

Research Article

Throughput-Guaranteed Distributed Channel Assignment and Scheduling Algorithms with Low Complexity for Multichannel Wireless Sensor Networks

Xinyan Xu,¹ Fan Zhang², Gangqiang Yang³, Ligu Xie,⁴ Qing Liu,⁴ and Baozhu Li⁵

¹Department of Computer and Software Engineering, Shandong College of Electronic Technology, Jinan 250200, China

²School of Information Engineering, Shandong Management University, Jinan 250357, China

³School of Information Science and Engineering, Shandong University, Qingdao 226237, China

⁴Postdoctoral Innovation Practice Base, Shandong Cable Interactive Service Ltd, Jinan 250014, China

⁵Shandong Provincial Key Laboratory of Network Based Intelligent Computing, School of Information Science and Engineering, University of Jinan, Jinan 250022, China

Correspondence should be addressed to Fan Zhang; ccfov@126.com and Gangqiang Yang; g37yang@uwaterloo.ca

Received 18 October 2018; Revised 8 August 2019; Accepted 22 August 2019; Published 14 October 2019

Academic Editor: Qiang Wu

Copyright © 2019 Xinyan Xu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks, an improved throughput capacity region can be achieved by equipping multiple channels. However, such approach inevitably brings the issue of solving the coupled channel assignment and scheduling problem. This paper put forward a low-complexity distributed channel assignment and scheduling policy for multichannel wireless sensor networks with single-hop traffic flows, named LDCS, as well as its multihop multipath extension. Under the proposed algorithms, random access and backoff time techniques are introduced to keep the complexity low and independent of the number of links and channels. Through theoretical analysis and simulation experiments, it is proved that the proposed algorithms are throughput guaranteed, and in some network scenarios, the achieved capacity region can be larger than that of other comparable distributed algorithms.

1. Introduction

In wireless networks, datagrams are transmitted from source to destination through a routing selection mechanism in network layer associated with scheduling in MAC layer. When the paths for all users are fixed, scheduling policies play a key role in utilizing limited bandwidth effectively to achieve better throughput performance. There have been a number of findings and research results on the topic of improving throughput capacity region via designing efficient scheduling algorithms. It has been demonstrated that the maximum capacity region can be guaranteed by the throughput-optimal scheduling algorithms such as max-weight scheduling [1] for single-path traffic and back-pressure scheduling [2] for multipath cases. However, such policies require centralized management along with great limitation. The centralized scheduling algorithms require management node

or base station in the network to collect information flow and carry out global control, and as a consequence, it is difficult to implement in many network scenarios such as wireless sensor networks (WSN) and ad hoc networks (MANET).

In order to enhance the adaptability, a series of distributed maximal scheduling (MS) algorithms that are more applicable in practice have been proposed under various interference models [3, 4]. So-called “distributed” refers to the behavior of network nodes deciding to forward data or not according to their own conditions and the information obtained from neighborhood without centralized control. It has been proved that MS is throughput guaranteed. That is, it can achieve at least a certain fraction of the maximum throughput capacity region. Even so, such maximal-matching-type algorithms need to execute a maximal matching finding process at each time slot, which requires a remarkable number of iterations increasing logarithmically

with the number of links in the system [5] and consequently causes nontrivial implementation complexity. Considering such drawback, designing distributed scheduling policies with low complexity is worth further research. The authors in [6] presented a distributed Q-SCHED algorithm based on random access and backoff time techniques [7], whose throughput performance could be arbitrarily close to that of MS. The complexity of Q-SCHED is low and independent of the network size, which argues its applicability for large-scale network systems.

The achievements mentioned above are all acquired on single-channel conditions. In fact, the application of multichannel technology can significantly improve the throughput performance of WSN. At present, IEEE 802.11 series has provided resource base for the application of multichannel, such as 802.11a with 12 nonoverlapping channels operating on 5 GHz band and 802.11 b/g with 3 channels on 2.4 GHz frequency band. In multichannel environments, channel assignment and scheduling are usually combined with each other to jointly affect the throughput performance [8–10], which poses a technical challenge for designing distributed algorithms that solve the coupled channel assignment and scheduling problem. Through in-depth research, we find that many single-channel scheduling algorithms cannot be extended straightforward to multichannel networks because such extension may lead to poor throughput performance. Due to the existence of channel diversity, the designing of distributed channel assignment and scheduling strategy in multichannel networks is much more complicated than that in single-channel scenarios. Until now, there have been a series of research results on this issue. However, these existing results still have problems to be solved, such as complexity. Obviously, the application of multichannel in wireless networks is bound to bring an increased implementation complexity of the distributed algorithms.

In view of the above-discussed facts, the goal (or the main contribution) of this paper is to devise distributed algorithms for multichannel wireless networks with low complexity. We present a channel assignment and scheduling algorithm, named LDCS, for cases with single-interface nodes. In order to keep the complexity low, LDCS extends the idea of the above-mentioned Q-SCHED to fit into multichannel circumstances. It is theoretically proved that LDCS is throughput guaranteed, and the achieved capacity region can be larger than that of other comparable distributed algorithms in a variety of scenarios. After that, we extend the policy to the cases of multipath routing with multihop flows. The simulation experiments verify our theoretical analysis.

2. Related Work

In multichannel scenarios, it is truly difficult to guarantee the throughput performance (characterized by capacity region) in a distributed manner. The objective of this issue is to maintain the system stable with finite queue lengths inside a specific capacity region. The implementation of scheduling algorithms should be combined with data flow allocation and channel assignment because of channel diversity. Toward this direction, in [9], a provably efficient

algorithm (referred therein as SP) and its multipath extension MP have been developed for multichannel multi-interface (MC-MI) networks. SP and MP were proved to be throughput guaranteed, which implies that they can ensure a certain fraction of the maximum capacity region. In SP and MP, each link maintains multiple per-channel queues as well as one common link queue. Arriving packets first enter the common link queue and are then assigned to each channel via executing a data flow allocation mechanism, as illustrated in Figure 1. According to the per-channel queues, maximal scheduling is carried out to determine the set of forwarding queues. Such relay-forwarding-based process definitely has a negative impact on system performance. At the same time, aiming at achieving reliable throughput capacity region under SP or MP, each link collects queue length and channel rate information from neighborhood before allocating data flow, which leads to extra overhead inevitably. In this paper, we try to adopt a more concise way to implement effective configuration of data traffic.

Besides SP and MP, for the purpose of solving the coupled resource allocation problem systematically in MC-MI networks, Cheng et al. raised another method, named tuple-based MS [11]. In addition, a novel model was put forward in [11] under which the original nodes, interfaces, and channels are transformed into multiple node-radio-channel tuples. A simple example is shown in Figure 2, where the original link l is composed of node m and n while the mapped tuple link η is formed by tuple (m, i, c) and (n, j, c) . Such framework enables the single-channel MS algorithm to be extended straightforward to MC-MI wireless networks with guaranteed throughput performance. Compared with SP, the tuple-based MS has been shown to ensure a larger capacity region with lower average backlog. However, it could be observed that, with regard to SP, MP, and tuple-based MS, they all necessarily attempt to operate a maximal matching process during each time slot, which incurs significant complexity. In fact, at least $O(\log |L||C|)$ number of iterations have to be conducted for the purpose of obtaining a maximal schedule [12], where $|L|$ and $|C|$ denote respectively the number of links and channels in the network. In order to reduce the complexity, we employ random access and backoff time techniques.

