-phase information in the control channel may collide while nodes are moving, which results in much higher slot wastes.

The results of Figure 6 imply the performances on the degree of how quickly the methods can resolve the collision situation, and the proposed method shows the best. On average, slot waste of *Proposed* is 50%, 14.6% and 1.2% of the slot waste of *Adjustment*, *Ext-Desync* and *MH-Desync*, respectively.

Figure 7 depicts the slot utilization performance, which is defined as the ratio of the sum of slot sizes that have successfully transmitted data out of the total size of allocated slots to nodes. The less the wasted slots are, the more the slots available for data transmission are. *Proposed* with the least wasted slots shows the best slot utilization performance, while *MH-Desync* shows the worst. *Proposed* shows better slot utilization performances of 1.8%, 5.4% and 12.3% in average than *Adjustment*, *Ext-Desync* and *MH-Desync*, respectively.

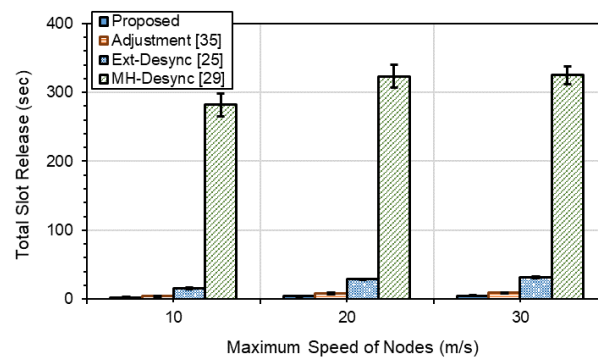


Figure 6. Total slots wasted due to collisions.

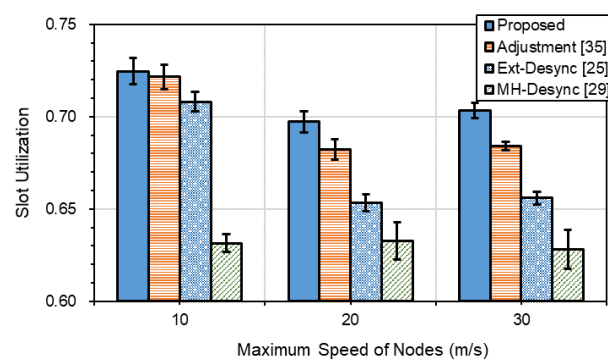


Figure 7. Average slot utilization.

4.2. Packet Delivery Performances

To compare packet delivery performances, the Ad hoc On-demand Distance Vector (AODV) routing protocol is considered. For packet generation, traffic flows are randomly paired among nodes, where we let K be the number of traffic flows. The packet size and the packet inter-arrival times are set constant at 356 bits and 0.03 s, respectively.

Figure 8a,b show the packet delivery performances in terms of the packet delivery ratio (PDR) and the end-to-end delay (E2ED), respectively, when the number of traffic flows (K) is 15. As the nodes' maximum speeds increase, PDRs decrease while E2EDs increase for all methods. Among the Ext-Desync-based schemes, as the maximum speed increases, *Proposed* shows the best, *Adjustment* and *Ext-Desync* the next, and *MH-Desync* the lowest in both PDR and E2ED performances.

Adjustment, *Ext-Desync*, *MH-Desync* and CSMA/CD show lower PDR performances of 4.5%, 13.1%, 51.7%, and 11% on average, respectively, than *Proposed*. The delay of *Proposed* is 9.4%, 19.4%, and 66.5% lower on average than *Adjustment*, *Ext-Desync*, and *MH-Desync*, respectively. On the other hand, *Proposed* shows 8.4% higher delay than CSMA/CD. The reason for the results can be derived from those shown in Figure 6. That is, as the maximum speed increases, the collision probability of firing messages increases, and times to resolve the collision situation are that *Proposed* is the shortest, *Adjustment* and *Ext-Desync* is the next, and *MH-Desync* is the longest.

On the other hand, PDRs and E2EDs for CSMA/CD are not significantly affected by the increase in maximum speed. As shown in Figure 6, *MH-Desync* shows a high amount of collisions when nodes move. Therefore, it is analyzed that data cannot be transmitted in time and has a very high delay than others. Particularly, as the maximum speed increases, PDRs of CSMA/CD are similar to *Adjustment* and show better than *Ext-Desync*. While, E2EDs of CSMA/CD show the best when the maximum speed is over 20 m/s. This is because CSMA/CD is a random access-based protocol and can transmit packets immediately as soon as it occupies the channel. Whereas, other methods can transmit packets only within slots allocated based on TDMA.

