12-2015

A Time-Efficient Strategy For Relay Selection and Link Scheduling In Wireless Communication Networks

Chenxi Qiu
Clemson University, chenxiq@g.clemson.edu

A Time-Efficient Strategy For Relay Selection and Link Scheduling In Wireless Communication Networks

---

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Computer Engineering

---

by
Chenxi Qiu
December 2015

Accepted by:
Dr. Haiying Shen, Committee Chair
Dr. Harlan B. Russell
Dr. Kuang-Ching (KC) Wang
Dr. Jason O. Hallstrom

Abstract

Despite the unprecedented success and proliferation of wireless communication, sustainable reliability and stability among wireless users are still considered important issues in the underlying link protocols. Existing link-layer protocols, like ARQ [44] or HARQ [57,67] approaches are designed to achieve this goal by discarding a corrupted packet at the receiver and performing one or more retransmissions until the packet is successfully decoded or a maximum number of retransmission attempts is reached. These strategies suffer from degradation of throughput and overall system instability since packets need to be en/decode in every hop, leading to high burden for relay nodes especially when the traffic load is high. On the other hand, due to the broadcast nature of wireless communication, when a relay transmits a packet to a specific receiver, it could become interference to other receivers. Thus, rather than activating all the relays simultaneously, we can only schedule a subset of relays in each time slot such that the interference among the links will not cause some transmissions to fail. Accordingly, in this dissertation, we mainly address the following two problems:

1) *Relay selection*: given a route (i.e., a sequence of relays), how to select the relays to en/decode packets to minimize the latency to reach the destination?

2) *Link scheduling*: how to schedule relays such that the interference among the relays will not cause transmission failure and the throughput is maximized?

Relay Selection Problem. To solve the relay selection problem, we propose a Code Embedded Distributed Adaptive and Reliable (CEDAR) link-layer framework that targets low latency. CEDAR is the first theoretical framework for selecting en/decoding relays to minimize packet latency in wireless communication networks. It employs a theoretically-sound framework for embedding channel codes in each packet and performs the error correcting process in selected intermediate nodes in packet's route. To identify the intermediate relay nodes for en/decoding to minimize average packet

latency, we mathematically analyze the average packet delay, using Finite State Markovian Channel model and priority queuing model, and then formalize the problem as a non-linear integer programming problem. To solve this problem, we design a scalable and distributed scheme which has very low complexity. The experimental results demonstrate that CEDAR is superior to the schemes using hop-by-hop decoding and destination-decoding in terms of both packet delay and throughput. In addition, the simulation results show that CEDAR can achieve the optimal performance in most cases.

Link Scheduling Problem. As for the link scheduling problem, we formulate a new problem called *Fading-Resistant Link Scheduling (*Fadin-R-LS*) problem*, which aims to maximize the throughput (the sum data rate) for all the links in a single time slot. The problem is different from the existing link scheduling problems by incorporating the Rayleigh-fading model to describe the interference. This model extends the deterministic interference model based on the Signal-to-Interference Ratio (SIR) using stochastic propagation to address fading effects in wireless networks. Based on the geometric structure of *Fadin-R-LS*, we then propose three centralized schemes for *Fadin-R-LS*, with $O(g(L))$, $O(g(L))$, and $O(1)$ performance guarantee for packet latency, where $g(L)$ is the number of length magnitudes of link set L. Furthermore, we propose a completely distributed approach based on game theory, which has $O(g(L)^{2\alpha})$ performance guarantee.

Furthermore, we incorporate a *cooperative communication (CC)* technique, e.g., maximum ratio combining (MRC), into our system to further improve the throughput, in which receivers are allowed to combine messages from different senders to combat transmission errors. In particular, we formulate two problems named *cooperative link scheduling problem (CLS)* and *one-shot cooperative link scheduling problem (OCLS)*. The first problem aims to find a schedule of links that uses the minimum number of time slots to inform all the receivers. The second problem aims to find a set of links that can inform the maximum number of receivers in one time slot. We prove both problems to be NP-hard. As a solution, we propose an algorithm for both CLS and OCLS with $g(\mathcal{K})$ approximation ratio, where $g(\mathcal{K})$ is so called the *diversity of key links*. In addition, we propose a greedy algorithm with $O(1)$ approximation ratio for OCLS when the number of links for each receiver is upper bounded by a constant.

In addition, we consider a special case for the link scheduling problem, where there is a group of vehicles forming a platoon and each vehicle in the platoon needs to communicate with the leader vehicle to get the leader vehicle's velocity and location. By leveraging a typical feature of a

platoon, we devise a link scheduling algorithm, called the Fast and Lightweight Autonomous link scheduling algorithm (FLA), in which each vehicle determines its own time slot simply based on its distance to the leader vehicle.

Finally, we conduct a simulation on Matlab to evaluate the performance of our proposed methods. The experimental results demonstrate the superior performance of our link scheduling methods over the previous methods.

Acknowledgments

First, I would like to express my sincere gratitude to my advisor Dr. Haiying Shen for the continuous support of my Ph.D study and research, for her patience, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the time of research and writing of this thesis. Her passion for scientific research and hard working will always affect me in the rest of my academia life.

Besides my advisor, I would like to thank the rest of my thesis committee: Dr. Harlan B. Russell, Dr. Kuang-Ching Wang, and Dr. Jason O. Hallstrom, for their encouragement, insightful comments, and hard questions.

I thank my fellow labmates in Pervasive Communications Laboratory: Ze Li, Kang Chen, Guoxin Liu, Yuhua Lin, Bo Wu, Liuhua Chen, Zhuozhao Li, Jinwei Liu, Li Yan, Ankur Sarker, Rand Ali, Haoyu Wang, Xiang Zhang, Shenghua He, Pengfei Zou, Heng Zhou, for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in these years.