In recent years, a series of research results on channel allocation or scheduling in multichannel networks have emerged from other different perspectives. Authors of [13] developed a DES-Chan framework for distributed channel assignment. However, such framework did not address the scheduling issue or delve into throughput performance. Some other efforts have been made on designing algorithms for OFDM-based multichannel downlink relay networks [14–16]. However, in OFDM-based systems, each channel can only be occupied by one link at each time slot because of opportunism. By comparison, our results apply to more general scenarios where different links can operate on the same channel provided that they do not interfere with each other. In addition, there have emerged some appealing results focusing on centralized schemes with low delay or complexity [17, 18]. Differently, we only consider the online

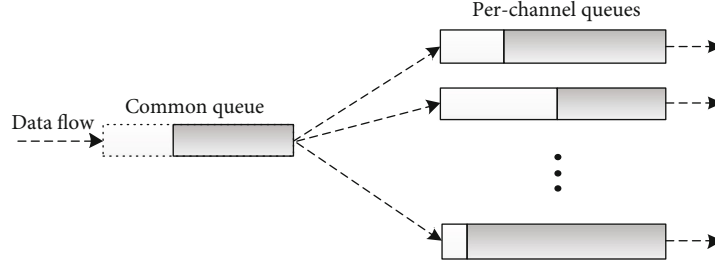


FIGURE 1: The relay-forwarding framework in SP and MP.

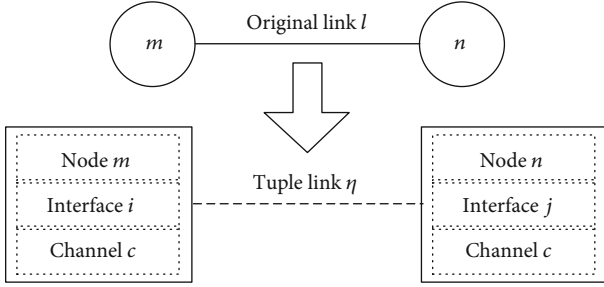


FIGURE 2: Transformation process involved in the tuple-based model.

fashion that is more suitable for distributed WSN in this paper.

The rest of the paper is organized as follows. System model and notations are introduced in Section 3. In Section 4, we propose the LDCS algorithm and prove that it can guarantee a certain fraction of the maximum throughput capacity region with low complexity. We next extend our results to multihop multipath cases in Section 5. Simulation results are given in Section 6. Finally, we conclude in Section 7.

3. System Model and Notations

We consider a single-interface multichannel wireless network where each node is equipped with one interface which is able to switch among channels dynamically if necessary. Let L and C denote respectively the set of all links and available non-overlapping channels in the network. For any set Γ , $|\Gamma|$ refers to the cardinality of Γ . With single-hop data flows, each link represents a transmission path for a pair of source and receiver. Radio packets leave the system once they reach the destination. The time is slotted and synchronized for all links in the network. During each time slot, channel assignment and scheduling policy jointly decide the sets of scheduled links and occupied channels. Relevant definitions and terminologies are introduced now.

With regard to the collision model, each link l corresponds to an interference set I_l that includes links interfering with l over the same channel and let $l \in I_l$. If l and another link in I_l operate on the same channel simultaneously, neither of them transmits successfully. Assume the interference relationship is symmetric, i.e., $l \in I_k$ if and only if $k \in I_l$. Denote by K the maximum number of links that can be scheduled on a given channel simultaneously within the interference set of arbitrary link. Due to the single-interface

configuration, adjacent links cannot transmit simultaneously even if they are scheduled on different channels. Thus, one can split I_l into two mutually exclusive subsets, denoted respectively by $I_{\text{Int}}(l)$ and $I_{\text{Cha}}(l)$. Specifically, $I_{\text{Int}}(l)$ is composed of links that are adjacent to l (including l itself) and $I_{\text{Cha}}(l)$ is formulated by $I_{\text{Cha}}(l) = I_l \setminus I_{\text{Int}}(l)$. Here, define K_1 as the maximum number of links that can be scheduled at the same time within $I_{\text{Int}}(l)$ of any link l . Similarly, within the set $I_{\text{Cha}}(l)$ of any link l , use K_2 to represent the maximum number of links that can be scheduled on arbitrary channels simultaneously. In this paper, $I_{\text{Int}}(l)$ and $I_{\text{Cha}}(l)$ are respectively named by interface and channel interference set while $K/K_1/K_2$ are named respectively by link/interface/channel interference degree.

The transmission rate (or capacity) of link l on channel c is denoted by r_l^c . It is assumed that $r_l^c > 0$ for any l and c . Because of the presence of channel diversity, every link has a unique rate on each channel. Denote by $A_l(n)$ the number of arrivals at link l in time slot n , the arrival process is then defined as $\{A_l(n)\}$. We make some simple assumption that the arrival process $\{A_l(n)\}$ is i.i.d. across time with the mean value λ_l . In fact, our method can be extended to suit more general arrival processes. For simplicity, this paper focuses only on situations with the assumption mentioned above. To store data packets, each link l needs to maintain one channel queue for every channel c , denoted by (l, c) . One can obtain from the description above that the set of channel queues interfering with (l, c) denoted by $\varepsilon_{(l,c)}$ is as

$$\varepsilon_{(l,c)} = \{(k, c): k \in I_{\text{Cha}}(l)\} \cup \left\{ (j, c'): j \in I_{\text{Int}}(l), c' \in C \right\}. \quad (1)$$

Denote the queue length of (l, c) by $q_l^c(n)$. Arriving packets are allocated immediately to multiple channel queues using some data-flow-allocation algorithm. Let $A_l^c(n)$ be the number of datagrams assigned to (l, c) , and the expectation is denoted by $\lambda_l^c = E[A_l^c(n)]$. Thus, the evolution of the queue length is given as

$$q_l^c(n+1) = [q_l^c(n) + A_l^c(n) - D_l^c(n)]^+, \quad (2)$$

where $D_l^c(n)$ denotes the number of datagrams transmitted by (l, c) in time slot n based on the channel assignment and scheduling algorithm. It is easy to find that $D_l^c(n) = r_l^c$ if (l, c) is scheduled at n ; otherwise, $D_l^c(n) = 0$. For convenience, we

TABLE 1: Table of all notations.

Variable	Notation
L, C, S	Set of all links, channels, and users
I_l	Set of links interfering with l over the same channel
$I_{\text{Cha}}(l), I_{\text{Int}}(l)$	Channel/interface interference set
K, K_1, K_2	Link/interface/channel interference degree
r_l^c	Rate of link l on channel c
$A_l(n), A_s(n)$	Number of packets arriving at link l and user s at time slot n
λ_l, λ_s	Average packet arriving rate on link l and user s
(l, c)	Channel queue maintained by link l on channel c
$\mathcal{E}_{(l,c)}$	Set of all channel queues that interfere with (l, c)
$q_l^c(n)$	Queue length of (l, c)
$A_l^c(n)$	Number of packets assigned by link l to channel c at time slot n
λ_l^c	Expectation of $A_l^c(n)$
$D_l^c(n)$	Number of packets served by (l, c) in time slot n
Ω, Ω'	Maximum capacity region for single-hop and multihop cases
γ, γ'	Efficiency ratio of algorithms LDCS and M-LDCS
H_{si}^l	Routing matrix
$P_{si}(n)$	Fraction of data packets assigned to path i by user s

will use the term “queue” to represent the “channel queue” in what follows. We say the network system is stable if for any $v > 0$, there exists a constant $\Psi > 0$ such that [19]

$$\lim_{n \rightarrow \infty} \Pr \left\{ \sqrt{\sum_{l \in L} \sum_{c \in C} [q_l^c(n)]^2} > \Psi \right\} < v, \quad (3)$$

where $\Pr\{S\}$ represents the probability of the event S . System stability implies that network can bear current input load. It can be obtained that the system is stable and all the queue lengths remain finite if the Markov chain $\{q_l^c(n): l \in L, c \in C\}$ is positively recurrent.

