

Article

A Multi-Path MAC Scheduling Scheme for Multi-Channel Wireless Sensor Networks

Fan Zhang ¹  and Gangqiang Yang ^{2,*} 

¹ School of Information Engineering, Shandong Management University, Jinan 250357, China; ccfov@126.com

² School of Information Science and Engineering, Shandong University, Qingdao 266237, China

* Correspondence: g37yang@sdu.edu.cn

Abstract: Designing reasonable MAC scheduling strategies is an important means to ensure transmission quality in wireless sensor networks (WSNs). When there exist multiple available routes from the source to the destination, it is necessary to combine a data traffic allocation mechanism and design a multi-path MAC scheduling scheme in order to ensure QoS. This paper develops a multi-path resource allocation method for multi-channel wireless sensor networks, which uses random-access technology to complete MAC scheduling and selects the transmission path for each packet according to the probability. Through theoretical analysis and simulation experiments, it can be found that the proposed strategy can provide a reliable throughput capacity region. Meanwhile, due to the use of random-access technology, the computational complexity of the proposed algorithm can be independent of the number of links and channels.

Keywords: MAC scheduling; random-access; capacity region; computational complexity



Citation: Zhang, F.; Yang, G. A Multi-Path MAC Scheduling Scheme for Multi-Channel Wireless Sensor Networks. *Entropy* **2023**, *25*, 1197. <https://doi.org/10.3390/e25081197>

Academic Editors: Ramón Agüero, Javier Rubio-Loyola and César Augusto Azurdia Meza

Received: 12 June 2023

Revised: 9 August 2023

Accepted: 10 August 2023

Published: 11 August 2023



Copyright: © 2023 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

In wireless sensor networks (WSNs), the use of scheduling policies as the MAC layer protocol can bring better QoS than the traditional competition-based CSMA/CD protocol [1]. According to the different methods of resource allocation, one can adopt centralized or distributed methods to achieve network transmission in WSNs. Correspondingly, the MAC scheduling schemes can also be classified as centralized scheduling and distributed scheduling. A centralized scheduler requires the central controller in the network to allocate resources overall, whereas distributed schedulers let each node complete the scheduling and transmission of data messages by itself. In the 5G era, distributed WSNs can serve as the underlying access network below the base station to provide a large number of Internet of Things (IoT) services. Therefore, this paper mainly studies the distributed scheduling strategy in WSNs. As we know, queue-stable is an important state of wireless network systems, which means that the network can bear the current input load [2]. Reasonable MAC scheduling can maintain the queue-stable state of the network system when the input load is located within a certain capacity region. On the contrary, if the network system does not reach a queue state, the queue length may reach infinity, which would impose a huge burden on the network nodes.

In a single-hop or single-route network scenario, the MAC scheduling algorithm determines whether each link queue transmits packets in the current time slot. In actual network scenarios, if there exist multiple alternative paths available between the source node and the destination node, it is necessary to allocate data traffic (i.e., arriving packets) to different paths reasonably based on MAC scheduling, so as to better balance the network load and obtain better transmission performance [3]. In a multi-path environment, as shown in Figure 1, we aim to ensure a reliable throughput capacity region for a multi-path MAC scheme that combines a distributed MAC scheduler and a data traffic allocation mechanism.

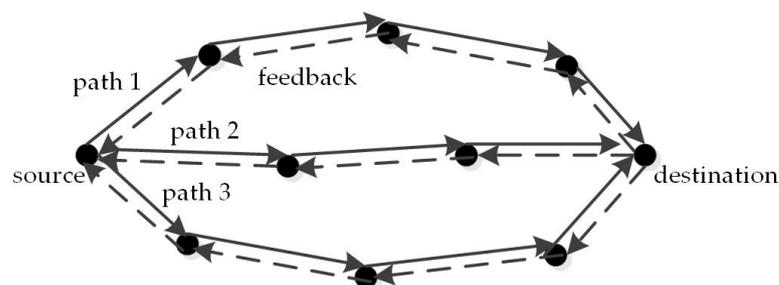


Figure 1. Transmission with multiple routes.

Researchers have made a series of research progress in addressing the distributed MAC scheduling problem. We organize and summarize the current research status from two aspects: single-channel scheduling and multi-channel scheduling and introduce the ideas and innovations of this paper.

1.1. Scheduling for Single-Channel Networks

With regard to the distributed MAC scheduling problem in single-channel networks, some researchers have proposed a maximal scheduling (MS) strategy for various interference models [4,5] and demonstrated that MS can ensure a queue-stable state within a certain throughput capacity region. Subsequently, Neely et al. further proposed a multi-path extension scheme for MS [6]. For the purpose of further reducing the implementation complexity of the distributed algorithm, researchers put forward a Q-SCHED (queue-length scheduler) algorithm that completes scheduling by obtaining the queue-length information of neighbors [7,8]. Q-SCHED has been proven to ensure a queue-stable state within a certain throughput capacity region. Moreover, the implementation complexity of Q-SCHED does not increase with the network size. Because of such a feature, Q-SCHED is applicable to large-scale dense networks. In addition, Xie et al. considered the interference intensity and channel gain between links under directional transmission and designed a link scheduling algorithm in directional wireless communication networks [9]. This algorithm utilizes the idea of reinforcement learning to dynamically allocate resources. Each node can independently make intelligent decisions on its own link scheduling by sensing the communication environment within its range, thereby maximizing the network capacity. After analysis, the above-distributed scheduling strategies could all ensure the queue-stable state of the network within a certain throughput capacity region.

1.2. Scheduling for Multi-Channel Networks

In wireless networks, wireless spectrum resources can be divided into multiple different channels, each occupying a certain frequency range. Wireless network nodes can choose a certain channel to transmit data on their own. Thus, links within the same interference area can transmit data without interference by using different channels, which increases throughput and bandwidth [10]. If multiple interfaces are configured for each wireless node, a single node can perform transmission and reception operations simultaneously, which further enhances the transmission efficiency [11], as shown in Figure 2. It should be noted that the application of multiple channels not only improves transmission efficiency and bandwidth but also brings challenges in software implementation.