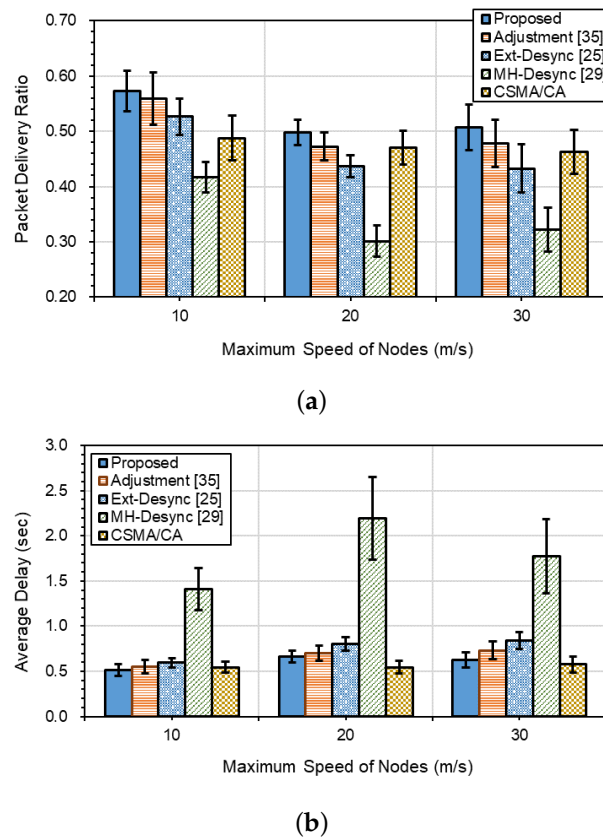


Figure 8. Packet delivery performances for $K = 15$: (a) packet delivery ratio, and (b) end-to-end delay.

When K is 20, compared with Figure 8 for $K = 15$, PDRs decrease while E2EDs increase for all methods, as shown in Figure 9a,b, respectively. As we can see from Figure 9a, PDR performances show the best in *Proposed*, followed by *Adjustment*, *Ext-Desync*, *MH-Desync* and *CSMA/CD*. *Adjustment*, *Ext-Desync*, *MH-Desync* and *CSMA/CD* show average 4.7%, 14.6%, 52.5% and 23.9%, respectively, lower PDR performances than *Proposed*. Unlike the case when $K = 15$ shown in Figure 8a, *CSMA/CD* show lower PDR than *Proposed*, *Adjustment*, and *Ext-Desync*. This is because *CSMA/CD* is a contention-based protocol.

That is, as K increases, the number of packets transmitted increases, and the degree of the contention to occupy the channel becomes serious. Likewise, as the traffic increases, *Ext-Desync*-based schemes show better PDRs than *CSMA/CD*. For both low and heavy traffic flows, *Proposed* shows the best. As shown in Figure 9b, *Proposed* shows the best E2EDs at all maximum speeds: the delays of *Adjustment*, *Ext-Desync*, *MH-Desync* and *CSMA/CD* are, respectively, 9.6%, 23.2%, 147.6%, and 4.5% on average higher than *Proposed*. While, as the maximum speed increases over 20 m/s, *CSMA/CD* outperforms *Adjustment* and *Ext-Desync* in E2ED. On the other hand, though *Proposed* is based on TDMA, it shows similar or better E2EDs with *CSMA/CA* even for high-speed moving environments.