The capacity region achieved by a particular channel assignment and scheduling algorithm is defined as the set of λ that stabilizes the network system, where $\lambda = [\lambda_1, \lambda_2, \dots, \lambda_L]$ is exactly the input load. As we know, the maximum capacity region Ω can be obtained by the throughput optimal scheduling [17] in a centralized way. Under a distributed algorithm, if the queue lengths remain finite for any $\lambda \in \gamma\Omega$ with constant $\gamma \in (0, 1]$, the efficiency ratio of this algorithm is then denoted by γ . The key variables in this paper are summarized in Table 1.

4. LDCS and Performance Analysis

At each time slot in LDCS, packets are assigned to channel queues immediately after arriving based on our data flow

allocation mechanism without extra relay-forwarding common queue, which leads to a reduced transmission delay [17]. For channel assignment and scheduling, LDCS employs the random access and backoff time techniques (the idea of above-mentioned Q-SCHED) instead of executing maximal matching process. The LDCS algorithm consists of two parts given as follows.

4.1. Data Flow Allocation. For data flow allocation, a rate-proportional-based policy is designed as

$$A_l^c(n) = A_l(n) \frac{r_l^c}{\sum_{d \in C} r_l^d}, \quad (4)$$

which can be realized by using a probability approach. It is easy to see that the process $\{A_l^c(n)\}$ is also i.i.d. across time slot. The allocation process is implemented locally and is shown in Figure 3.

In LDCS, each link does not require information of other links in the data flow allocation phase, which can decrease the overhead effectively compared with the existing SP policy [9].

4.2. Channel Assignment and Scheduling. After data allocation, LDCS then utilizes its channel assignment and scheduling mechanism to make the decision whether a queue should be scheduled or not. Following the idea of Q-SCHED [6], the random access and backoff time techniques are employed. To be specific, each time slot is divided into two subslots: scheduling slot and transmission slot, either of which has a fixed length. The scheduling slot is further divided into M minislots. Such time slot division is shown in Figure 4. According to the backoff time technique, each queue (l, c) selects a minislot with probability as

$$\begin{cases} \Pr\{I_l^c = M + 1\} = e^{-p_l^c}, \\ \Pr\{I_l^c = m\} = e^{-p_l^c(m-1/M)} - e^{-p_l^c(m/M)}, \quad m = 1, 2, \dots, M, \end{cases} \quad (5)$$

where I_l^c represents the backoff time picked by (l, c) and the parameter p_l^c is computed as

$$p_l^c = \alpha \frac{q_l^c(n)/r_l^c}{\max_{(j,d) \in \mathcal{E}_{(l,c)}} \left[\sum_{(k,c') \in \mathcal{E}_{(j,d)}} q_k^{c'}(n)/r_k^{c'} \right]}, \quad (6)$$

where $\alpha = \log M$. In formula (5), if a queue chooses $M + 1$ as its backoff time, it will remain static in the current time slot. Instead of waiting for the entire scheduling slot, a queue enters the transmission slot to send messages immediately after the selected backoff time expires as long as no one in its interference set has already picked smaller backoff time. If two or more queues select the same minislot, all of them fail to accomplish the transmission successfully because of collision.

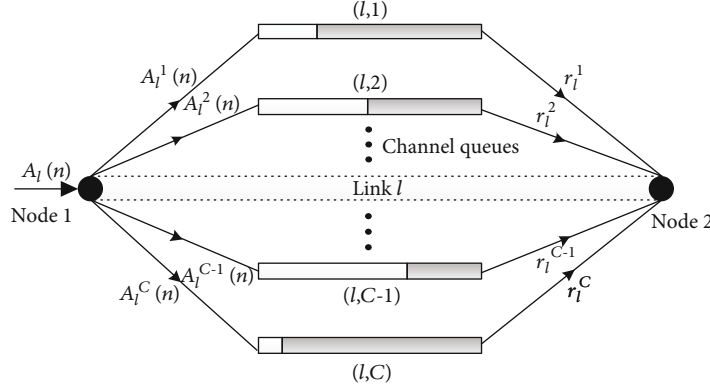


FIGURE 3: Data flow allocation based on channel queues.

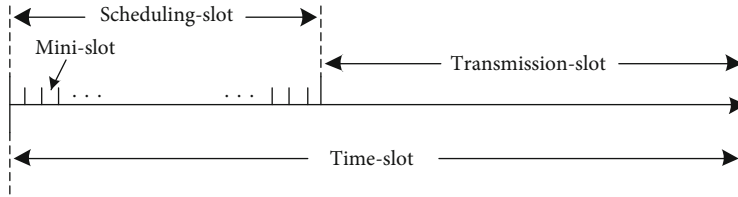


FIGURE 4: The time slot division under the proposed LDCS.

4.3. Stability Region. For the purpose of analyzing stability, we introduce two lemmas and construct the Lyapunov function as

$$V(n) = \max_{l \in L, c \in C} \left[\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right]. \quad (7)$$

Lemma 1. At each time slot n , there exists a constant $G > 0$ such that if $V(n) \geq G$, then for any queue (j, d) satisfying

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} \geq \vartheta(V(n) - C_1 - C_2\mu), \quad (8)$$

the LDCS policy guarantees

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \Pr\{S_l^c\} \geq \vartheta \left(1 - \frac{\log M + 1}{M} - \mu \right), \quad (9)$$

for any $\mu > 0$, $\vartheta \in [0, 1]$ and constants $C_1, C_2 > 0$, where S_l^c denotes the event that (l, c) is scheduled. The proof of Lemma 1 is omitted here. One can refer to [6] (proof of Lemma 1 therein) to derive Lemma 1 easily based on (5) and (6).

Lemma 2. Under LDCS, if the input load meets

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{\lambda_l^c}{r_l^c} < 1 - \frac{\log M + 1}{M} - 4\mu, \quad (10)$$

for every queue (j, d) , then there will exist some integer $H > 0$ and a constant $\Phi > 0$ such that with $V(n) \geq \Phi$,

$$\Pr \left\{ \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n+H)}{r_l^c} \leq V(n) - H\mu \right\} \geq 1 - \varsigma \quad (11)$$

holds for any queue (j, d) and any $\varsigma > 0$. We present the proof in the appendix. Based on the two lemmas above, one can derive the following theorem which indicates the condition for system stability.