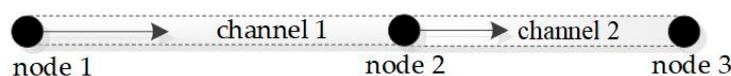


Figure 2. Transmission equipping multiple channels.

For distributed MAC scheduling in multi-channel networks, due to capacity differences between different channels, directly extending the single-channel MAC scheduling strategies to a multi-channel scenario may result in extremely poor throughput

performance [12,13]. Following this motivation, Lin and Zhang et al. utilized the idea of relay forwarding and developed a distributed SP (single path) algorithm [14] as well as its multi-path extension scheme MP (multiple path) [15]. Yun et al. analyzed the impact of channel switching delay on the performance of scheduling strategies based on SP [16]. Bhandari et al. put forward a MAC scheduling policy without using relay forwarding for multi-channel single-interface networks [17]. Cheng et al. designed an appealing mathematical model. This model can virtualize a multi-interface multi-channel network into a single-channel network. Hence, distributed scheduling strategies in single-channel networks can be directly extended in multi-channel networks without sacrificing throughput performance [18]. Zhang et al. employed the single-channel Q-SCHED scheduling idea to propose a novel distributed algorithm with low complexity (DA-LC) [19,20]. In addition, Xu et al. proposed a low-complexity distributed channel allocation and scheduling strategy (LDCS) [21] in single-interface multi-channel WSNs. The implementation complexity of both DA-LC and LDCS is independent of the network size. However, they can only apply to single-path network environments. In addition to the above achievements, some researchers have systematically analyzed the performance of existing MAC scheduling strategies from the perspectives of throughput and latency [22,23].

1.3. Idea and Innovation

In summary, wireless network MAC scheduling technology that ensures QoS has attracted widespread attention, and designing excellent MAC scheduling strategies has become one of the important technical means to solve resource allocation problems in wireless networks. However, by examining existing research results, it can be seen that existing MAC scheduling schemes for multi-path environments still use a maximal-matching mechanism during the scheduling phase. In multi-channel WSNs, such a mechanism brings high implementation complexity. For a multi-channel network containing $|L|$ links and $|C|$ channels, performing a complete maximal match requires at least $O(|L||C|)$ iterations [24]. Hence, the computational complexity of the multi-path MAC scheduling scheme based on maximal-matching increases with the number of links and channels and is not suitable for large-scale dense distributed networks.

In the 5G era, a large number of IoT applications have emerged. As the underlying network, the network scale of distributed WSNs is becoming larger and the node density is becoming higher. In this case, the implementation complexity of network protocols or algorithms continues to increase, which restricts network performance. It is necessary to reduce the complexity of MAC scheduling strategies in order to meet the requirements of network construction in the 5G era. Based on the above-mentioned analysis, a multi-path MAC scheduling scheme with low complexity, named M-Scheduler, is proposed in this paper. Under the M-Scheduler policy, a tuple-based mathematical model is employed and the random-access technique [25] is used to complete distributed scheduling. Furthermore, a data traffic allocation mechanism is developed. M-Scheduler is proven to ensure the queue-stable state within a reliable throughput capacity region. Meanwhile, a random-access technique is used for scheduling and, hence, the implementation complexity of the algorithm is independent of the network scale. The research results of this paper would provide technical support for the application of MAC scheduling technology in distributed networks in 5G environments.

2. System Model

Reference [18] points out that using a tuple-based mathematical model can virtualize a multi-interface multi-channel network into a single-channel network. After virtualization, MAC scheduling strategies for single-channel environments can be extended to multi-channel environments, which evidently leads to better performance. Such a tuple-based mathematical model is employed in this paper. Under the tuple-based model, each node, interface, and channel are virtualized as a tuple node. The link between a pair of tuple nodes is defined as a tuple link, as shown in Figure 3. In practical networks, one can use a

simple mechanism to implement this model. For example, in a certain time slot, tuple node (m, i, c) works means that the original node m works on channel c using interface i . The tuple link between (m, i, c) and (g, j, c) works means that node m uses interface i to send data to node g on channel c and node g uses interface j to receive data. After virtualization, the set of all tuple links in the network is denoted by L , and the set of tuple nodes is N . For convenience, in the remaining part of this paper, the word “node” is directly used to represent the tuple node, and “link” represents the tuple link.

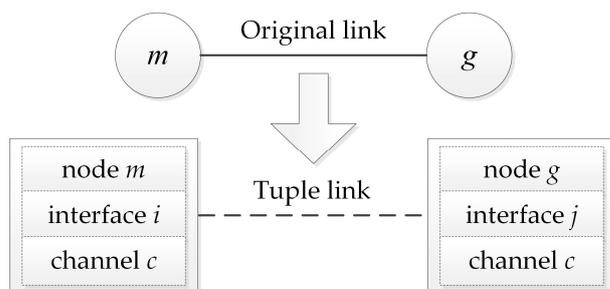


Figure 3. Virtualization process of the tuple-based mathematical model.

In the scenario defined in this paper, it is assumed that there are S data sources in total and there exist $I(s)$ transmission paths from each data source node s (tuple node) to the destination node. We use H as the path indicator variable. If the path i of source node s passes through link l , then we have $H_{si}^l = 1$; otherwise, $H_{si}^l = 0$. Assuming that $A_s(n)$ represents the number of data packets generated by the source node s in time slot n . Without loss of generality, it is further assumed that the random variable $A_s(n)$ is independent and identically distributed across time slot, and the mean value of $A_s(n)$ is λ_s . Hence, the data traffic arriving at link l in time slot n can be denoted by:

$$A_l(n) = \sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l A_s(n) T_{si}(n) \tag{1}$$

where $T_{si}(n)$ represents the proportion of the number of data packets allocated by source node s to path i in time slot n . When a packet arrives in link l , it is queued according to the “first in, first out” principle, forming the queue length $q_l(n)$ in time slot n . Let r_l represent the capacity of link l (i.e., transmission rate). According to the output results of the MAC scheduling algorithm, the indicator variable $\phi_l(n)$ indicates whether link l sends data in time slot n . If l is scheduled and $q_l(n) > 0$, then $\phi_l(n) = 1$; otherwise, $\phi_l(n) = 0$. Thus, the queue updating is expressed as:

$$\begin{aligned} q_l(n+1) &= [q_l(n) + A_l(n) - \phi_l(n)r_l]^+ \\ &= [q_l(n) + \sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l A_s(n) T_{si}(n) - \phi_l(n)r_l]^+ \end{aligned} \tag{2}$$

where $[]^+$ is the projection on $[0, +\infty)$. The queue update process is described in Figure 4.