Last but not the least, I would like to thank my wife, Jie Wang, and my parents, Junxing Qiu and Qinxia Jiang, for supporting me spiritually throughout my life.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Despite the unprecedented success and proliferation of wireless communication, there are major shortcomings in the underlying link-layer protocols in providing sustainable reliability and stability among wireless users. Popular wireless link-layer protocols, such as the retransmission ARQ and Forward Error Correction (FEC) based ARQ (HARQ) approaches (employed by the IEEE 802.xx and LTE standard suite) have been designed to achieve some level of reliability by discarding a corrupted packet at the receiver and performing one or more retransmissions until the packet is decoded/received error-free or a maximum number of retransmission attempts is reached. This methodology suffers from degradation of throughput and overall system instability since decoding failures at the receiver due to a small number of bit errors lead to packet drops and discarding a large number of correctly delivered data bits.

Besides selecting the intermediate nodes to provide highly sustainable reliability and stability for the system, we also need to consider avoiding interference among transmissions. Due to the broadcast nature of wireless communication, when a sender transmits a packet to a specific receiver, it could become interference to other receivers. Thus, when scheduling links, we need to consider how to select links such that the interference among the links will not fail transmissions. When we study the scheduling problem in wireless networks, the choice of the interference model is of fundamental significance.

1.1 Problem Statement

Many leading research efforts [11, 16, 22, 28, 29, 36, 37, 43, 56, 57, 64, 67] have highlighted the inefficiencies of these link-layer protocols and proposed a variety of remedy solutions. The majority of these efforts either consider variations of the ARQ, HARQ or a hybrid approach of both schemes [11,28,37,44,57,67]. They largely follow the traditional "store-and-forward" link-layer design paradigm: each data packet must be fully received and corrected by every relay node before it is forwarded. This design paradigm increases stability but still cannot provide high stability due to its hop-by-hop operation.

It is our belief that achieving the ultimate objective of the development of ubiquitous and heterogeneous wireless networks demands fundamental and radical changes to the conventional link-layer protocol design. Thus, we study and develop alternative optimal and low-complexity error recovery strategies in link-layer design to achieve high reliability and stability by partially and optimally selecting relay nodes. The objectives of the strategies are to ensure: (1) Low end-to-end latency and rapid delivery of packets; (2) High throughput with minimum data loss. To meet these objectives, we develop solutions that address the following key issues:

(1) *Minimizing propagation and transmission (prop&tran) delay*: at which intermediate nodes (if any) a link-layer packet should be detected to minimize packet delay?

(2) *Minimizing queuing delay*: as multiple relay nodes in a route perform error recovery on the same packet stream and one node may perform error recovery for multiple packet streams, how to select relay nodes that provides global reliability and stability in a wireless network with many source-destination packet streams?

Note that our work shares the same objectives as some previous works on en/decoding schemes and network coding (e.g., PPR [28] and MIXIT [37]). However, unlike these previous works that focus on route determination or en/decoding scheme design, our work aims to determine the intermediate nodes to en/decode packets given a route and an en/decoding scheme. Our work can be employed in those en/decoding schemes and network coding schemes for further performance enhancement.

Besides relay selection, we also need to consider how to schedule links to avoid interference among relay nodes. Although the approximation algorithms for scheduling problem have been widely studied based on different interference models [4, 10, 15, 18, 20, 21, 25, 30, 33, 35, 40, 45, 53, 61, 62, 65],

none of the these works take into account the fluctuating fading effect, or cooperative communication (CC) in transmissions. Also, no previous work considers applying link scheduling methods to some special scenarios, like the vehicle platoon network. Accordingly, we consider the problem in the following cases:

(1) *Rayleigh fading SIR interference model is applied*: One of the most commonly used interference models in the traditional scheduling problem is the graph based model [30–33,35,40,45,52,53]. It only considers the interference on a receiver from other senders within the transmission range. However, although the interference from a single far-away sender can be relatively small, the accumulated interference from several such senders can be sufficiently high to corrupt a transmission. Hence, the scheduling problem solutions based on the graph based model cannot be guaranteed to work in many real scenarios. Another interference model, named the physical interference model or the Signal-to-Interference Ratio (SIR) model, offers a more realistic representation of wireless networks [4, 15, 18, 20, 21, 65]. In this model, a message is received successfully iff the SIR is no smaller than a hardware-defined threshold. This definition of a successful transmission, as opposed to the graph based definition, accounts for interference generated by senders located far away.

However, the SIR model still uses a limited view of signal propagation. Its main assumption is that any signal transmitted at power level P is always received at distance d with strength $Pd^{-\alpha}$, where α is path loss exponent. The real signal propagation is not deterministic, e.g., the links may become susceptible to fading fluctuations in signal strength due to mobility in a multi-path propagation environment [42]. Therefore, some advanced models using stochastic approaches to consider fading effects have been proposed [3, 24]. Most prominently, in the Rayleigh-fading model, the signal strength is modeled by an exponentially distributed random variable with mean $Pd^{-\alpha}$ [13, 42]. This however also makes the SIR non-deterministic, and hence causes the judgment of successful transmission in analyzing the link scheduling problem much more complicated. As a result, finding solutions for the link scheduling problem with the Rayleigh-fading model is a non-trivial task.

(2) *The CC technique (e.g., MRC) is allowed*: It has been shown that MRC technique has a great potential to increase the capacity of wireless networks [1, 48, 60]. In wireless networks, before a message reaches the destination (receiver), it may have several copies stored by other

nodes. For example, the sender's neighboring nodes can store the unintended message from the sender due to the broadcast nature of wireless transmission; also, in multi-hop transmission, relay nodes can store the copies of the original message. In CC, the nodes storing the copies (including the original message) are allowed to send the copies to the receiver together, and the receiver combines the signal power of the received copies in an additive fashion using MRC to recover the message. Similar to Fading-R-LS, we formulated the link scheduling problem based on the SIR model, namely *cooperative link scheduling problem (CLS)* and *one-shot cooperative link scheduling problem (OCLS)*.

(3) *Vehicles in a platoon*: Finally, we apply the link scheduling method to the vehicle platoon system, which is considered as a type of next-generation of land transportation systems. In a platoon, one leader vehicle and several follower vehicles drive in a single lane, where each vehicle maintains a shorter distance from its preceding vehicle, which requires to build a well-connected communication network for a platoon so that vehicles can quickly adjust their velocities through fast communication. Considering vehicles in platoon may change their locations [46], directly employing the previous link scheduling algorithm to the platoon network would lead to much higher communication cost and longer transmission delay. Considering the poor channel capacity for the vehicle to vehicle (V2V) communication [7], a challenge is *how to conduct link scheduling with low delay and low communication cost in a decentralized platoon network?* In this dissertation, we aim to resolve the link scheduling problem arisen in the platoon network.