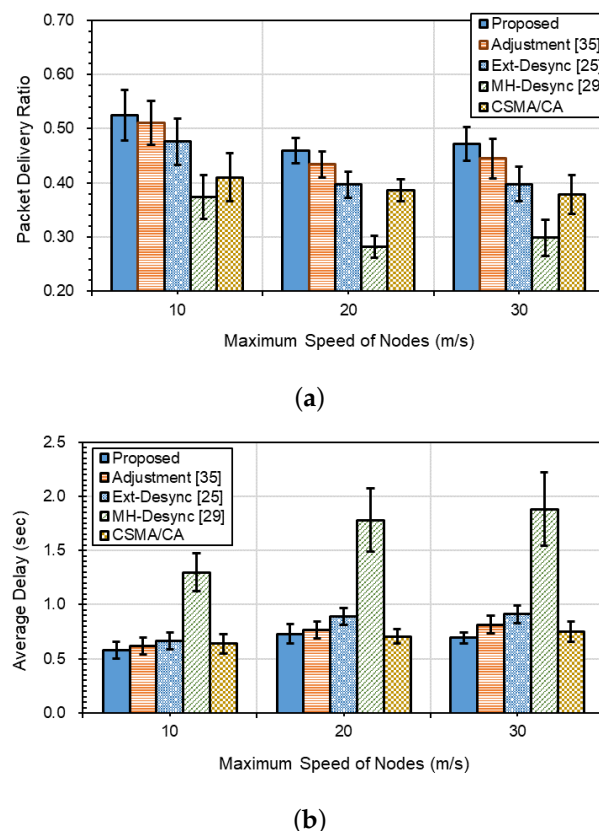


Figure 9. Packet delivery performances for $K = 20$: (a) packet delivery ratio, and (b) end-to-end delay.

5. Related Work and Discussion

In Desync-TDMA, all nodes periodically broadcast their *firing* messages every period. The firing time and the slot for each node to be utilized in the next period can be determined using the information included in firing messages from neighbors and their receiving times. The detailed process on the slot allocation is described in Section 2. The slot scheduling process is performed in a fully distributed way. After a certain period, the slot allocation situation converges if the network topology has not changed.

Desync-TDMA does not require global slot synchronization schemes as in ordinary TDMA-based methods [23,24]. Some extensions of Desync-TDMA [12–16,18–22] propose algorithm to cope with packet loss in lossy networks [12], to reduce desynchronization errors and convergence time [13,14,16,20,22], to provide weighted slot scheduling according to traffic demand of each node [15,17], or to reduce energy consumption [21]. The authors in [18] proposed the firing offset adjustment scheme to overcome the message split due to the firing message within a slot. In particular, Kuramoto-Desync [19] has all of the above advantages. However, they are not suitable for applying to MANET because they do not support the multi-hop MAC function.

In order to support multi-hop MAC function, in the Extended-Desync TDMA [25], the firing message transmitted by each node includes firing phase-related information of one-hop neighbors. However, when collisions of firing message are occurred due to the movement of the nodes, the nodes that caused the collisions change their firing phases based on the fixed probability without considering the about the surrounding environment. Therefore, as the maximum speed of the nodes increases, the performance of the Extended-Desync TDMA is extremely decreased due to the frequent change of the firing phases.

Decentralized round-robin and proportional fair scheduling [26] reduces desynchronization errors and convergence time of the Ext-Desync. MD-MAC [27], MH-Desync TDMA [29], and MH-PCO-D [34] are extensions of [25] with frame structure. These show better throughput than existing MAC protocols. Weighted-Desync TDMA [30] is an ex-

tension of [29] that provides weighted slot scheduling. However, due to the fixed frame structure, [27,29,30,34] require an additional global slot synchronization scheme to use in MANET.

Self-organizing transmission scheduling is a MAC protocol for efficiently transferring data from sensor nodes to a BS (Base Station). Each sensor node propagates its own hop count from the BS and uses schemes to avoid collisions with nodes having the same hop counts. However, it is difficult to use for MANET that can be operated in an environment in which there are a plurality of BSs or even each sensor node determines the situation by itself without the BS. Decentralized round-robin and proportional fair scheduling [26] provide weighted slot scheduling using two firing messages (start and end beacons).

The researchers in [28] proved that [26] provides MAC function without message collision even in locally connected networks. PulseSS [31] is a multi-hop extension of [26] to target an environment where at least one CH (Cluster Head) exists within the transmission range of each node. However, dynamic CH selection algorithm according to the network topology is additionally required.

In [32], the cross-layer approach in conjunction with Ext-Desync and ad-hoc routing protocols was proposed. In [33], a call admission control scheme was proposed to support QoS in an Ext-Desync-based MANET environment. In the Extended-Desync TDMA with adjustment of firing phase changing probability [35], each node adjusts its firing phase changing probability in every period based on the collision of firing messages occurred in the neighbors.