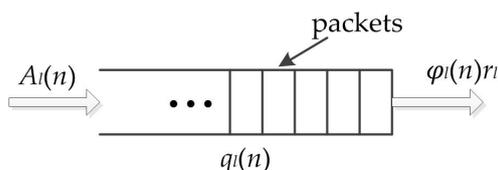


Figure 4. Queue update process.

As we know, there exists interference in the information transmission process in wireless networks. In multi-channel multi-interface networks, interference may come from shared channel or interface conflict, the former is also known as co-frequency interference. The purpose of designing a MAC scheduling strategy is to avoid interference. Based on

this motivation, we define an interference set ε_l for each link l , which includes all links that may cause interference to l . Since we are using a tuple link here, the interference set of l includes all cases of channel interference and interface conflicts. Define the interference degree of link l as the number of links in the set ε_l that can simultaneously send data. The overall interference degree of a network is defined as the maximum interference degree of all links, represented by the symbol K .

In practical network systems, queue-stable is an important state measuring network throughput performance, which refers to the network being able to bear the current input load. If there exists a MAC scheduling strategy that can maintain the queue-stable state of the network when the input load vector $[\lambda_1, \dots, \lambda_s]$ is located within a certain region Ω , then the region Ω is named as the maximum throughput capacity region. In the multi-path scenario used in this paper, the maximum throughput capacity region Ω can be defined as a set of all input loads $[\lambda_1, \dots, \lambda_s]$ that meet the following conditions:

$$\begin{aligned} & \left[\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \tilde{T}_{si} \right] \in \Lambda \\ & \text{s.t. } \sum_{i=1}^{I(s)} \tilde{T}_{si} = 1, \tilde{T}_{si} \geq 0 \end{aligned} \tag{3}$$

where the set Λ is the maximum capacity region defined in reference [2] for a single-path network environment. On the other hand, if a MAC scheduling strategy can achieve a capacity region of $k\Omega$, then the strategy is called throughput guaranteed, and the efficiency ratio is k ($0 \leq k \leq 1$). Obviously, a larger value of k means that the MAC scheduling strategy has better throughput performance. The multi-path MAC scheduling scheme proposed in this paper is throughput guaranteed. Moreover, the implementation complexity of the proposed scheme is independent of the network size. The following sections will provide a detailed explanation and analysis of the execution process and the performance of the proposed algorithm.

3. M-Scheduler Scheme

In multi-path network scenarios, there exist multiple transmission routes between the data source and the destination node. Therefore, in order to ensure the throughput performance of MAC scheduling strategies, it is necessary to simultaneously consider the traffic allocation problem of data packets on multiple paths. Excellent MAC scheduling strategies can always reasonably share the network load and fully utilize the network resources. The detailed implementation process of the multi-path MAC scheduling scheme (named M-Scheduler) proposed in this paper can be divided into the following three steps.

Step 1: Data traffic allocation. In each time slot n , when a source node s has packets to send, it first calculates the packet allocation proportion vector $T_s(n) = [T_{s1}(n), \dots, T_{sI(s)}(n)]$. Such a proportion vector indicates the proportion of data packets sent by the source node to each path and its calculation is completed by solving the following maximum value problem:

$$\begin{aligned} \max_{\vec{T}_s(n)} & \left\{ -\frac{\phi_s}{2} \sum_{i=1}^{I(s)} [T_{si}(n)]^2 \right. \\ & \left. - \sum_{i=1}^{I(s)} T_{si}(n) \sum_{l=1}^L \frac{H_{si}^l}{r_l} \sum_{j \in \varepsilon_l} \left[\sum_{k \in \varepsilon_j} \frac{q_k(n)}{r_k} \right]^v \right\} \\ \text{s.t. } & T_{si}(n) \geq 0, \sum_{i=1}^{I(s)} T_{si}(n) = 1 \end{aligned} \tag{4}$$

where ϕ_s and v are positive constants. After solving the problem of (4), the vector $T_s(n)$ can be obtained. In detail, the element in the vector $T_s(n)$ represents the probability of the source node sending a data packet to the corresponding path. For example, if $T_{si}(n) = 0.2$, the source node s sends data to path i with a probability of 20% in time slot n . In this way, $A_s(n)T_{si}(n)$ represents the total number of data packets sent to path i by source node s in time slot n .

Step 2: Time slot segmentation. In order to make the implementation complexity of the MAC scheduling strategy independent of the network size, it is necessary to divide the time length of each time slot n into two parts: a calculation time slot and a transmission

time slot, as shown in Figure 5. Furthermore, the calculation time slot is divided into P mini time slots. Within time slot n , each link selects one of the P mini-time-slots according to the mechanism proposed in step 3 as the starting time for data transmission.

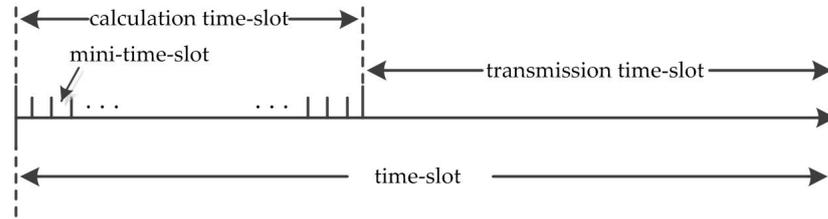


Figure 5. Division of each time slot.