1.2 Research Approach

1.2.1 A Low-Latency and Distributed Relay Selection Strategy for Packet Recovery in Wireless Networks

The design of the strategies requires network-of-queues models that capture the error correction process and networking effects of traffic flows over multi-hop wireless paths. Accordingly, we develop mathematical models for the prop&tran delay and queuing delay for a packet based on the path length between two consecutive decoding nodes in a route (route segment length). Through rigorous mathematical analysis on the models, we derive two propositions that (1) can identify the

intermediate nodes for decoding which minimize prop&tran delay of a packet, and prove that (2) balanced en/decoding load distribution among decoding nodes in the network minimizes the queuing delay. Based on the propositions, we formulate the problem of minimizing delay as a non-linear integer programming problem. However, due to the NP-hard nature of the problem and impracticability of collecting all required information to find the global optimal solution, we propose a sub-optimal Code Embedded Distributed Adaptive and Reliable (CEDAR) link-layer framework for wireless networks. CEDAR is a distributed and cooperative error recovery design, which represents a new paradigm in both transmission and distributed recovery processing and promises significant increase of capacity and throughput gain in wireless networks. CEDAR provides an adaptive environment for various error recovery strategies with respect to reliability, stability and energy consumption constraints. We believe that CEDAR is the first comprehensive theoretical framework for studying the selection of en/decoding relay nodes to increase networks' reliability and stability.

1.2.2 Link Scheduling

1.2.2.1 Fading Resistant Link Scheduling

To address the link scheduling problem in fading environment, we formulate a link scheduling problem called *Fading-Resistant Link Scheduling problem* (Fading-R-LS), in which the interferences among links are modeled by the Rayleigh-fading channel model. Given a set of links L, *Fading-R-LS* is to determine which subset of L should be activated such that the total throughput is maximized in one time slot. We first prove that this problem is NP-hard, and then propose three solutions and analyze their performance guarantees:

1) **Link diversity partition algorithm (LDP)**. According to the geometric structure of *Fading-R-LS*, LDP builds several link classes based on link lengths and schedule the links in each class separately. We prove that LDP has the performance guarantee of $O(g(L))$, where $g(L)$ is the number of length magnitudes of link set L.

2) **Recursive link elimination algorithm (RLE)**. RLE is proposed in the case that data rate of each link is the same. RLE iteratively picks up the unpicked link with the shortest link length and eliminates other links that interfere with the picked link. We prove RLE has a constant performance guarantee.

3) **Decentralized link scheduling algorithm (DLS)**. In DLS, each sender makes its own

decision based on local information. We analyze DLS as a game where the senders are the players and prove that each achieved Nash equilibrium in this game results in an expected throughput that is close to optimal.

1.2.2.2 Cooperative Communication Link Scheduling

To solve cooperative link scheduling problem and one-shot cooperative link scheduling problem, we propose two *link length diversity (LLD)* based algorithms LLD-CLS and LLD-OCLS to solve CLS and OCLS, respectively. The basic idea of these two algorithms is to partition all the links into several classes based on their length (*i.e.*, distance between the link's sender and receiver) and schedule the links in each class separately. We prove that both LLD-CLS and LLD-OCLS have $g(\mathcal{K})$ approximation ratio, where $g(\mathcal{K})$ denotes the diversity of key links (Definition 5.2.2 and Definition 5.1.1). In addition, we consider a special case of the OCLS problem, in which the number of links for each receiver is upper bounded by a constant, and propose a simple greedy algorithm for it: in each iteration, the algorithm greedily picks up the "strongest" unpicked links and excludes any link that conflicts with the links we have selected. We prove that this greedy algorithm has $O(1)$ approximation ratio.

1.2.2.3 The Fast and Lightweight Autonomous Link Scheduling Algorithm (FLA)

In this part, we aim to resolve the link scheduling problem arisen in the vehicle platoon network. More specifically, we propose a Fast and Lightweight Autonomous link scheduling algorithm (FLA) that takes advantage of a typical feature of the platoon. Different from general wireless networks, where the nodes are arbitrarily distributed, in the platoon, vehicles drive in a single lane and the distance between neighboring vehicles is equal to the safety distance [66]. Based on this feature, to avoid interference, we let vehicles use the same time slot only when their distance is beyond the interference range and let vehicles within the interference range use different time slots. *Interference range* is the distance range that makes the interference upper bounded by an acceptable value for packet decoding. Specifically, we geometrically partition the platoon into segments of the same length δ such that each segment contains at most one vehicle. Then, we consider every g consecutive segments as a group, and allocate g different time slots to the segments in each group. g is the minimum number of time slots needed to avoid the interference. The aforementioned platoon feature also enables a vehicle to locate its segment position in a group and then autonomously decide

its time slot accordingly. As a result, the vehicles using the same time slot have a distance equals to the interference range in between.

1.3 Contributions

The contributions of the dissertation include

- For relay selection problem:

  1) ***Channel model for code embedded packet transmission in multiple hops***. We first derive the closed form of the probability of decoding failure of a code embedded packet traveling through a given number of hops based on the Finite State Markovian Channel (FSMC) model;

  2) ***Closed form of packet delay***. Combining the derived packet failure rate with the propagation delay model, transmission delay model, and priority queuing model, we then derive the closed form of the prop&tran delay and queuing delay for each packet.

  3) ***Formal formulation of the relay selection problem***. According to the derived packet delay, we formalize the problem of choosing the intermediate en/decoding nodes for minimum delay and minimum difference of en/decoding load of all the nodes as a non-linear integer programming, problem which is an NP-hard problem. As far as we know, this is the first theoretical work for choosing en/decoding nodes that targets on the reliability and stability of wireless networks.

  4) ***Time-efficient distributed methods***. Due to the hardness of the relay selection problem, we propose two distributed sub-optimal strategies for CEDAR in heavy traffic environment and light traffic environment, respectively, to achieve higher reliability, stability and en/decoding load balancing compared with previous methods.