Table 3 shows the comparisons for representative Desync-TDMA and Ext-Desync-based proposals in terms of some essential networking features. Multi-hop support is one of the most important key components in distributed networking. Desync-TDMA-based methods focus on a single-hop Wireless Sensor Network (WSN) environment and do not support multi-hop. On the other hand, Ext-Desync-based ones support multi-hop communication as described in Section 2.2.

In multi-hop communication environments, collisions may occur due to the hidden terminal problem depending on the addition or mobility of nodes. Most of the methods supporting multi-hop communication provide ways to resolve collisions by the hidden terminal problem due to the addition of nodes. Still, they consider the situation in which the nodes are fixed. On the other hand, the proposed method and [35] support node mobility together.

Most Desync-TDMA and Ext-Desync-based methods focus on speeding up the convergence time for evenly distributed or weighted slot usages. However, as we raised in Section 3.1, the degradation effect of convergence time, packet loss, and slot utilization by firing message collision is not considered. Only the proposed method and [35] consider the firing message collision problem. In addition, only the proposed method utilizes the optimal criterion of firing phase change scheme, then provides the capability of quickly recognizing conflict situations and responding to conflict resolution faster than other ones. As a result, experiments showed that we could achieve performance improvements in convergence time, packet loss, and slot utilization.

In addition, most of the Desync-TDMA and Ext-Desync-based methods do not require global time synchronization. This means that Desync-TDMA and Ext-Desync-based methods are very effective ones that can utilize the advantages of TDMA fully without implying the global time synchronization problem, which is the critical limitation of TDMA-based methods.

Table 3. Comparisons of Desync-TDMA and Ext-Desync-based Proposals.

Features	Proposals																					
	Degeys et al. [10,11]	Hinterhofer et al. [12]	Choochaisri et al. [13]	Lien et al. [14]	Taniguchiet al. [15]	Gao et al. [16]	Cerioti et al. [17]	Kim, Shin et al. [18]	Yu, Choi et al. [19]	Hyun et al. [20]	Alshudukhi et al. [21]	Ron et al. [22]	Muhlberger et al. [25]	Pagliari et al. [26]	Zheng et al. [27]	Ferrari et al. [28]	Kim, Choi et al. [29]	Yu, Jung et al. [30]	Gentz et al. [31]	Jung et al. [34]	Lee et al. [35]	This work
MH support	X	X	X	X	X	X	X	X	X	X	X	X	O	O	O	O	O	O	O	O	O	O
HTP resolution in MH comm.	X	X	X	X	X	X	X	X	X	X	X	X	O	X	O	X	O	O	O	O	O	O
nodes' mobility support	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O
robustness on firing message collision	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O	O
consideration of PL degradation in resolving collisions	X	O	X	X	X	O	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O
consideration of SU degradation in resolving collisions	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	O
requirement on global time synchronization	X	X	X	X	X	X	X	X	X	X	X	O	X	X	O	X	O	O	X	O	X	X

Note: MH (Multi-hop), HTP (Hidden Terminal Problem), PL (Packet Loss), and SU (Slot Utilization).

6. Conclusions

In this paper, we dealt with the potential problem that Ext-Desync-based schemes have when operated in MANETs, which has been overlooked in other studies. Then, we derived a mathematical model to evaluate the problem. With the derived model, we proposed a method for a collided node to determine optimally whether it changes the firing phase or not in MANET. The proposed method enables collided nodes optimally to determine their phase changes in a distributed manner by considering both the collision situation and the slot utilization. In this way, the collision situation can be resolved effectively.

We showed the proposed method could resolve the collision situations much faster than other Ext-Desync-based schemes. The performances of the proposed method have been compared with CSMA/CA and other existing Ext-Desync-based schemes in terms of PDRs and E2EDs. Since Ext-Desync-based schemes are based on TDMA while CSMA/CA is random access and contention-based protocol, there have been trade-offs in PDRs and E2EDs among those schemes according to the variations of node moving speeds and the number of traffic flows. It was shown that, in all cases, the proposed method outperformed other comparable methods, including CSMA/CA.

TDMA-based approaches are particularly preferred in multihop tactical networking since TDMA provides dedicated network channels for users without contention while sharing network capacity. With the adjustment of dedicated channel sizes, this can also support Quality of Services (QoS) prioritization.