Theorem 1. Under LDCS, if for some $\mu > 0$, the input rates satisfy

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{\lambda_l^c}{r_l^c} < 1 - \frac{\log M + 1}{M} - 4\mu, \quad (12)$$

for any queue (j, d) , the irreducible aperiodic discrete state Markov chain $\{q_l^c(n) : l \in L, c \in C\}$ is then positively recurrent. One can refer to [6] for the detail of the proof of Theorem 1. Hence, we obtain the capacity region guaranteed by LDCS using the following proposition.

Proposition 1. The efficiency ratio γ of LDCS is given as

$$\gamma = \frac{\tau}{K_2 + K_1|C|} \left(1 - \frac{\log M + 1}{M} - 4\mu \right), \quad (13)$$

where μ is a positive number with an arbitrary value and

$$\tau = \min_{l \in L} \left[\frac{\sum_{c' \in C} r_l^{c'}}{\max_{b \in C} r_l^b} \right]. \quad (14)$$

Proof. Based on the relevant definitions described in Section 3, the achieved capacity region of LDCS is said to be $\gamma\Omega$ if for any input load λ such that the centralized optimal algorithm can stabilize the system at $\lambda\gamma$, the LDCS can keep the system stable with λ . Under our scenarios, the necessary condition for system stability at $\lambda\gamma$ is that there exists some $\tilde{x}_l^c \in [0, r_l^c]$ for each (l, c) such that

$$(1 + \partial)^2 \frac{\lambda_l}{\gamma} \leq \sum_{c \in C} \tilde{x}_l^c, \quad (15)$$

$$\sum_{k \in I_{\text{Cha}}(l)} \sum_{c \in C} \frac{\tilde{x}_k^c}{r_k^c} \leq K_2, \quad (16)$$

$$\sum_{k \in I_{\text{Int}}(l)} \sum_{c \in C} \frac{\tilde{x}_k^c}{r_k^c} \leq K_1 \quad (17)$$

hold for any link l with parameter $\partial > 0$ [10]. Note that inequality (15) comes from the rate constraint, while (16) and (17) represent the channel and interface restriction, respectively. Set a parameter $\beta = (1 + \partial)^2 / \gamma$ and replace γ with the right side of (13), we have according to (4) and the inequalities (15)–(17) that for all (l, c) and some $\mu > 0$,

$$\begin{aligned} \sum_{(k, c') \in \mathcal{E}(l, c)} \frac{\lambda_k^{c'}}{r_k^{c'}} &= \sum_{k \in I_{\text{Cha}}(l)} \frac{\lambda_k^c}{r_k^c} + \sum_{k \in I_{\text{Int}}(l)} \sum_{c' \in C} \frac{\lambda_k^{c'}}{r_k^{c'}} \\ &\leq \frac{1}{\tau \beta} \left[\sum_{k \in I_{\text{Cha}}(l)} \sum_{c' \in C} \frac{\tilde{x}_k^{c'}}{r_k^{c'}} + |C| \sum_{k \in I_{\text{Int}}(l)} \sum_{c' \in C} \frac{\tilde{x}_k^{c'}}{r_k^{c'}} \right] \\ &\leq \frac{1}{\tau \beta} [K_2 + |C|K_1] < 1 - \frac{\log M + 1}{M} - 4\mu. \end{aligned} \quad (18)$$

Hence, Proposition 1 has been proved using Theorem 1.

4.4. Performance Analysis. From (13), one can infer that with $M \rightarrow \infty$ and $\mu \rightarrow 0$, the efficiency ratio of LDCS policy arbitrarily approaches $\tau/(K_2 + K_1 |C|)$. Here, the parameter μ can be selected close to 0. Due to the fixed length of scheduling slot, M tend to infinity if we let the length of each minislot be small enough by improving hardware. Therefore, it can be argued that the developed LDCS can achieve a throughput capacity region of $\tau\Omega/(K_2 + K_1 |C|)$ approximately.

Then, we compare the throughput performance of LDCS with other existing distributed algorithms. According to [11], tuple-based MS attains an efficiency ratio of $1/\mathcal{Y} > 1/(K + 2)$, where \mathcal{Y} denotes the interference degree over the tuple-based equivalent model and $1/(K + 2)$ is the efficiency ratio of SP algorithm [9]. Hence, one can conclude that as $M \rightarrow \infty$, the capacity region of LDCS is larger than that of the tuple-based MS or SP algorithm on condition that the network sce-

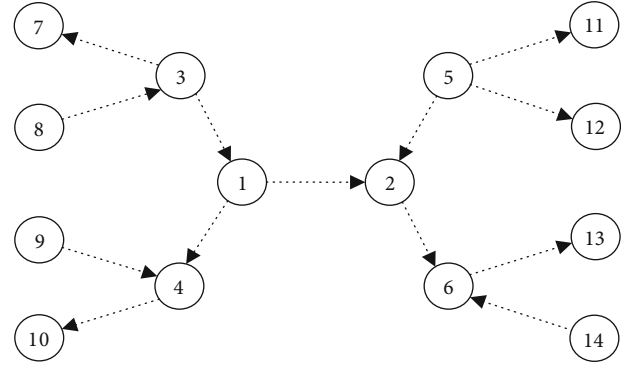


FIGURE 5: Network topology for illustration with single-hop flows.

nario satisfies $\tau/(K_2 + K_1 |C|) > 1/\mathcal{Y} > 1/(K + 2)$. A simple topology in Figure 5 with single-hop flows and 2-hop interference model is illustrated. Suppose there are 14 nodes and 4 available channels with rates equalling 1, 1, 2, and 2 packet-slot/time slot, respectively, for all links. Thus, it is not hard to figure out that the efficiency ratio of LDCS equals to $1/4$, whereas that of tuple-based MS and SP are, respectively, $1/5$ and $1/6$, we have $1/4 > 1/5 > 1/6$. Therefore, in practice, compared with SP and tuple-based MS, better throughput performance can be obtained by LDCS as long as the network scenario satisfies $\tau/(K_2 + K_1 |C|) > 1/\mathcal{Y} > 1/(K + 2)$.

On the other hand, the complexity of LDCS is low and independent of the number of links and channels. In LDCS, each node calculates the backoff time according to equations (5) and (6), which incurs $O(K_1 + K_2)$ number of iterations [12]. In most WSN scenarios in practice, the complexity of LDCS is bounded since parameters K_1 and K_2 of the network are bounded [8]. In contrast, recall that the complexity of the existing maximal-matching-based algorithms (SP or tuple-based MS) [9, 11] is $O(\log |L||C|)$ which increases logarithmically with the numbers of links and channels. From the perspective of computation time, based on the analysis above, LDCS merely requires a scheduling slot (or M minislots) to compute a schedule, which has no concern with both the size of network and the number of channels. With the improvement of hardware performance in the future, the length of a minislot can be set arbitrarily small so that the length of scheduling slot converges to a small positive number even M approaches infinity. Contrastively, for the maximal-matching-based algorithms with $O(\log |L||C|)$ complexity, the computation time of each schedule bounds to increase with the number of links and channels. Therefore, in large-scale intensive multichannel networks, the maximal-matching-based algorithms inevitably need longer time than LDCS to complete a schedule because of their inherent maximal matching mechanism [6, 7]. Currently, due to the limited hardware performance, we cannot set a minislot at an arbitrarily small value, and hence, this paper does not quantitatively compare the computation time of LDCS with that of maximal-matching-based methods.

It is worth mentioning that LDCS only ensures the weakly stability [19] of the system with a possibly unbounded delay experienced by customers and it is designed for multichannel wireless networks with single-interface wireless nodes.