- For fading resistant link scheduling:

  1) ***The fading resistent link scheduling problem (Fading-R-LS)***. We formulate the *Fading-R-LS* problem that takes into account the fading effect, which is not considered in previous scheduling problems. In addition, we first give an integer linear programming (ILP) formulation of *Fading-R-LS* and prove it is NP-hard;

  2) ***The link diversity partition algorithm (LDP)***. According to the geometric structure of *Fading-R-LS*, we propose the LDP centralized method. It builds a number of link classes

based on link lengths and schedule the links in each class separately. We prove that LDP has the performance guarantee of $O(g(L))$, where $g(L)$ is the number of length magnitudes of link set L. To the best of our knowledge, no previous works propose an approximation algorithm for the link scheduling problem in fading environment;

3) ***The recursive link elimination algorithm (RLE)***. We then consider a special case of *Fading-R-LS*, in which the data rate of each link is the same, and propose the RLE algorithm accordingly. RLE iteratively picks up the unpicked link with the shortest link length and eliminates other links that interfere with the picked link. We prove RLE has the performance guarantee of $O(\Delta^{\alpha})$ for throughput, which has a better performance guarantee than LDP (Δ is the ratio between the maximum and the minimum distances between nodes). In addition, the experimental results demonstrate that our fading-resistant link scheduling algorithms (LDP and RLE) outperform the previous methods in term of the successful transmission probability [18, 20];

4) ***The decentralized link scheduling algorithm (DLS) based on game theory***. We propose DLS which allows each sender to make its own decision based on local information. We analyze DLS as a game where the senders are the players and prove that each achieved Nash equilibrium in this game results in an expected throughput that is close to optimal;

- In addition, in the case of cooperative communication link scheduling:

  1) ***The cooperative link scheduling problem (CLS) and the one-shot cooperative link scheduling problem (OCLS)***. We formulate two problems: CLS and OCLS. The objective of CLS is to inform all the receivers using as few time-slots as possible. The objective of OCLS is to maximize the number of receivers informed concurrently in one time slot. We also prove both CLS and OCLS to be NP-hard. As far as we know, this paper is the first work studying the link scheduling problem in cooperative communication networks;

  2) ***The link length diversity (LLD) based algorithm for CLS and OCLS (LLD-CLS and LLD-OCLS)***. We propose algorithms LLD-CLS and LLD-OCLS for CLS and OCLS, respectively, where both algorithms have $g(\mathcal{K})$ approximation ratios. Furthermore, we propose an algorithm with $O(1)$ approximation guarantee for OCLS when the number of senders in each request is upper bounded by a constant. The experimental results indicate that our cooperative link scheduling algorithms outperform the non-cooperative algorithms [18, 20];

- Finally, we applied our link scheduling algorithm to the application of vehicle platoon:

  1) ***The vehicle link scheduling problem (VLS).*** We formally formulate the VLS and prove the VLS remains NP-hard;

  2) ***The fast and lightweight autonomous link scheduling algorithm (FLA)***. By leveraging a typical feature of a platoon, i.e., there exists a safety distance between consecutive vehicles in single lane, we devise a link scheduling algorithm, called the Fast and Lightweight Autonomous link scheduling algorithm (FLA), in which each vehicle determines its own time slots simply based on its distance to the leader vehicle. The experimental results demonstrate the superiority of FLA over the previous methods in terms of packet latency and delivered ratio.

1.4 Dissertation Organization

The remainder of this dissertation is arranged as followings. Chapter 2 introduces the related works. Chapter 3 introduces the system model used through this paper and problems formulated based on this model. Chapter 4 and Chapter 5 introduce the detailed designs of relay selection and link scheduling in this dissertation. Chapter 6 presents the performance evaluation of these methods. Finally, Section 7 concludes this dissertation with remarks on future work.

Chapter 2

Related Work

2.1 Relay Selection

Error detection and correction is one the richest problems in communication literature. The link-layer protocol of the current TCP/IP stack has adopted variations of error recovery mechanisms to provide reliability for point-to-point communication especially for wireless systems. Different wireless communication standards currently utilize variations of error control protocols that generally can be categorized into ARQ [44] and HARQ-based [57,67] protocols. For instance IEEE802.11 WiFi uses ARQ where a receiving node discards corrupted packets (even when there is only a single bit error) and requests for a retransmission. The 4G/LTE deploys HARQ with Turbo Codes where the sender node encodes the packet payload using Turbo channel codes [14] prior to the transmission. Accordingly, the receiver node requests for a retransmission when the decoding of the received packet fails. In conjunction with the current wireless link-layer standards, there is significant work and research conducted to improve the performance of either ARQ- or HARQ-based protocols. Several kinds of HARQ protocols (see [57,67] and the reference therein) improve the throughput of the ARQ schemes by packet combining, e.g. by keeping the erroneous received packets and utilizing them for detection and packet recovery. Examples of recent efforts for combating the inefficiency of ARQ-based wireless protocols include Partial Packet Recovery (PPR) [28], SOFT [64], and Automatic Code Embedding (ACE) framework [55]. Some of these approaches, such as PPR and SOFT, exploit physical layer information regarding the quality of individual bits to increase the probability of recovering corrupted packets. Other schemes, such as ACE, utilize information available in the

current 802.11 link-layer protocols in conjunction with error correcting codes to recover corrupted packets. Ilyas *et al.* [26] proposed the "Poor Man's SIMO System" (PMSS) to reduce packet losses in networks of commodity IEEE 802.15.4 sensor motes using cooperative communication and diversity combination. Based on mathematical analysis, Jelenkovi *et al.* [29] proposed a new dynamic packet fragmentation algorithm that can adaptively match channel failure characteristics. Reuven *et al.* [11] considered a new scenario, in which when a base station wishes to multi-cast information to a large group of nodes using application-layer forward error correction (FEC) codes. It has been shown that network coding improves network reliability by reducing the number of packet retransmissions in lossy networks [16]. Thus, by coupling channel coding and network coding, Guo *et al.* [22] proposed a scheme named Non-Binary Joint Network-Channel Decoding (NB-JNCD) for reliable communication in wireless networks. These aforementioned works have significantly improved the ARQ- and HARQ-based link-layer performance and provide a comprehensive error control approach for wireless communication. However, virtually all of these efforts follow the conventional TCP/IP link-layer "store-and-forward" design paradigm, where each relay node verifies the correctness of each packet before forwarding it to the next node. This inherently introduces substantial overhead on bandwidth utilization and throughput and the overall end-to-end delay. In addition the point-to-point error recovery is not an optimal approach for energy constrained dense wireless networks. Though the previous work MIXIT [37] has jettisoned reliable link-layer error detection and recovery altogether using a symbol level network coding, its coding/decoding algorithms is more demanding for computational capacities of nodes than traditional store and forward methods. Comparing to MIXIT's implementation on software radios, CEDAR is more suitable for the devices with constrained processing capability, e.g., sensors, because CEDAR implements the decoding process by Reed Solomon, which can be encoded and decoded by hardware.