For TDMA operations, time synchronization among TDMA nodes should be required. However, in tactical MANET environments without infrastructure support, global time synchronization support is limited. As mentioned before, Ext-Desync-based methods are effective in utilizing the advantages of TDMA fully without the global time synchronization problem. In particular, the proposed method is effectively applicable to the MANET environment by providing a fast recovery in firing message collision situations caused by moving nodes, which are not considered in other studies.

This is expected to support the differentiated traffic transmission with different QoS requirements by integrating the proposed method with existing works studied on QoS issues in Desync-TDMA based networks. Additionally, in tactical networking environments, the firing messages may be lost at the physical layer by jammers. It is expected that it will be possible to distinguish the collision situations by jamming or simultaneous transmissions by extending the collision monitoring process in Ext-Desync and the proposed method. We will study these issues in future works.

Author Contributions: Conceptualization, C.-W.L., G.-M.L. and B.-H.R.; Data curation, C.-W.L. and B.-H.R.; Formal analysis, C.-W.L. and B.-H.R.; Funding acquisition, B.-H.R.; Investigation, C.-W.L., G.-M.L. and B.-H.R.; Methodology, C.-W.L., G.-M.L. and B.-H.R.; Project administration, B.-H.R.; Resources, B.-H.R.; Software, C.-W.L. and G.-M.L.; Supervision, B.-H.R.; Validation, C.-W.L.; Writing—original draft, C.-W.L., G.-M.L. and B.-H.R.; Writing—review/editing, B.-H.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Future Combat System Network Technology Research Center program of Defense Acquisition Program Administration and Agency for Defense Development (UD190033ED).

Informed Consent Statement: Not applicable.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Glossary

Abbreviation	Meaning
BS	Base Station
CSMA/CA	Carrier Sense Multiple Access with Congestion Avoidance
Desync-TDMA	Desynchronization-based TDMA
E2ED	End-to-End Delay
Ext-Desync	Extended Desync-TDMA
MAC	Medium Access Control
MANET	Mobile Ad-hoc Network
MH-Desync	Multi-hop Desync-TDMA
NOMA	Non-Orthogonal Multiple Access
PDR	Packet Delivery Ratio
QoS	Quality of Service
RTS/CTS	Request To Send/Clear To Send
TDMA	Time Division Multiple Access
WSN	Wireless Sensor Network