Step 3: Scheduling. In each time slot n , each link l selects a certain mini time slot in the set $\{1, \dots, P + 1\}$ with probability:

$$\begin{aligned} \Pr\{Y_l = P + 1\} &= e^{-\psi_l} \\ \Pr\{Y_l = p\} &= e^{-\psi_l \frac{p-1}{P}} - e^{-\psi_l \frac{p}{P}} \quad p = 1, \dots, P \end{aligned} \tag{5}$$

where Y_l denotes the mini time slot picked by link l , and the parameter ψ_l is calculated using the following formula:

$$\psi_l = \alpha \frac{\frac{q_l(n)}{r_l}}{\max_{j \in \varepsilon_l} \left[\sum_{k \in \varepsilon_j} \frac{q_k(n)}{r_k} \right]} \tag{6}$$

where $\alpha = \ln P$. When link l picks $P + 1$, it means that it will not send data in the current time slot. If two neighboring links that interfere with each other (interference may come from a shared channel or interface conflict) choose the same mini time slot, then both will not send data in this time slot. Meanwhile, if link l detects other links in its interference set sending data when it is about to start sending, it will remain silent during this time slot. From Formula (6), it can be seen that the calculation of parameter ψ_l requires the neighbor queue information of link l , which can be assisted by using the RTS/CTS mechanism in the MAC protocol.

4. Performance Analysis

The performance of the M-Scheduler scheme is analyzed here from two aspects: throughput and complexity. With regard to throughput performance, we demonstrate that the proposed M-Scheduler is throughput guaranteed, and the throughput efficiency ratio of the algorithm is further obtained. In terms of complexity, we will explain that the implementation complexity of M-Scheduler is independent of the network size. Based on this characteristic, the M-Scheduler strategy has significant advantages in complexity compared with the existing maximal-matching-based multi-path MAC scheduling schemes.

4.1. Throughput Performance

For the purpose of achieving the throughput performance of M-Scheduler, the following lemma is needed.

Lemma 1. For any positive constant $\mu > 0$, there exists $\tau_0 > 0$ such that when $\tau \geq \tau_0$, we have:

$$\frac{\sum_{j \in L} \left(\sum_{l \in \varepsilon_j} \frac{q_l(n)}{r_l} \right)^{1+\tau}}{\max_{j \in L} \sum_{l \in \varepsilon_j} \frac{q_l(n)}{r_l}} \geq e^{-\mu} \sum_{j \in L} \left(\sum_{l \in \varepsilon_j} \frac{q_l(n)}{r_l} \right)^\tau. \tag{7}$$

Proof of Lemma 1. Firstly, define a parameter b_j regarding link j , denoted as:

$$b_j = \frac{\sum_{l \in \epsilon_j} \frac{q_l(n)}{r_l}}{\max_{u \in L} \sum_{l \in \epsilon_u} \frac{q_l(n)}{r_l}}. \tag{8}$$

It can be seen that the value range of b_j is $0 \leq b_j \leq 1$. At the same time, it can be found that there must be a link j in the network that makes $b_j = 1$. Without loss of generality, let $b_1 = 1$. Hence, if the following inequality holds for every b_j ($j \in L$ and $j \neq 1$), Lemma 1 is then obtained:

$$1 + \sum_{j \in L, j \neq 1} b_j^{1+\tau} \geq e^{-\mu} \left(1 + \sum_{j \in L, j \neq 1} b_j^\tau \right) \tag{9}$$

To prove (9), we divide it into two cases. For any $\tau > 0$, if $b_j \geq e^{-\mu}$, we have $(b_j)^{1+\tau} \geq (b_j)^\tau e^{-\mu}$; if $b_j \leq e^{-\mu}$, then we have $(b_j)^{1+\tau} \geq 0$. Hence, by combining the two cases, the following inequality can be obtained:

$$b_j^{1+\tau} \geq b_j^\tau e^{-\mu} - e^{-\mu(1+\tau)}. \tag{10}$$

Thus, we further have:

$$1 + \sum_{j \in L, j \neq 1} b_j^{1+\tau} \geq 1 + \sum_{j \in L, j \neq 1} b_j^\tau e^{-\mu} - (|L| - 1)e^{-\mu(1+\tau)} \tag{11}$$

where $|L|$ denotes the cardinality of the set L . Hence, when the parameter $\mu > 0$ is given, we can appropriately choose a positive number τ_0 to make $1 - (|L| - 1)e^{-\mu(1+\tau_0)} \leq e^{-\mu}$. Thus, for arbitrary $\tau > \tau_0$, it is not difficult to deduce (9). Therefore, Lemma 1 was proven. After obtaining Lemma 1, the following theorem is given to describe the throughput performance of M-Scheduler. \square

Theorem 1. *Employing the M-Scheduler strategy, for any $\mu > 0$, if the input load vector denoted by $\lambda = [\lambda_1, \dots, \lambda_S]$ satisfies*

$$\sum_{l \in \epsilon_j} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \tilde{T}_{si}}{r_l} \leq \frac{e^{-\mu}}{1+\mu} \left(1 - \frac{\ln P+1}{P} \right) \tag{12}$$

s.t. $\sum_{i=1}^{I(s)} \tilde{T}_{si} = 1, \tilde{T}_{si} \geq 0$

then the network system can maintain a queue-stable state.