Accordingly, in this paper, we pursue a paradigm shift in the conventional link-layer design and propose a distributed, low-complexity, and adaptive scheme to achieve high reliability, stability and energy-efficiency in packet transmission. CEDAR is introducing a new chapter in link-layer design for future wireless networks comprising of energy constrained nodes where error recovery is optimally conducted in selected nodes.

2.2 Link Scheduling

Based on the choice of interference models, the previous works can be classified to two groups: *graph based scheduling* [21, 30–33, 35, 40, 45, 52, 53] and *SINR based scheduling* [4, 15, 18, 20, 65].

2.2.1 Graph based scheduling

Graph models have been served as the useful abstraction for studying scheduling problems for many years. For example, Sharma *et al.* [53] defined a k-hop interference model, in which no two links within k-hops can successfully transmit simultaneously, and proved that the scheduling problem is NP-hard when $k > 1$. Lin and Shroff [45] proposed *Greedy Maximal Scheduling* (GMS), which can be implemented in a distributed manner. Joo *et al.* [33] further provided numerous analytic results to characterize the performance limits of GMS and Jiang *et al.* [31] introduced a modified GMS scheduling algorithm based on CSMA random access. Wang *et al.* [61] studied the link scheduling problem for a multi-hop wireless network to maximize throughput. They assumed each node has different transmission range and interference range, and the methods they presented can achieve a constant factor of the optimum. Cheng *et al.* [10] studied the problem in multi-radio multi-channel wireless networks, and proved that the problem is NP-hard in this scenario, in both the k-hop interference model and unit disc model. Wang *et al.* [62] developed joint TCP congestion control and carrier sense multiple access (CSMA) scheduling schemes for Internet traffic over distributed multi-hop wireless links, in which the interference among the links is modeled by a conflict graph. Kar *et al.* [35] considered the question of obtaining tight delay guarantees for throughout-optimal link scheduling in arbitrary topology wireless ad-hoc networks. Jiang *et al.* [30] presented a distributed randomized scheme for scheduling and congestion control. Krifa and Barakat [40] investigated both the problems of scheduling and buffer management in delay tolerant networks. They proposed a centralized optimal scheme and a distributed scheme using statistical learning to approximate the required global knowledge. There are also some graph-based link scheduling scheme considering fading effect [32, 52]. Reddy *et al.* [52] analyzed the performance of GMS where the capacity of links changes over time. Joo *et al.* [32] considered the link scheduling problem in fading environment as a Maximum Weight Independent Set (MWIS) problem and proved it to be NP-hard.

Although these algorithms present extensive theoretical analysis, they are constrained to the limitations of the graph interference model that omits the accumulative nature of wireless signals.

Comparing to graph model, SINR model offers a more realistic representation of wireless networks. As proved by Gronkvist *et al.* [21] using both theoretical analysis and experiments, the graph based scheduling protocols are inefficient in the SINR model.

2.2.2 SINR based scheduling

There have been many works studying the problem of joint link scheduling and power control in the SINR model [39, 41, 50]. For example, Kozat *et al.* [39] addressed the joint problem to minimize the total transmit power subject to the end-to-end bandwidth guarantees and the bit error rate requirements of each transmission. The problem is proved NP-hard by constructing a reduction from integer programming. Leung and Wang [41] proved that the problem of maximizing throughput by adaptive modulation and power control while meeting packet error constraints is NP-hard. In [50], Pei and Kumar set the goal of the problem as maximizing capacity region of the network, *i.e.* the maximum attainable network throughput. They also proposed a low complexity distributed algorithm for this problem. In addition, Hong and Scaglione [21] showed that the graph based scheduling protocols are inefficient in SINR model using both theoretical analysis and experiments [15,21,25,65]. ElBatt *et al.* [15] introduced a joint scheduling and power control algorithm for multicast ad hoc networks based on the SINR model. Huang *et al.* [25] presented an optimization-based formulation for joint scheduling and resource allocation in the uplink OFDM access network and proposed heuristic solutions. Xu *et al.* [65] studied periodic scheduling for data aggregation with minimum delay under various interference models. They proposed a family of real-time query scheduling protocols and propose schedulability test schemes to test whether, for a set of queries, each query job can Some other works focused on designing algorithms with lower approximation guarantee [4, 18, 20]. Brar *et al.* [4] proposed a polynomial time algorithm and proved an approximation ratio for their method under uniform random node distribution. Goussevskaia *et al.* [18] formulated the scheduling problem in the geometric SINR model, proved its NP-hardness, and proposed a greedy solution with performance guarantee $O(g(L))$. Goussevskaia *et al.* also proposed a scheduling algorithm with constant approximation guarantee, which is independent of the network topology and size [20]. They further formulated a variation of the problem, in which analog network coding is allowed, and presented NP-hard proof of the problem [19]. Goussevskaia *et al.* [19] also formulated a variation of the problem, in which analog network coding is allowed, and presented NP-hard proof the problem [19]. Chafekar *et al.* [6] proposed an algorithm for the scheduling problem under SINR constraints with $O(g(D))$ per-

formance guarantee, where $O(g(D))$ is the ratio between the maximum and the minimum distances between nodes. Brar *et al.* [4] proposed a greedy scheduling algorithm with performance guarantee of $O(N^{1-\frac{2}{\psi(\alpha)+\epsilon}}(\log N)^{\frac{2}{\psi(\alpha)+\epsilon}})$ based on the assumption that nodes are distributed uniformly in a square of unit area, where $\psi(\alpha)$ is a constant depending on the path loss exponent α. The SINR model offers a more realistic representation of wireless networks than that of graph model, but However, the SINR model still uses a limited view of signal propagation since it does not consider the fading fluctuations in received signal strength (e.g., caused by the mobility in a multi-path propagation environment). Though Dams *et al.* [13] have studied the relationship between the non-fading SINR model and the Rayleigh-fading model for the scheduling problem, they did not discuss either the complexity of the problem considering fading or how to design an efficient algorithm based on the Rayleigh-fading model. Our work is the first that analyzes the hardness of the link scheduling problem under Rayleigh-fading model and proposes approximation algorithms for the problem.