References

1. Zhang, F.; Yang, G. A Stable Backup Routing Protocol for Wireless Ad Hoc Networks. *Sensors* **2020**, *20*, 6743. [\[CrossRef\]](#) [\[PubMed\]](#)
2. Kaszuba-Chęcińska, A.; Chęciński, R.; Gajewski, P.; Łopatka, J. Cognitive Radio MANET Waveform Design and Evaluation. *Sensors* **2021**, *21*, 1052. [\[CrossRef\]](#) [\[PubMed\]](#)
3. Sarkar, S.K.; Basavaraju, T.; Puttamadappa, C. *Ad Hoc Mobile Wireless Networks: Principles, Protocols, and Applications*, 2nd ed.; CRC Press: Boca Raton, FL, USA, 2013. [\[CrossRef\]](#)
4. Sobrinho, J.L.; De Haan, R.; Brazio, J.M. Why RTS-CTS is not your ideal wireless LAN multiple access protocol. In Proceedings of the IEEE Wireless Communications and Networking Conference 2005, New Orleans, LA, USA, 13–17 March 2005. [\[CrossRef\]](#)
5. Ye, Y.; Zhang, X.; Xie, L.; Qin, K. A Dynamic TDMA Scheduling Strategy for MANETs Based on Service Priority. *Sensors* **2020**, *20*, 7218. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Hasan, K.F.; Wang, C.; Feng, Y.; Tian, Y.C. Time synchronization in vehicular ad-hoc networks: A survey on theory and practice. *Veh. Commun.* **2018**, *14*, 39–51. [\[CrossRef\]](#)
7. Jain, M.; Sharma, N.; Gupta, A.; Rawal, D.; Garg, P. Performance Analysis of NOMA Assisted Mobile Ad Hoc Networks for Sustainable Future Radio Access. *IEEE Trans. Sustain. Comput.* **2021**, *6*, 347–357. [\[CrossRef\]](#)
8. Zhang, Z.; Long, K.; Wang, J.; Dressler, F. On Swarm Intelligence Inspired Self-Organized Networking: Its Bionic Mechanisms, Designing Principles and Optimization Approaches. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 513–537. [\[CrossRef\]](#)
9. Jung, J.; Choi, H.; Lee, J. Survey of Bio-Inspired Resource Allocation Algorithms and MAC Protocol Design Based on a Bio-Inspired Algorithm for Mobile Ad Hoc Networks. *IEEE Commun. Mag.* **2018**, *56*, 119–127. [\[CrossRef\]](#)
10. Degesys, J.; Rose, I.; Patel, A.; Nagpal, R. DESYNC: Self-organizing desynchronization and TDMA on wireless sensor networks. In Proceedings of the 2007 6th International Symposium on Information Processing in Sensor Network, Cambridge, MA, USA, 25–27 April 2007. [\[CrossRef\]](#)
11. Patel, A.; Degesys, J.; Nagpal, R. Desynchronization: The theory of self-organizing algorithms for round-robin scheduling. In Proceedings of the First International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2007), Cambridge, MA, USA, 9–11 July 2007. [\[CrossRef\]](#)
12. Hinterhofer, T.; Schwefel, H.P.; Tomic, S. RD2: Resilient dynamic desynchronization for TDMA over Lossy Networks. In Proceedings of the 2012 IEEE 31st Symposium on Reliable Distributed Systems, Irvine, CA, USA, 8–11 October 2012. [\[CrossRef\]](#)
13. Choochaisri, S.; Apicharttrisor, K.; Korprasertthaworn, K.; Taechalertpaisarn, P.; Intanagonwiwat, C. Desynchronization with an artificial force field for wireless networks. *ACM SIGCOMM Comput. Commun. Rev.* **2012**, *42*, 7–15. [\[CrossRef\]](#)
14. Lien, C.; Chang, S.; Chang, C.; Lee, D. Anchored desynchronization. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012. [\[CrossRef\]](#)
15. Taniguchi, Y. Desynchronization-based weighted scheduling adaptive to traffic load for wireless networks. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014. [\[CrossRef\]](#)
16. Gao, H.; Wang, Y. On phase response function based decentralized phase desynchronization. *IEEE Trans. Signal Process.* **2017**, *65*, 5564–5577. [\[CrossRef\]](#)
17. Ceriotti, M.; Murphy, A.L. Reins-MAC: Firefly Inspired Communication Scheduling for Reliable Low-Power Wireless. In Proceedings of the 2018 IEEE 12th International Conference on Self-Adaptive and Self-Organizing Systems (SASO), Trento, Italy, 3–7 September 2018; pp. 140–149. [\[CrossRef\]](#)
18. Kim, K.; Roh, B.; Shin, S. Firing Offset Adjustment of Bio-Inspired DESYNC-TDMA to Improve Slot Utilization Performances in Wireless Sensor Networks. *KSII Trans. Internet Inf. Syst.* **2017**, *11*, 1492–1509. [\[CrossRef\]](#)