5. Multipath Extension

The discussion above focused on the single-hop flows. For multihop flows with multiple paths, a routing selection problem needs to be resolved to ensure effectiveness and fairness. For multihop multipath cases, the arrival process and the maximum capacity region are redefined as follows. Suppose a system composed of S users with $I(s)$ alternate paths for each user s . Let $A_s(n)$ denote the number of packets offered by user s at time slot n . It is further assumed that $A_s(n)$ is i.i.d. across time slot with average λ_s . Define H_{si}^l as the routing indicator variable. If path i of user s passes through link l , $H_{si}^l = 1$; otherwise, $H_{si}^l = 0$. Thus, the arrivals at link l in time slot n can be given by

$$A_l(n) = \sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l A_s(n) P_{si}(n), \quad (19)$$

where $P_{si}(n)$ denotes the fraction of data packets assigned to path i by user s during time slot n . The extended maximum throughput capacity region Ω' for multihop scenes with multiple paths is redefined as the set of $[\lambda_1, \lambda_2, \dots, \lambda_S]$ such that there exists for all user s and path i satisfying

$$\begin{aligned} & \left[\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \bar{P}_{si} \right] \in \Omega \\ & \text{s.t.} \quad \sum_{i=1}^{I(s)} \bar{P}_{si} = 1, \end{aligned} \quad (20)$$

where \bar{P}_{si} can be interpreted as the average fraction of traffic assigned to path i on a long-term basis and Ω has been defined by link-based rate vectors λ in Section 3. We now combine the MAC layer LDCS policy with a routing selection mechanism so that the joint routing, channel assignment, and scheduling algorithm, which is called M-LDCS, can guarantee a certain fraction of the extended maximum capacity region Ω' . The same data flow allocation strategy as LDCS is utilized and the joint algorithm is performed in two steps as

Step 1. Each source first computes the fraction vector $P_s(n) = [P_{s1}(n), P_{s2}(n), \dots, P_{sI(s)}(n)]$ at each time slot n through solving the following problem

$$\begin{aligned} & \max_{P_s(n)} - \frac{\varphi_s}{2} \sum_{i=1}^{I(s)} [P_{si}(n)]^2 - \sum_{i=1}^{I(s)} P_{si}(n) \sum_{l \in L, c \in C} \frac{H_{si}^l}{\sum_{b \in C} r_l^b} \sum_{(j,d) \in \mathcal{E}_{(l,c)}} \\ & \quad \cdot \left[\sum_{(k,c') \in \mathcal{E}_{(j,d)}} \frac{q_k^{c'}(n)}{r_k^{c'}} \right]^\theta \\ & \text{s.t.} \quad \sum_{i=1}^{I(s)} P_{si}(n) = 1, \quad P_{si}(n) \geq 0, \end{aligned} \quad (21)$$

for some $\theta > 0$, where φ_s is a positive constant. In fact,

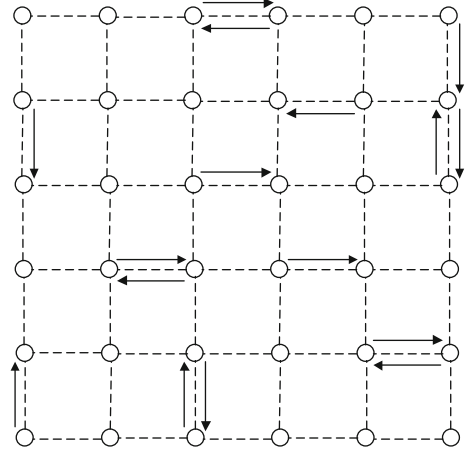


FIGURE 6: Network topology for experiments.

the transmission path is selected by (21). When a data packet is generated by user s , it will be assigned to path i with probability $P_{si}(n)$.

Step 2. The same channel assignment and scheduling mechanism as in Section 4 is employed to determine the set of operating queues and the queue is updated by

$$q_l^c(n+1) = \left[q_l^c(n) + \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l A_s(n) P_{si}(n)}{\sum_{b \in C} r_l^b} - D_l^c(n) \right]^+. \quad (22)$$

Note that the first term of (21) is to avoid an oscillation problem in routing selection process [20]. In the following proposition, the efficiency ratio of M-LDCS is given.

Proposition 2. For any $\mu > 0$, there exists some $\theta_0 > 0$ such that for any $\theta > \theta_0$, the joint routing and LDCS algorithm can ensure $\gamma' \Omega'$ capacity region where

$$\gamma' = \frac{(1-\mu)\tau}{(1+\mu)(K_2 + K_1|C|)} \left(1 - \frac{\log M + 1}{M} \right). \quad (23)$$

The proof is presented in the appendix.

6. Simulation Results

In this section, we evaluate the performances of the proposed algorithms through simulation experiment using NS2-2.31. The topology we adopt is shown in Figure 6 which contains 36 nodes (denoted by circles), 60 links (denoted by dashed lines), and 4 available channels with rates equalling 1, 1, 2, and 2 packets/time slot, respectively. Each node is configured with a single interface that can switch among channels without hindrance. 2-hop interference model is used for simulations. In order to attain the throughput capacity region, we gradually increase the input rates (or offered loads) and observe the experimental data of average backlog. The unit of input rate is packets/time slot. We first consider the single-hop scenario and compare the developed LDCS

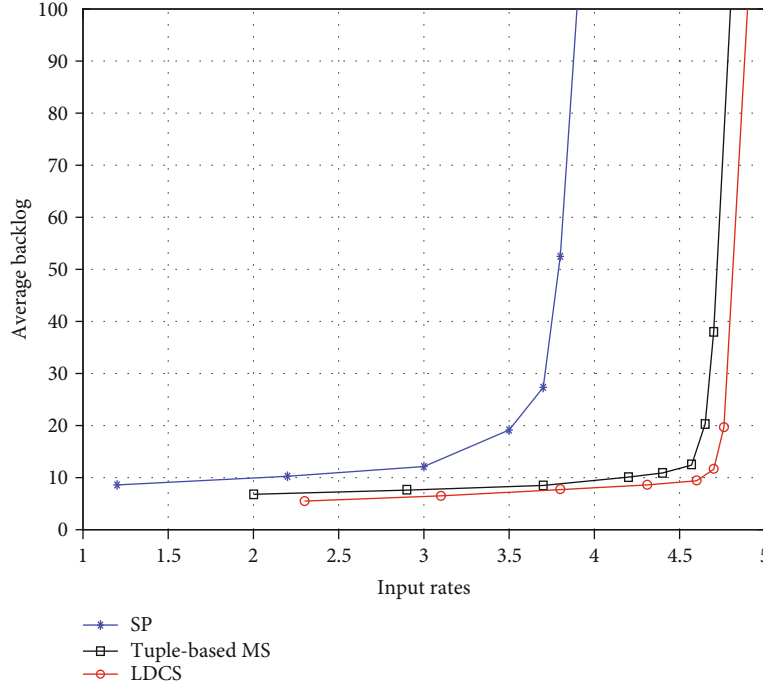


FIGURE 7: Average backlog versus input rates in single-hop scenario.