Chapter 3

Problem Statement

In this chapter, we first introduce the system model we will use throughout this paper in Section 3.1. Based on the system model, we formally formulate both the relay selection problem (Section 3.2.1) and the link scheduling problem (Section 3.2.2), where the link scheduling problem includes the fading resist link scheduling problem (Section 3.2.2.1), the CC link scheduling problem (Section 3.2.2.2), and the vehicle scheduling problem (Section 3.2.2.3).

3.1 System Model

3.1.1 Relay Selection

Network model. First, we consider a wireless network comprised of N *nodes* denoted by $\mathcal{V} = \{v_1, ..., v_N\}$. Each traffic flow from a *source node* to a *destination node* transverses over a predetermined set of *links* (a route specified by the network layer). Let $\mathcal{R} = \{\mathbf{r}_1, ..., \mathbf{r}_K\}$ denote the set of transmission routes. Each route $\mathbf{r}_k$ ($\mathbf{r}_k \in \mathcal{R}$) carries a data stream following Poisson distribution with arrival rate λ_k. We use $\mathbf{r}_k = \{v_{k_1}, ..., v_{k_{n_k}}\}$ to represent the node sequence in $\mathbf{r}_k$ ($v_{k_1}, ..., v_{k_{n_k}} \in \mathcal{V}$), where n_k is the number of nodes in $\mathbf{r}_k$. We consider a network with heterogeneous types of traffic, i.e., a combination of real-time traffic with delay constraint and traffic with no delay constraint. We use U_k to denote the delay constraint of route r_k; $U_k = \infty$ if the packet in route r_k has no delay constraint. Finally, we use indicator variable $y_{i,k}$ to denote whether v_i is in $\mathbf{r}_k$. If yes, $y_{i,k} = 1$; otherwise $y_{i,k} = 0$.

Channel model. Finite State Markovian Channel model (FSMC) [59] is a channel model that uses finite state Markov chain to describe the process, under which errors are introduced into a transmitted packet over a wireless route. The model has a finite set of error states $\Psi = (\psi_1, \psi_2, ..., \psi_B)$ ($|\Psi| = B$), each corresponding to a Binary Symmetric Channel (BSC). The channel model can be considered as a combination of B number of various BSCs with unique BERs (ϵ) (i.e., $\epsilon_l \neq \epsilon_j$ for $l \neq j$, $l, j = 1, 2, ..., B$). Assuming packets are transmitted during discrete time slots τ_i ($i = 1, 2, 3...$) which can be referred as *transmission intervals*. During the i^{th} transmission interval, a packet is transmitted from a BSC to another BSC with cross-over BER ϵ_i. Each ϵ_i of a particular τ_i is valued from Ψ. The Markovian model assumes a homogenous and stationary Markov chain with transition probability matrix $\mathcal{T} = (t_{ij})_{B\times B}$ and initial probability $\pi = (\pi_1, ..., \pi_B)$. $\mathcal{T} = (t_{ij})_{B\times B}$ can be trained on real channel traces by using the statistics of previous transmission intervals. This captures the effects of multi-path fading and interferences on the channel BER in every transmission interval using a single aggregated model [59]. The system average BER can be calculated as: $\bar{\epsilon} = \sum_{k=1}^{B} \pi_k \epsilon_k$. Based on this prior work, we calculate the average BER for consecutive wireless links within a route segment in a cascaded system, and derive Lemma 3.1.1.

Lemma 3.1.1 *The BER in a cascade system where a node travels along links with states $\psi_{a_1}\psi_{a_2}...\psi_{a_n}$ ($1 \leq a_1, a_2, ..., a_n \leq K$) can be given by:*

$$\overline{\epsilon^n} \approx \sum_{\{\psi_{a_1}...\psi_{a_n}\}\in\mathcal{S}^n} \left(\pi_{a_1} \prod_{j=1}^{n-1} t_{a_j a_{j+1}} \sum_{i=1}^{n} \epsilon_i \right) \tag{3.1}$$

where Ψ^n represents all the possible set of series which is composed of n elements and each element is contained in Ψ (notice each series can have duplicated elements).

Proof Let

$$\mathcal{E}_i = \begin{bmatrix} 1-\epsilon_i & \epsilon_i \\ \epsilon_i & 1-\epsilon_i \end{bmatrix} \tag{3.2}$$

be the transition probability matrix when the packet's channel is in s_i. We then derive that

$$\mathcal{E}_i = \mathcal{B}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & 1-2\epsilon_i \end{bmatrix} \mathcal{B}, \tag{3.3}$$

where $\mathcal{B} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Then, we consider the situation that one bit goes through the cascade of n nodes and the bit's channel state is changed in the sequence of $\psi_{a_1}, \psi_{a_2}, ...\psi_{a_i}, ..., \psi_{a_n}$ $(1 \leq a_i \leq K,$ $1 \leq i \leq n)$. In this case, the transition probability matrix through n nodes, denoted as $\mathcal{E}_{a_1 a_2 ... a_n}$, is given by

$$\begin{aligned} \mathcal{E}_{a_1 a_2 ... a_n} &= \mathcal{E}_{a_1} \mathcal{E}_{a_2} ... \mathcal{E}_{a_n} && (3.4) \\ &= \mathcal{B}^{-1} \begin{bmatrix} 1 & 0 \\ 0 & \prod_{i=1}^{n} (1 - 2\epsilon_{a_i}) \end{bmatrix} \mathcal{B} && (3.5) \\ &= \begin{bmatrix} \frac{1+\prod_{i=1}^{n}(1-2\epsilon_{a_i})}{2} & \frac{1-\prod_{i=1}^{n}(1-2\epsilon_{a_i})}{2} \\ \frac{1-\prod_{i=1}^{n}(1-2\epsilon_{a_i})}{2} & \frac{1+\prod_{i=1}^{n}(1-2\epsilon_{a_i})}{2} \end{bmatrix} && (3.6) \end{aligned}$$