Proof of Theorem 1. Firstly, construct the following Lyapunov function:

$$V(n) = \frac{1}{1+\tau} \sum_{j \in L} \left(\sum_{l \in \epsilon_j} \frac{q_l(n)}{r_l} \right)^{1+\tau} \tag{13}$$

To simplify the subsequent proof process, we define two new parameters $x_l(n)$ and Υ_j as:

$$x_l(n) = E[A_l(n)] = \sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s T_{si}(n) \tag{14}$$

$$\Upsilon_j = \sum_{l \in \epsilon_j} \frac{q_l(n)}{r_l}$$

where $E[\cdot]$ represents the mathematical expectation. Due to $E[A_s(n)T_{si}(n)] = \lambda_s T_{si}(n)$, it can be then obtained that:

$$\begin{aligned}
 & E[V(n+1) - V(n)|\mathbf{q}(n)] \\
 & \leq \sum_{j \in L} (\gamma_j)^\tau \left[\sum_{l \in \varepsilon_j} \frac{x_l(n)}{r_l} - \sum_{l \in \varepsilon_j} \Pr\{Z_l\} \right] \\
 & \quad + \sum_{j \in L} o \left[(\gamma_j)^\tau \right]
 \end{aligned} \tag{15}$$

where $\mathbf{q}(n) = [q_j(n), j \in L]$, Z_l indicates the event that link l is scheduled, and $o[\cdot]$ denotes the higher order infinitesimal. Furthermore, according to Equations (5) and (6), it is not difficult to obtain:

$$\sum_{l \in \varepsilon_j} \Pr\{Z_l\} \geq \sum_{p=1}^P e^{(-\frac{p}{P}\alpha)} \left(\sum_{l \in \varepsilon_j} \frac{\psi_l}{P} \right) \geq \frac{\gamma_j}{\max_{k \in L} \gamma_k} \left(1 - \frac{\ln P + 1}{P} \right) \tag{16}$$

which holds for arbitrary link j . Thus, according to Lemma 1, for any positive constant μ , there exists $\tau_0 > 0$ such that when $\tau \geq \tau_0$, we have:

$$\begin{aligned}
 & \sum_{j \in L} (\gamma_j)^\tau \left[\sum_{l \in \varepsilon_j} \Pr\{Z_l\} \right] \\
 & \geq \frac{(1 - \frac{\ln P + 1}{P})}{\max_{k \in L} \gamma_k} \sum_{j \in L} (\gamma_j)^{(\tau+1)} \\
 & \geq e^{-\mu} \left(1 - \frac{\ln P + 1}{P} \right) \sum_{j \in L} (\gamma_j)^\tau.
 \end{aligned} \tag{17}$$

Based on the usage conditions of Theorem 1 shown in Equations (12) and (15), we can obtain:

$$\begin{aligned}
 & E[V(n+1) - V(n)|\mathbf{q}(n)] \\
 & \leq \sum_{j \in L} (\gamma_j)^\tau \left[\sum_{l \in \varepsilon_j} \frac{x_l(n)}{r_l} - (1 + \mu) \sum_{l \in \varepsilon_j} \frac{x_l^\#}{r_l} \right] + \sum_{j \in L} o \left[(\gamma_j)^\tau \right] \\
 & = \sum_{j \in L} (\gamma_j)^\tau \left[\sum_{l \in \varepsilon_j} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s T_{si}(n)}{r_l} \right. \\
 & \quad \left. - (1 + \mu) \sum_{l \in \varepsilon_j} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \tilde{T}_{si}}{r_l} \right] + \sum_{j \in L} o \left[(\gamma_j)^\tau \right] \\
 & = \sum_{s=1}^S \lambda_s \left[\sum_{i=1}^{I(s)} T_{si}(n) \sum_{l \in L} \frac{H_{si}^l}{r_l} \sum_{j \in \varepsilon_l} (\gamma_j)^\tau \right. \\
 & \quad \left. - (1 + \mu) \sum_{i=1}^{I(s)} \tilde{T}_{si} \sum_{l \in L} \frac{H_{si}^l}{r_l} \sum_{j \in \varepsilon_l} (\gamma_j)^\tau \right] + \sum_{j \in L} o \left[(\gamma_j)^\tau \right]
 \end{aligned} \tag{18}$$

where $x_l^\# = \sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \tilde{T}_{si}$. According to the data-traffic-allocation mechanism as in Equation (4), it is not difficult to obtain the following inequality:

$$\begin{aligned}
 & \sum_{i=1}^{I(s)} \left[T_{si}(n) - \tilde{T}_{si} \right] \sum_{l \in L} \frac{H_{si}^l}{r_l} \sum_{j \in \varepsilon_l} (\gamma_j)^\tau \\
 & \leq -\frac{\phi_s}{2} \sum_{i=1}^{I(s)} [T_{si}(n)]^2 + \frac{\phi_s}{2} \sum_{i=1}^{I(s)} [\tilde{T}_{si}]^2 \\
 & \leq \phi_s \cdot [I(s)].
 \end{aligned} \tag{19}$$

Hence, based on Equations (18) and (19), we can further deduce that:

$$\begin{aligned}
 & E[V(n+1) - V(n)|\mathbf{q}(n)] \\
 & \leq -\mu \sum_{j \in L} (\gamma_j)^\tau \left[\sum_{l \in \varepsilon_j} \frac{x_l^\#}{r_l} \right] \\
 & \quad + \sum_{i=1}^{I(s)} \lambda_s \phi_s I(s) + \sum_{j \in L} o \left[(\gamma_j)^\tau \right].
 \end{aligned} \tag{20}$$

Thus, the inequality Equation (20) implies that the network system maintains queue stability [26], and Theorem 1 is proved. \square

It should be emphasized that if the interference degree of the entire network is K , then the maximum throughput capacity region Ω can be denoted by the set of all input load vectors λ meeting:

$$\sum_{l \in \varepsilon_j} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \tilde{T}_{si}}{r_l} \leq K \quad (21)$$

$$s.t. \quad \sum_{i=1}^{I(s)} \tilde{T}_{si} = 1, \quad \tilde{T}_{si} \geq 0$$

for each link j [27]. Hence, we can conclude that when the value of μ is small enough and P is large enough, the right half of inequality Equation (12) approaches 1, i.e.:

$$\lim_{\mu \rightarrow 0, P \rightarrow +\infty} \frac{e^{-\mu}}{1 + \mu} \left(1 - \frac{\ln P + 1}{P} \right) = 1. \quad (22)$$