($M = 10^3$) with existing SP and tuple-based MS policies. As illustrated in Figure 6, there exist 16 traffic flows (represented by arrows) in our environment. Assume that the input rates of all flows are the same, denoted by λ . The average backlog is defined as the mean total backlog in the system divided by the number of traffic flows, which is used to reflect the throughput performances of the evaluated algorithms. In the simulation, the extra control overhead is omitted so that the throughput performance difference of the algorithms can be clearly shown.

Figure 7 shows the performance comparison of the three protocols. As we can see from the figure, along with the increasing of offered load, queues and the network become accordingly congested. At the same time, average backlog grows sharply to infinity when λ approaches a certain value. Such inflection point can be regarded as the edge of the throughput capacity region achieved by the corresponding algorithms. In addition, Figure 7 shows the superiority of the proposed LDCS in terms of throughput performance under our scenario, as expected. In fact as analyzed in Section 4.4, it can be said that LDCS guarantees larger capacity region than SP and tuple-based MS provided that $\tau/(K2 + K1 |C|) > 1/\mathcal{Y}$. The model we use for experiment is one of the scenarios who satisfy such condition.

We next investigate the performance of the multihop multipath extension (ME) of LDCS using the same topology in Figure 8. Differently, let four source-destination pairs randomly picked across multiple hops replace the 16 single-hop flows. Assume that there exist three alternative paths for each user. Source node selects forwarding path for every data packet according to (21). We let θ be 10^3 to make sure μ is small enough based on Proposition 2 and set $\varphi_s = 10^5$ so that the route fraction vectors calculated

by (21) are not too sensitive to the queue length updates. In this simulation, the comparison objects are MP and the ME of tuple-based MS. In practice, multipath extension implies a cross-layer control method. That is, for the purpose of ensuring throughput, link layer and network layer exchange information to solve a joint routing, channel assignment, and scheduling problem. Figure 8 displays the comparison of three solutions. As expected, ME of LDCS performs much better than MP since it achieves larger capacity region, which confirms theoretical analysis. As we can observe, the capacities attained by ME of LDCS and tuple-based MS are fairly close. Furthermore, for the same input rates, ME of tuple-based MS achieves the lowest average backlog. The reason for this phenomenon is that ME of tuple-based MS additionally considers transmission delay and solve the cross-layer problem with path selection convex optimization technique. For MP, it utilizes a two-stage queuing structure that stores datagrams in a common queue firstly and relays them to channel queues, which aggravates congestion and leads to imperfect performance.

It is worth noting that the number of interfaces and channels would impact the system capacity region inevitably. This paper purely focuses on single-interface scenarios. In reality, most existing communication or sensor systems employed single-interface nodes at present.

7. Conclusions

This paper presents distributed algorithms for multichannel single-interface wireless sensor networks with both single-hop and multihop multipath scenes, named, respectively, LDCS and ME of LDCS. It is theoretically demonstrated that the proposed algorithms can achieve guaranteed

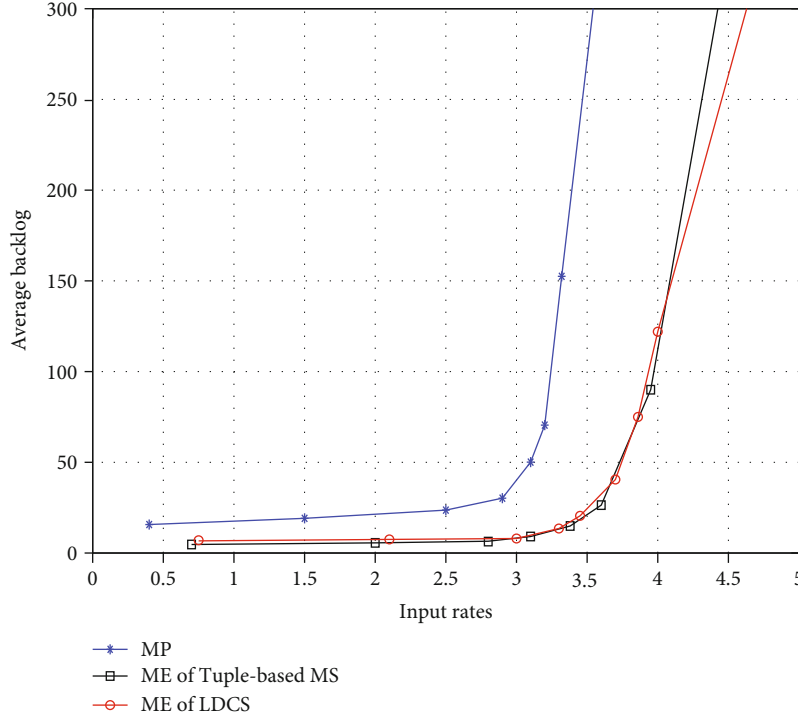


FIGURE 8: Average backlog versus input rates in multihop multipath scenario.

throughput capacity regions that are comparable with other distributed maximal-matching-based algorithms (such as SP, MP, and tuple-based MS) with low implementation complexity.

LDCS utilize a rate-proportional-based mechanism to allocate datagrams locally and applies random access technique based on probability to complete the channel assignment and scheduling. Such designing avoids executing maximal-matching-finding process and attains lower complexity that is independent of the numbers of links and channels. It is further theoretically proved that LDCS can achieve better throughput performance in some specific situations compared with other existing policies. LDCS is then extended to be appropriate for multihop multipath scenarios and a cross-layer solution is given. Simulation experiments confirm our analysis. In the future research, we will be interested in improving LDCS to be able to maintain the network under a strongly stable status with bounded average delay for both single-interface and multi-interface cases.

Appendix

A.1. Proof of Lemma 2

Because the arrivals and departures are both upper bounded during one time slot, then there must exist a positive number B such that

$$\left| \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n+1)}{r_l^c} - \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} \right| \leq B \quad (\text{A.1})$$

holds for any (j, d) . Here, we discuss in two cases. If (j, d) satisfies

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} \leq V(n) - H(B + \mu), \quad (\text{A.2})$$

it is then easy to get that Lemma 2 holds. For the other case when

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} > V(n) - H(B + \mu), \quad (\text{A.3})$$

for any $t \in \{n+1, \dots, n+H\}$, we have

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(t)}{r_l^c} > V(n) - H(B + \mu) - B(t - n) \geq V(t) - H(3B + \mu). \quad (\text{A.4})$$

Employ Lemma 1 and let $\vartheta = 1$, one can draw a conclusion that there exists a constant $\Phi > 0$ such that if $V(n) \geq \Phi$, then

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \Pr\{S_l^c\} \geq 1 - \frac{\log M + 1}{M} - \mu \quad (\text{A.5})$$

holds for all $t \in \{n+1, \dots, n+H\}$. Clearly, Φ is chosen as a function of H . Define an indicator variable $M_{(l,c)}(t)$ such that $M_{(l,c)}(t) = 1$ in case of queue (l, c) is scheduled

in time slot t , and otherwise, $M_{(l,c)}(t) = 0$. Then, from (A.5), a probability model can be described by

$$\begin{aligned} E \left[\sum_{(l,c) \in \mathcal{E}_{(j,d)}} M_{(l,c)}(t) \middle| \left\{ M_{(j',d')}(\bar{h}) \right\}_{(j',d') \in \mathcal{E}_{(l,c)}, \bar{h}=n+1, \dots, t-1} \right] \\ = \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \Pr\{S_l^c\} \geq 1 - \frac{\log M + 1}{M} - \mu. \end{aligned} \quad (\text{A.6})$$