Thus, the BER of the cascade of n nodes (a_1, a_2, a_3, ..., a_n) equals:

$$\epsilon_{a_1 a_2 ... a_n} = \frac{1 - \prod_{i=1}^{n} (1 - 2\epsilon_{a_i})}{2} \tag{3.7}$$

The probability that such a aforementioned situation occurs equals:

$$\Pr\left[X = \{\psi_{a_1} \psi_{a_2} ... \psi_{a_n}\}\right] = \pi_{a_1} \prod_{j=1}^{n-1} t_{a_j a_{j+1}} \tag{3.8}$$

where X is a random variable represents the series. Then, the expectation of error bit through the cascade of n hops is given by:

$$\overline{\epsilon^n} = \sum_{\{\psi_{a_1} ... \psi_{a_n}\} \in \Psi^n} \Pr\left[X = \{\psi_{a_1} \psi_{a_2} ... \psi_{a_n}\}\right] \times \epsilon_{a_1 a_2 ... a_n} \tag{3.9}$$

$$= \sum_{\{\psi_{a_1} ... \psi_{a_n}\} \in \mathcal{S}^n} \pi_{a_1} \prod_{j=1}^{n-1} t_{a_j a_{j+1}} \frac{1 - \prod_{i=1}^{n} (1 - 2\epsilon_{a_i})}{2} \tag{3.10}$$

When ϵ_1, ϵ_2, ..., $\epsilon_n \ll 1$

$$\overline{\epsilon^n} \approx \sum_{\{\psi_{a_1} ... \psi_{a_n}\} \in \mathcal{S}^n} \left(\pi_{a_1} \prod_{j=1}^{n-1} t_{a_j a_{j+1}} \sum_{i=1}^{n} \epsilon_{a_i} \right) \tag{3.11}$$

3.1.2 Link Scheduling

Network model. We consider a wireless network with N communication links $L = \{(s_1, r_1), ..., (s_N, r_N)\}$, where (s_i, r_i) represents a transmission link from sender s_i to receiver r_i with transmission rate λ_i. We do not consider the scenario in which either a sender transmits to multiple receivers or multiple senders transmit to a receiver, so we assume that $s_i \neq s_j$ and $r_i \neq r_j \ \forall i \neq j$. The set of receivers and the set of senders are denoted by $R = \{r_1, ..., r_N\}$ and $S = \{s_1, ..., s_N\}$, respectively. For each receiver r_j, we call s_i the *desired sender* of r_j if $j = i$; otherwise an *interfering sender* of r_j. The Euclidean distance between sender s_i and receiver r_j is denoted by $d_{i,j}$ (or d_{s_i,r_j}), and that two senders s_i and s_j is denoted by d_{s_i,s_j}. We call $d_{i,i}$ the *length* of link (s_i, r_i). We assume a time slotted system with time slots normalized to integral units, so that slot boundaries occur at times $t \in \{0, 1, 2, ...\}$, and slot t refers to the time interval $[t, t+1)$. It is assumed that the length of every link are known at the beginning of each time slot.

Channel model. We consider time-varying and frequency-flat fading wireless channels. The channel effects from sender s_i to receiver r_j can be modeled by a single, complex and random channel coefficient $h_{i,j}$. We consider the Rayleigh fading channel model [42], in which all $|h_{i,j}|^2$ are independent and exponentially distributed with a mean value

$$\sigma_{i,j}^2 = P d_{i,i}^{-\alpha} \tag{3.12}$$

where α is path loss exponent. Equ. (3.12) actually describes the path loss for signal propagation in the case of far-field, i.e., when the transmission distance is larger than d_0 (i.e., a reference distance for the antenna far-field), where d_0 is typically assumed to be 1–10m indoors and 10–100m outdoors [17]. In this dissertation, we only consider the far-field case. Here, we set P by 1. Also, by convention, we assume that $\alpha > 2$. We use $Z_{i,j}$ to represent the instantaneous signal power received by r_j from s_i. $Z_{i,j}$ is a random variable with Cumulative Distribution Function (CDF) of $F_{Z_{i,j}} = \Pr\{Z_{i,j} \leq x\} = 1 - e^{-x/P d_{i,j}^{-\alpha}}$. When multiple users transmit simultaneously, they interfere with each other. We model interference by regarding all competing transmissions. We denote $Z_{\mathcal{P},j}$ as the sum signal that r_j receives from sender set $\mathcal{P}$ ($\mathcal{P} \subset S$), i.e., $Z_{\mathcal{P},j} = \sum_{s_i \in \mathcal{P}} Z_{i,j}$. We use a non-negative random variable X_j to represent the signal to interference ratio (SIR) received by r_j : $X_j = \frac{Z_{j,j}}{Z_{\mathcal{P}\setminus s_j,j}}$. Here, like [18,20], we ignore the noise power, which has no significant effect on the results. Receiver r_j can correctly decode the message (or informed) iff $X_j \geq \gamma_{\text{th}}$, where γ_{th} is decoding threshold ($\gamma_{\text{th}} = 1$).

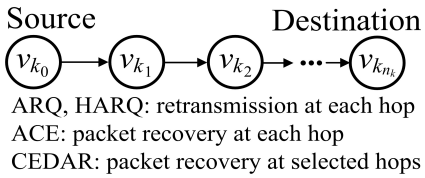


Figure 3.1: Protocols for packet recovery.

In fading channel models, the probability of successful transmission never can be 0, so we assume an acceptable error probability ε for transmission. That is, for any receiver r_j, we say r_j can be *informed* by its desired sender s_j if the probability of $X_j < \gamma_{\text{th}}$ is no larger than ε.