By comparing Inequalities (12) and (21), it can be concluded that the throughput efficiency ratio of the M-Scheduler strategy is close to $1/K$. That is to say, M-Scheduler can ensure the queue-stable state of the network system when $\lambda \in \Omega/K$. Thus, the proposed scheme is throughput guaranteed. According to the above analysis, the larger the number of mini time slots P , the closer the throughput efficiency ratio of M-Scheduler is to $1/K$. In the future, the hardware performance of the communication terminal can be improved to ensure that the value of P is large enough. In theory, one can compare the throughput performance of M-Scheduler with other existing comparable strategies. The MP algorithm proposed in [15] and the LDCS policy proposed in [21] are both suitable for multi-channel multi-interface network environments with multiple paths and are comparable to the M-Scheduler proposed in this paper. Through comparison, one can find that the efficiency ratio (represented by ER) of the three schemes has a relationship of $ER_{\text{M-Scheduler}} > ER_{\text{LDCS}} > ER_{\text{MP}}$. Among them, LDCS adopts a relay-forwarding mechanism in the process of traffic allocation, which brings better throughput performance than the MP algorithm. Furthermore, M-Scheduler utilizes the tuple-based model to virtualize multi-channel networks into single-channel networks, which improves the transmission performance significantly. Therefore, the efficiency ratio of M-Scheduler is greater than that of LDCS. In Section 5, we conduct simulation experiments on the throughput performance of the three schemes to verify the theoretical analysis conclusions.

4.2. Implementation Complexity

From the execution process of M-Scheduler, it can be seen that it adopts a distributed implementation mechanism. Compared to centralized implementation, distributed implementation has lower complexity and is easier to deploy in distributed networks such as WSNs. Meanwhile, compared to existing multi-path distributed MAC scheduling schemes applied to wireless multi-channel networks, M-Scheduler does not need to execute the maximal matching processes and does not require iterative operations. The research shows that if the multi-channel network contains $|L|$ links and $|C|$ channels, the MAC scheduling based on the maximal matching method needs to perform $O(|L| |C|)$ iterations in each time slot. Obviously, such implementation complexity is not conducive to the deployment of the corresponding scheduling strategies in large-scale dense wireless networks. Under the proposed M-Scheduler policy, each link selects its own data transmission timestamp based on probability in each time slot. Therefore, the implementation complexity of M-Scheduler does not increase with the number of available channels and links in the network. Hence, it is said that the complexity of M-Scheduler is independent of the network scale and can be better applied to large-scale dense wireless networks.

5. Simulations

To verify the effectiveness of the multi-path MAC scheduling strategy proposed in this paper in practical network scenarios, the throughput performance of M-Scheduler is compared with the existing MP and LDCS schemes using the NS-2.31 tool. We use a grid topology as shown in Figure 6 with dashed lines representing the original links. A pair of

source/destination nodes are selected randomly in the topology. It can be seen that there are multiple alternative paths between the source and the destination. In order to better test the applicability of the algorithm in different network environments, two network scenarios are employed. In scenario 1, there are four available channels with capacities of 2, 2, 1, and 1 packets/time slot, respectively. Each node in the network is configured with 2 wireless interfaces. Under M-Scheduler, the number of divided mini time slots is set to $P = 10^3$. The simulation results in this scenario are shown in Figure 7. In scenario 2, there are 6 available channels with capacities of 5, 5, 5, 3, 3, and 2 packets/time slots. Configure 3 interfaces for each wireless node and sets the number of mini time slots for the M-Scheduler policy as $P = 10^5$. The simulation results of scenario 2 are shown in Figure 8.

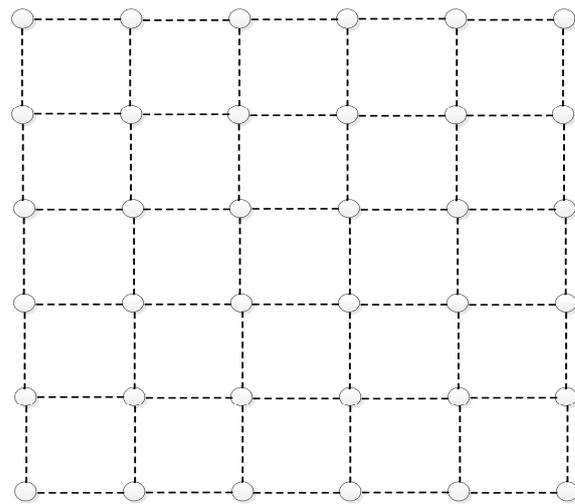


Figure 6. Network topology for simulation.

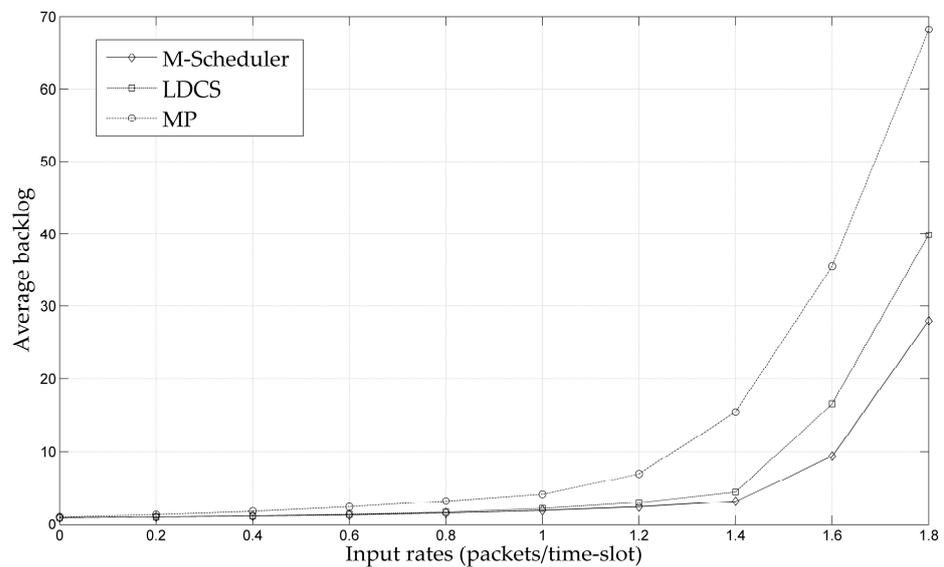


Figure 7. Average backlog versus input rates under scenario 1.