Suppose a random variable defined as

$$\rho_t = \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{M_{(l,c)}(t)}{\kappa} \in \left\{ 0, \frac{1}{\kappa}, \frac{2}{\kappa}, \dots, 1 \right\}, \quad (\text{A.7})$$

where κ is the maximum number of queues that can be scheduled simultaneously in the interference set of any queue in the network. For any $\phi \geq 0$, based on (A.6) and the convexity of the exponential function $f(x) = e^{-\phi x}$ [21], it is not hard to obtain that

$$E[e^{-\phi \rho_t} | \rho_{n+1}, \dots, \rho_{t-1}] \leq 1 - \frac{1}{\kappa} \left(1 - \frac{\log M + 1}{M} - \mu \right) (1 - e^{-\phi}). \quad (\text{A.8})$$

Define $Y = \sum_{t=n+1}^{n+H} \rho_t$. Now, we aim to obtain an upper bound of $E[e^{-\phi Y}]$. It is well-known that the Chernoff bound for $E[e^{-\phi Y}]$ can be obtained if the variables ρ_t 's were independent across time. Obviously, according to the above-mentioned definition of ρ_t as well as the Markov characteristics of the random process $\{M_{(l,c)}(t)\}$, ρ_t 's are not independent across time. In order to upper bound $E[e^{-\phi Y}]$, we introduce a new random variable $Y' = \sum_{t=n+1}^{n+H} \chi_t$, where χ_t 's are independent and take values of 0 or 1 and with probability

$$\Pr\{\chi_t = 1\} = \frac{1}{\kappa} \left(1 - \frac{\log M + 1}{M} - \mu \right). \quad (\text{A.9})$$

Obviously, $E[e^{-\phi Y'}]$ can be upper bounded through Chernoff bound [22] as follows

$$E[e^{-\phi Y'}] \leq \exp \left[\frac{H}{\kappa} \left(\frac{\log M + 1}{M} + \mu - 1 \right) (1 - e^{-\phi}) \right]. \quad (\text{A.10})$$

Furthermore, based on (A.8), one can obtain $E[e^{-\phi Y}] \leq E[e^{-\phi Y'}]$ by induction [23]. Hence,

$$E[e^{-\phi Y}] \leq \exp \left[\frac{H}{\kappa} \left(\frac{\log M + 1}{M} + \mu - 1 \right) (1 - e^{-\phi}) \right]. \quad (\text{A.11})$$

Thus, from Markov inequality and (A.11), we get

$$\begin{aligned} \Pr \left\{ \kappa Y \leq H \left(1 - \frac{\log M + 1}{M} - 2\mu \right) \right\} \\ \leq \exp \left\{ \frac{H}{\kappa} \left[\phi + \left(\frac{\log M + 1}{M} + \mu - 1 \right) (1 - e^{-\phi}) \right. \right. \\ \left. \left. - \phi \left(\frac{\log M + 1}{M} + 2\mu \right) \right] \right\}. \end{aligned} \quad (\text{A.12})$$

For some $\phi \geq 0$, it is not hard to find a constant $\omega_1 > 0$ such that

$$\Pr \left\{ \kappa Y \leq H \left(1 - \frac{\log M + 1}{M} - 2\mu \right) \right\} \leq e^{-H\omega_1}. \quad (\text{A.13})$$

For the arriving process, define

$$Z = \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \sum_{t=n+1}^{n+H} \frac{A_l^c(t)}{r_l^c}. \quad (\text{A.14})$$

Considering the condition that the offered load meets in Lemma 2, one can easily infer for some $\omega_2 > 0$ that

$$\Pr \left\{ Z \geq H \left(1 - \frac{\log M + 1}{M} - 3\mu \right) \right\} \leq e^{-H\omega_2}. \quad (\text{A.15})$$

Hence, based on (A.13) and (A.15), we have

$$\begin{aligned} \Pr \left\{ \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n+H)}{r_l^c} \leq V(n) - H\mu \right\} \\ = \Pr \left\{ \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} + Z - \kappa Y \leq V(n) - H\mu \right\} \\ = \Pr \left\{ Z \leq V(n) - \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n)}{r_l^c} + \kappa Y - H\mu \right\} \\ \geq \Pr \{ Z \leq \kappa Y - H\mu \} \geq 1 - e^{-H\omega_1} - e^{-H\omega_2}. \end{aligned} \quad (\text{A.16})$$

Thus, by setting the value of H appropriately such that $e^{-H\omega_1} + e^{-H\omega_2} \leq \varsigma$, then we can obtain

$$\Pr \left\{ \sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{q_l^c(n+H)}{r_l^c} \leq V(n) - H\mu \right\} \geq 1 - \varsigma. \quad (\text{A.17})$$

This ends the proof of Lemma 2.

A.2. Proof of Proposition 2

Referring to the proof process of Proposition 1 as well as the corresponding definition of extended maximum capacity region for multihop flows with multiple paths in (20),

the efficiency ratio in (23) implies there is $\bar{P}_{si} \geq 0$ for all user s and path i satisfying

$$\sum_{(l,c) \in \mathcal{E}_{(j,d)}} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^l \lambda_s \bar{P}_{si}}{\sum_{b \in C} r_l^b} < \frac{1-\mu}{1+\mu} \left(1 - \frac{\log M + 1}{M}\right) \quad \text{s.t.} \quad \sum_{i=1}^{I(s)} \bar{P}_{si} = 1. \quad (\text{A.18})$$

For multihop multipath case, the following Lyapunov function is used,

$$U(n) = \frac{1}{\theta + 1} \sum_{l \in L} \sum_{c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^{1+\theta}. \quad (\text{A.19})$$

From (4) and (19) and the fact that $E[A_s(n)P_{si}(n)] = \lambda_s P_{si}(n)$, we thus have

$$\begin{aligned} E[U(n+1) - U(n)|q(n)] &\leq \sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta \\ &\cdot \left[\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{A_j^d(n)}{r_j^d} - \sum_{(j,d) \in \mathcal{E}_{(l,c)}} \Pr\{S_j^d\} \right] \\ &+ \sum_{l \in L, c \in C} o \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta, \end{aligned} \quad (\text{A.20})$$

where $o(\cdot)$ means higher order. Further on the basis of Lemma 1, it can be easy to obtain that

$$\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \Pr\{S_j^d\} \geq \frac{\sum_{(j,d) \in \mathcal{E}_{(l,c)}} q_j^d(n)/r_j^d}{V(n)} \left(1 - \frac{\log M + 1}{M}\right), \quad (\text{A.21})$$

for all queues (l, c) , where $V(n)$ has already been defined before in Section 4. In addition, by the binomial expansion [6], it is not difficult to obtain that for any $\mu > 0$, there will exist some $\theta_0 > 0$ such that

$$\frac{\sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} q_j^d(n)/r_j^d \right)^{(1+\theta)}}{V(n)} \geq (1-\mu) \sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta \quad (\text{A.22})$$

holds for all $\theta > \theta_0$. Hence, one can obtain

$$\begin{aligned} &\sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta \left[\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \Pr\{S_j^d\} \right] \\ &\geq \frac{(1 - ((\log M + 1)/M))}{V(n)} \sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^{\theta+1} \\ &\geq (1-\mu) \left(1 - \frac{\log M + 1}{M}\right) \sum_{l \in L, c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta. \end{aligned} \quad (\text{A.23})$$