19. Yu, U.; Choi, H.; Lee, J. Kuramoto-Desync: Distributed and Fair Resource Allocation in a Wireless Network. *IEEE Access* **2019**, *7*, 104769–104776. [\[CrossRef\]](#)
20. Hyun, S.; Kim, G.; Yang, D. PD-DESYNC: Practical and Deterministic Desynchronization in Wireless Sensor Networks. *KSII Trans. Internet Inf. Syst.* **2019**, *13*, 3880–3899. [\[CrossRef\]](#)
21. Alshudukhi, J.S.; Al-Mekhlafi, Z.G.; Alshammari, M.T.; Mohammed, B.A. Desynchronization Traveling Wave Pulse-Coupled-Oscillator Algorithm Using a Self-Organizing Scheme for Energy-Efficient Wireless Sensor Networks. *IEEE Access* **2020**, *8*, 196223–196234. [\[CrossRef\]](#)
22. Ron, D.; Lee, J.R. Fair and Distributed Resource Allocation in Wireless Networks Using Frogs' Calling Behavior Algorithm. *IEEE Access* **2019**, *7*, 87313–87322. [\[CrossRef\]](#)
23. Al-Mekhlafi, Z.G.; Hanapi, Z.M.; Shamsan Saleh, A.M. Firefly-Inspired Time Synchronization Mechanism for Self-Organizing Energy-Efficient Wireless Sensor Networks: A Survey. *IEEE Access* **2019**, *7*, 115229–115248. [\[CrossRef\]](#)
24. Butt, M.M.; Dey, I.; Dzaferagic, M.; Murphy, M.; Kaminski, N.; Marchetti, N. Agent-Based Modeling for Distributed Decision Support in an IoT Network. *IEEE Internet Things J.* **2020**, *7*, 6919–6931. [\[CrossRef\]](#)
25. Mühlberger, C.; Kolla, R. Extended Desynchronization for Multi-Hop Topologies. Technical Report. 2009. Available online: https://www.informatik.uni-wuerzburg.de/fileadmin/10030500/user_upload/staff/muehlberger/Extended_Desynchronization_for_Multi-Hop_Topologies_TR_460.pdf (accessed on 23 September 2021).
26. Pagliari, R.; Hong, Y.; Scaglione, A. Bio-inspired algorithms for decentralized round-robin and proportional fair scheduling. *IEEE J. Sel. Areas Commun.* **2010**, *28*, 564–575. [\[CrossRef\]](#)
27. Zheng, C.; Huang, S.; Wei, J.; Dong, Q. MD-MAC: A Distributed TDMA Protocol Based on Desynchronization for Multi-Hop Topologies. *Sensors* **2019**, *19*, 5102. [\[CrossRef\]](#)
28. Ferrari, L.; Scaglione, A.; Gentz, R.; Hong, Y.P. Convergence results on pulse coupled oscillator protocols in locally connected networks. *IEEE/ACM Trans. Netw.* **2016**, *25*, 1004–1019. [\[CrossRef\]](#)
29. Kim, Y.; Choi, H.; Lee, J. A bioinspired fair resource-allocation algorithm for TDMA-based distributed sensor networks for IoT. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 7296359. [\[CrossRef\]](#)
30. Yu, U.; Jung, J.; Kong, E.; Choi, H.; Lee, J. Weighted-DESYNC and its application to end-to-end throughput fairness in wireless multihop network. *Mob. Inf. Syst.* **2017**, *2017*, 2504604. [\[CrossRef\]](#)
31. Gentz, R.; Scaglione, A.; Ferrari, L.; Hong, Y. PulseSS: A pulse-coupled synchronization and scheduling protocol for clustered wireless sensor networks. *IEEE Internet Things J.* **2016**, *3*, 1222–1234. [\[CrossRef\]](#)
32. Kim, K.; Lee, C.; Roh, B.H.; Roh, B.S.; Han, M. Routing Metric based on Slot Length of AODV on Multihop DESYNC-TDMA. *EAI Endorsed Trans. Mob. Commun. Appl.* **2017**, *3*. [\[CrossRef\]](#)
33. Roh, B.S.; Han, M.; Hoh, M.; Park, H.; Kim, K.; Roh, B.H. Distributed Call Admission Control for DESYNC-TDMA in Mobile Ad Hoc Networks. *EAI Endorsed Trans. Mob. Commun. Appl.* **2017**, *3*. [\[CrossRef\]](#)
34. Jung, J.Y.; Lee, J.R. Distributed Fair Resource Allocation Algorithm in Wireless Multihop Network Systems. *IEEE Syst. J.* **2020**. [\[CrossRef\]](#)
35. Lee, C.; Lee, G.; Roh, B. Adjustment of firing phase changing probability in extended-Desync TDMA. In Proceedings of the 2018 Wireless Telecommunications Symposium (WTS), Phoenix, AZ, USA, 17–20 April 2018. [\[CrossRef\]](#)
36. Degesys, J.; Nagpal, R. Towards desynchronization of multi-hop topologies. In Proceedings of the 2008 Second IEEE International Conference on Self-Adaptive and Self-Organizing Systems, Venezia, Italy, 20–24 October 2008. [\[CrossRef\]](#)
37. Mühlberger, C. Analyzing a self-organizing multi-hop protocol: Ease of simulations and need for real-world tests. In Proceedings of the 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, Italy, 1–5 July 2013. [\[CrossRef\]](#)
38. Riverbed. Riverbed Modeler: The Fastest Discrete Event-Simulator for Network Modeling. Available online: <https://www.riverbed.com/sg/products/npm/riverbed-modeler.html> (accessed on 23 September 2021).