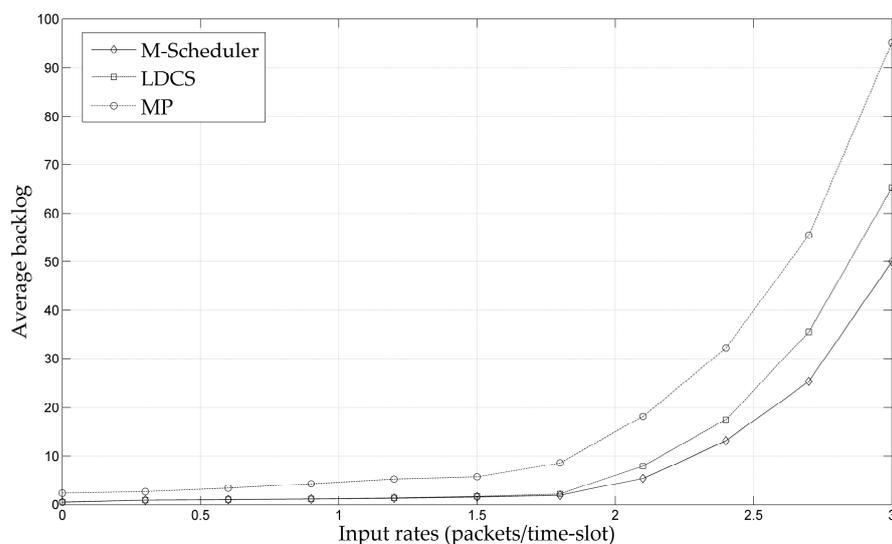


Figure 8. Average backlog versus input rates under scenario 2.

As shown in Figures 7 and 8, under the three scheduling strategies, when the input load reaches a critical value, the average backlog in the network would sharply increase. Such a critical value of the input load can be considered as the boundary of the throughput capacity region. This phenomenon indicates that the three scheduling strategies are all throughput guaranteed. It can be found that the M-Scheduler proposed in this paper has the largest throughput capacity region among these three algorithms, which is consistent with the theoretical analysis. That is because M-Scheduler virtualizes multi-channel networks into single-channel networks, resulting in improved throughput performance. At the same time, the capacity region of LDCS is larger than that of the MP strategy. The reason is that LDCS adopts a relay-forwarding mechanism to complete data traffic allocation, which can better balance the network load.

Moreover, by comparing Figures 7 and 8, it can be found that the throughput performances of the scheduling strategies in scenario 2 are better than that in scenario 1. That is because there are more available channels with higher transmission rates in scenario 2, which leads to a larger transmission bandwidth. Meanwhile, for the M-Scheduler policy, dividing more mini time slots implies better performance, which is also reflected in the simulation results. The above experiments have confirmed that the M-Scheduler proposed in this paper ensures throughput and can be deployed in practical network systems. In addition to throughput performance, M-Scheduler has significant advantages in terms of complexity, as discussed in detail earlier.

6. Conclusions

This paper put forward a novel MAC scheduling scheme for wireless multi-channel networks, named M-Scheduler. The scheme adopts a mathematical model based on tuples, which models nodes, interfaces, and channels as tuple nodes and models the link between a pair of tuple nodes as a tuple link. In this way, a multi-interface multi-channel network can be modeled as a single-channel network. At the same time, when there are multiple transmission paths from the source to the destination in the network, the proposed M-Scheduler first allocates the data traffic to each path reasonably in each time slot and adopts a random-access-based scheduling strategy to complete packet forwarding during the MAC scheduling stage. The theoretical basis for designing a data-traffic-allocation algorithm and random-access probabilities is to ensure the throughput performance of M-Scheduler. By using the Lyapunov analysis method, we have demonstrated that M-Scheduler can achieve a stable throughput capacity region. When the input load is located within the capacity region, the network system maintains a queue-stable state. In addition, due to the utilization of random-access technology to complete MAC scheduling, the implementation

complexity of M-Scheduler is independent of the network size. Such a feature makes M-Scheduler perform better in terms of complexity than the existing multi-path MAC scheduling schemes and also enhances its applicability in practical networks.

It should be noted that other types of distributed networks (such as Ad hoc) can also consider using the solution proposed in this paper. The difference lies in the relatively fixed position of nodes in WSNs and, hence, our proposed algorithm is easier to promote and apply in WSNs. In future work, we plan to transplant the M-Scheduler algorithm to the communication protocol stack of WSN terminals and collaborate with IoT companies to conduct practical experiments to verify the effectiveness of M-Scheduler and gradually promote its application in new products.

Author Contributions: Conceptualization, F.Z. and G.Y.; methodology, G.Y.; validation, F.Z. and G.Y.; formal analysis, F.Z.; investigation, G.Y.; resources, F.Z.; writing—original draft preparation, F.Z.; writing—review and editing, G.Y.; supervision, G.Y.; project administration, F.Z.; simulations, F.Z.; All authors have read and agreed to the published version of the manuscript.

Funding: This work was jointly supported by the Major Basic Research Program of the Shandong Provincial Natural Science Foundation (ZR2020ZD01) and supported in part by the Research Program of the Shandong Provincial Natural Science Foundation (ZR2022QF033), the Doctoral Research Funds of Shandong Management University (grant No. SDMUD201906), and the Research Team of Shandong Management University (Team ID: TD202306).