Utilizing (A.18) and the routing selection mechanism in (21), it is not hard to find that

$$\begin{aligned} &\sum_{i=1}^{I(s)} P_{si}(n) \sum_{l \in L} \sum_{c \in C} \frac{H_{si}^l}{\sum_{b \in C} r_l^b} \sum_{(j,d) \in \mathcal{E}_{(l,c)}} \left(\sum_{(k,c') \in \mathcal{E}_{(j,d)}} \frac{q_k^{c'}(n)}{r_k^{c'}} \right)^\theta \\ &- (1+\mu) \sum_{i=1}^{I(s)} \bar{P}_{si} \sum_{l \in L} \sum_{c \in C} \frac{H_{si}^l}{\sum_{b \in C} r_l^b} \sum_{(j,d) \in \mathcal{E}_{(l,c)}} \left(\sum_{(k,c') \in \mathcal{E}_{(j,d)}} \frac{q_k^{c'}(n)}{r_k^{c'}} \right)^\theta \\ &\leq -\frac{\varphi_s}{2} \sum_{i=1}^{I(s)} [P_{si}(n)]^2 + \frac{\varphi_s}{2} \sum_{i=1}^{I(s)} (\bar{P}_{si})^2 \leq \varphi_s I(s). \end{aligned} \quad (\text{A.24})$$

Therefore, the Lyapunov drift is negative and is given by

$$\begin{aligned} E[U(n+1) - U(n)|q(n)] &\leq -\mu \sum_{l \in L} \sum_{c \in C} \left(\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right)^\theta \\ &\cdot \sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{\sum_{s=1}^S \sum_{i=1}^{I(s)} H_{si}^j \lambda_s \bar{P}_{si}}{\sum_{b \in C} r_l^b} + \sum_{s=1}^S \lambda_s \varphi_s I(s) \\ &+ \sum_{l \in L} \sum_{c \in C} o \left[\sum_{(j,d) \in \mathcal{E}_{(l,c)}} \frac{q_j^d(n)}{r_j^d} \right]^\theta. \end{aligned} \quad (\text{A.25})$$

The stability of the system then follows [24].

Data Availability

The data used to support the findings of this study are available from the corresponding authors on reasonable request.

Conflicts of Interest

The authors declare that Xinyan Xu and Fan Zhang are the joint first author. The authors declare that Department of Computer and Software Engineering, Shandong College of Electronic Technology; School of Information Engineering, Shandong Management University are the joint first

institution and they jointly enjoy the copyrights of this paper. The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work was jointly supported by the National Natural Science Foundation of China under Grant 61501283, the Shandong Provincial Natural Science Foundation of China under Grant ZR2017BF013, and the University of Jinan Science and Technology Program under Grant XKY1813.

References

- [1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.
- [2] L. Tassiulas, "Scheduling and performance limits of networks with constantly changing topology," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 1067–1073, 1997.
- [3] X. Wu, R. Srikant, and J. R. Perkins, "Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 595–605, 2007.
- [4] P. Chaporkar, K. Kar, X. Luo, and S. Sarkar, "Throughput and fairness guarantees through maximal scheduling in wireless networks," *IEEE Transactions on Information Theory*, vol. 54, no. 2, pp. 572–594, 2008.
- [5] A. Israeli and A. Itai, "A fast and simple randomized parallel algorithm for maximal matching," *Information Processing Letters*, vol. 22, no. 2, pp. 77–80, 1986.
- [6] A. Gupta, X. Lin, and R. Srikant, "Low-complexity distributed scheduling algorithms for wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1846–1859, 2009.
- [7] X. Lin and S. B. Rasool, "Constant-time distributed scheduling policies for ad hoc wireless networks," *IEEE Transactions on Automatic Control*, vol. 54, no. 2, pp. 231–242, 2009.
- [8] P. Kyasanur, J. So, C. Chereddi, and N. H. Vaidya, "Multichannel mesh networks: challenges and protocols," *IEEE Wireless Communications*, vol. 13, no. 2, pp. 30–36, 2006.
- [9] X. Lin and S. B. Rasool, "Distributed and provably efficient algorithms for joint channel-assignment, scheduling, and routing in multichannel ad hoc wireless networks," *IEEE/ACM Transactions on Networking*, vol. 17, no. 6, pp. 1874–1887, 2009.
- [10] V. Bhandari and N. H. Vaidya, "Scheduling in multichannel wireless networks," in *Distributed Computing and Networking. ICDCN 2010. Lecture Notes in Computer Science*, vol. 5935, K. Kant, S. V. Pemmaraju, K. M. Sivalingam, and J. Wu, Eds., pp. 6–17, Springer, Berlin, Heidelberg, 2010.
- [11] Y. Cheng, H. Li, D. M. Shila, and X. Cao, "A systematic study of maximal scheduling algorithms in multiradio multichannel wireless networks," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1342–1355, 2015.
- [12] K. Jung and D. Shah, *A distributed matching algorithm*, Tech. Rep., MIT Press, Cambridge, MA, USA, 2006.
- [13] F. Juraschek, M. Günes, M. Philipp, B. Blywis, and O. Hahm, "DES-Chan: a framework for distributed channel assignment in wireless mesh networks," in *2011 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pp. 1–6, Melbourne, VIC, Australia, November 2011.
- [14] M. O. Kabaou, B. C. Rhaimi, M. N. Abdelkrim, and M. Marzoug, "Radio resource allocation for multimedia traffic over wireless channels in OFDMA downlink systems," in *2010 2nd International Conference on Advanced Computer Control*, pp. 27–29, Shenyang, China, March 2010.
- [15] Z. Ren, S. Chen, B. Hu, and W. Ma, "Energy-efficient resource allocation in downlink OFDM wireless systems with proportional rate constraints," *IEEE Transactions on Vehicular Technology*, vol. 63, no. 5, pp. 2139–2150, 2014.
- [16] Y. Li, S. Xia, X. Xiong, S. Liu, X. Liu, and M. Daneshmand, "Throughput-delay trade-off scheduling in multi-channel downlink wireless networks," *Wireless Personal Communications*, vol. 92, no. 2, pp. 681–694, 2017.
- [17] J. G. Choi and G. S. Choi, "Optimal scheduler with simplified queue structure for multi-channel wireless networks," *IEEE Communications Letters*, vol. 15, no. 9, pp. 962–964, 2011.
- [18] M. Li, S. Salinas, P. Li, X. Huang, Y. Fang, and S. Glisic, "Optimal scheduling for multi-radio multi-channel multi-hop cognitive cellular networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 139–154, 2015.
- [19] E. Leonardi, M. Mellia, F. Neri, and M. A. Marsan, "On the stability of input-queued switches with speed-up," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 104–118, 2001.
- [20] X. Lin and N. B. Shroff, "Utility maximization for communication networks with multipath routing," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 766–781, 2006.
- [21] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.
- [22] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, New York, NY, USA, 1995.
- [23] F. Zhang, Y. Cao, and D. Wang, "A note on 'low-complexity distributed scheduling algorithms for wireless networks'," *IEEE/ACM Transactions on Networking*, vol. 23, no. 4, pp. 1367–1369, 2015.
- [24] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