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Shan, C.; Xia, Y.; Guo, Z.; Wang, G.; Zhang, J. TSBS: A Two-Stage Backpressure Scheduling scheme over multihop wireless networks. *Ad Hoc Netw.* **2022**, *132*, 102874. [\[CrossRef\]](#)
2. Leonardi, E.; Mellia, M.; Neri, F.; Marsan, M.A. On the Stability of Input-Queued Switches with Speed-Up. *IEEE/ACM Trans. Netw.* **2001**, *9*, 104–118.
3. Aljubayri, M.; Yang, Z.; Shikh-Bahaei, M. Cross-layer multipath congestion control, routing and scheduling design in ad hoc wireless networks. *IET Commun.* **2021**, *15*, 1096–1108.
4. Wu, X.; Srikant, R.; Perkins, J.R. Scheduling Efficiency of Distributed Greedy Scheduling Algorithms in Wireless Networks. *IEEE Trans. Mob. Comput.* **2007**, *6*, 595–605.
5. Joo, C.; Lin, X.; Shroff, N. Understanding the Capacity Region of the Greedy Maximal Scheduling Algorithm. *IEEE/ACM Trans. Netw.* **2009**, *17*, 1132–1145.
6. Neely, M.J. Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic. *IEEE/ACM Trans. Netw.* **2009**, *17*, 1146–1159. [\[CrossRef\]](#)
7. Gupta, A.; Lin, X.; Srikant, R. Low-Complexity Distributed Scheduling Algorithms for Wireless Networks. *IEEE/ACM Trans. Netw.* **2009**, *17*, 1846–1859. [\[CrossRef\]](#)
8. Zhang, F.; Cao, Y.; Wang, D. A Note on “Low-Complexity Distributed Scheduling Algorithms for Wireless Networks”. *IEEE/ACM Trans. Netw.* **2015**, *23*, 1367–1369.
9. Xie, T.; Gao, S.; Zhao, H.; Lin, Y.; Xiong, J. A reinforcement-learning-based anti-interference resource scheduling algorithm for directed wireless communication networks. *Chin. J. Radio Sci.* **2020**, *35*, 531–541.
10. Liu, T.; Liao, W. Interference-aware QoS routing for multi-rate multi-radio multi-channel IEEE 802.11 wireless mesh networks. *IEEE Trans. Wirel. Commun.* **2009**, *8*, 166–175.
11. Xie, K.; Wang, X.; Liu, X.; Wen, J.; Cao, J. Interference-Aware Cooperative Communication in Multi-Radio Multi-Channel Wireless Networks. *IEEE Trans. Comput.* **2016**, *65*, 1528–1542. [\[CrossRef\]](#)
12. Song, Y.; Ye, D. Multi-channel transmission scheduling with hopping scheme under uncertain channel states. *J. Frankl. Inst.* **2023**, *360*, 3800–3824. [\[CrossRef\]](#)
13. Georgiadis, L.; Neely, M.J.; Tassiulas, L. Resource Allocation and Cross-Layer Control in Wireless Networks. *Found. Trends Netw.* **2006**, *1*, 1–144. [\[CrossRef\]](#)
14. Lin, X.; Rasool, S. Distributed and Provably-Efficient Algorithms for Joint Channel-Assignment, Scheduling and Routing in Multi-Channel Ad Hoc Wireless Networks. *IEEE/ACM Trans. Netw.* **2009**, *17*, 1874–1887.
15. Zhang, F.; Cao, Y. Feasible routing selection mechanism in multichannel wireless networks. *Electron. Lett.* **2014**, *50*, 402–403. [\[CrossRef\]](#)

16. Yun, M.; Zhou, Y.; Arora, A.; Choi, H.A. Channel-Assignment and Scheduling in Wireless Mesh Networks considering Switching Overhead. In Proceedings of the IEEE International Conference on Communications (ICC), Dresden, Germany, 14–18 June 2009.
17. Bhandari, V.; Vaidya, N.H. Scheduling in multichannel wireless networks. *Distrib. Comput. Netw.* **2010**, *5935*, 6–17.
18. Cheng, Y.; Li, H.; Shila, D.M.; Cao, X. A Systematic Study of Maximal Scheduling Algorithms in Multiradio Multichannel Wireless Networks. *IEEE/ACM Trans. Netw.* **2015**, *23*, 1342–1355. [[CrossRef](#)]
19. Zhang, F.; Xu, X.; Xie, L.; Liu, J. Distributed channel-assignment and scheduling algorithm for multi-channel networks. *IEICE Commun. Express* **2019**, *8*, 43–48. [[CrossRef](#)]
20. Lin, X.; Rasool, S.B. Constant-time distributed scheduling policies for ad hoc wireless networks. *IEEE Trans. Autom. Control* **2009**, *54*, 231–242. [[CrossRef](#)]
21. Xu, X.; Zhang, F.; Yang, G.; Xie, L.; Liu, Q.; Li, B. Throughput-Guaranteed Distributed Channel Assignment and Scheduling Algorithms with Low Complexity for Multichannel Wireless Sensor Networks. *J. Sens.* **2019**, *2019*, 6171625. [[CrossRef](#)]
22. Zhang, F.; Li, B.; Yang, G. Research on Throughput-Guaranteed MAC Scheduling Policies in Wireless Networks. *Entropy* **2022**, *24*, 1246. [[CrossRef](#)] [[PubMed](#)]
23. Li, M.; Salinas, S.; Li, P.; Huang, X.; Fang, Y.; Glisic, S. Optimal scheduling for multi-radio multi-channel multi-hop cognitive cellular networks. *IEEE Trans. Mob. Comput.* **2015**, *14*, 139–154. [[CrossRef](#)]
24. Konrad, C.; Naidu, K.K.; Steward, A. Maximum Matching via Maximal Matching Queries. In Proceedings of the 40th International Symposium on Theoretical Aspects of Computer Science (STACS), Hamburg, Germany, 7–9 March 2023.
25. Che, J.; Zhang, Z.; Yang, Z.; Chen, X.; Zhong, C. Massive Unsourced Random Access for NGMA: Architectures, Opportunities, and Challenges. *IEEE Netw.* **2023**, *37*, 28–35. [[CrossRef](#)]
26. Taghvaei, A.; Mehta, P.G. On the Lyapunov Foster Criterion and Poincaré Inequality for Reversible Markov Chains. *IEEE Trans. Autom. Control* **2022**, *67*, 2605–2609. [[CrossRef](#)]
27. Kodialam, M.S.; Nandagopal, T. Characterizing achievable rates in multi-hop wireless mesh networks with orthogonal channels. *IEEE/ACM Trans. Netw.* **2005**, *13*, 868–880. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.