3.2 Problem formulation

In this dissertation, there are two problems to be solved: the *relay selection* problem for reducing the average packet latency and the *link scheduling* problem for increasing the throughput.

3.2.1 Relay Selection Problem for Packet Recovery

As shown in Fig. 3.1, to reach the destination, each packet flow needs to travel through all nodes in the predetermined route, and some of these nodes are responsible for en/decoding the packets. In the ARQ and HARQ protocols [11, 57, 67], each hop drops distorted packets and requests for complete or partial retransmission of the original packets. These methods follow the conventional link-layer design paradigm and guarantee the reliability between any pair of nodes. However, this strategy causes high delays and low throughput (due to numerous retransmissions at every relay hop), leading to significant degradation in channel bandwidth utilization. Further, although decoding in each hop (adopted by the HARQ family) increases the reliability, it comes at the cost of high en/decoding overhead. In the existing proposed schemes (e.g., ACE [15]), each relay node stores an erroneous received packet for packet recovery (with no retransmission requirements) until the packet is corrected before forwarding it to the next hop. Though these schemes overcome the shortcomings of the ARQ and HARQ protocols to a certain extent, they are still not effective in achieving high throughput, and low energy and bandwidth consumption.

CEDAR introduces a new flexible environment for link-layer error recovery: (1) it employs a theoretically-sound framework and a corresponding strategy for embedding channel codes, using robust and adaptive code rates, in each packet; (2) the error correction process is performed in

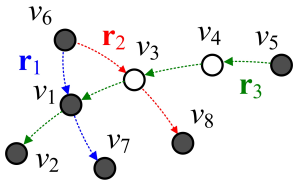


Figure 3.2: Route segment.

a distributed and suboptimal manner where selected (and not all) intermediate nodes participate in performing error recovery. The key problem in CEDAR is how to identify candidates among the intermediate nodes for the CEDAR en/decoding process to decrease the overall delay, increase throughput and fairness of en/decoding load over the entire network.

To this end, first, we build models to calculate the delay ($D(\mathbf{n}_i)$) and the en/decoding load ($L(\mathbf{n}_i)$) of each intermediate node v_i based on the lengths of the routing paths (denoted by $\mathbf{n}_i$) of the packets crossing v_i. We use these models to calculate the expected delays and en/decoding load of each node, and ultimately identify the positions of intermediate nodes for en/decoding in each route in CEDAR. Throughout the paper, we use the key terms provided in the following definitions:

Definition 3.2.1 *(*En/Decoding load*) The en/decoding load of v_i, denoted by $L(\mathbf{n}_i)$, is defined as the sum of the arrival rates for all the packet streams that v_i is responsible for en/decoding.*

Definition 3.2.2 *(*Key node*) A key node of route $\mathbf{r}_k$ is a node responsible for en/decoding the packets traveling along $\mathbf{r}_k$. Matrix $\mathbf{X} = (x_{i,k})_{N\times K}$ denotes whether v_i is a key node in $\mathbf{r}_k$:*

$$x_{i,k} = \begin{cases} 1, & v_i \text{ is the key node in } \mathbf{r}_k; \\ 0, & v_i \text{ is not the key node in } \mathbf{r}_k. \end{cases} \tag{3.13}$$

Definition 3.2.3 *(*Route segment*) A route segment of $\mathbf{r}_k$ is a section of the end-to-end path between one key node to either the endpoints or another key node. The length of a route segment is defined as the number of hops in the route segment.*

In each route segment, the packet sender (the first key node) encodes the packets and the packet receiver (the second key node) decodes the packets. In other words, the second key node is responsible for decoding for its route segment. Use matrix $\mathbf{N} = (n_{i,k})_{N\times K}$ to denote the length of a route segment with decoding node v_i in $\mathbf{r}_k$ and use vector $\mathbf{n}_i$ denote the lengths of route

segments responsible by v_i for all the routes, i.e., $\mathbf{N} = [\mathbf{n}_1, ..., \mathbf{n}_N]^T$ and $\mathbf{n}_i = [n_{i,1}, ..., n_{i,K}]$. Here we define $n_{i,k} = 0$ if v_i has no responsibility of decoding the packet in $\mathbf{r}_k$. For example, in Fig. 3.2, there are eight nodes $\mathcal{V} = \{v_1, ..., v_8\}$, and three routes $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, \mathbf{r}_3\}$, where $\mathbf{r}_1 = \{v_6, v_1, v_7\}$, $\mathbf{r}_2 = \{v_6, v_3, v_8\}$ and $\mathbf{r}_3 = \{v_5, v_4, v_3, v_1, v_2\}$. v_6, v_1 and v_7 are the key nodes in $\mathbf{r}_1$; v_6, v_3 and v_8 are the key nodes in $\mathbf{r}_2$; v_5, v_1 and v_2 are the key nodes in $\mathbf{r}_3$. Then, we can derive that $n_{1,1} = 1$ and $n_{1,3} = 3$ because there are one hop from v_6 to v_1 in $\mathbf{r}_1$ and three hops from v_5 to v_1 in $\mathbf{r}_3$. Also, $n_{1,3} = 0$ because v_1 is not responsible for decoding packets in $\mathbf{r}_3$. Hence, $\mathbf{n}_1 = [n_{1,1}\ n_{1,2}\ n_{1,3}] = [1\ 3\ 0]$.

Let $\lambda_{i,k}$ denote the arrival rate of the data stream that v_i is responsible for en/decoding in $\mathbf{r}_k$. Then $\lambda_{i,k} = \lambda_k \times x_{i,k}$. We use $\overline{D(\mathbf{n}_i)}$ to denote the average delay when a packet crosses v_i with route segment vector $\mathbf{n}_i$. Since the the number of packets at v_i within a unit time is $\sum_{k=1}^{K} \lambda_{i,k}$, then the total average packet delay decoding at v_i within a unit time equals $\overline{D(\mathbf{n}_i)} \sum_{k=1}^{K} \lambda_{i,k}$. Also, we use $\overline{L(\mathbf{n}_i)}$ to represent the average en/decoding load of v_i and use $\overline{L(\mathbf{N})}$ to represent the average en/decoding load of all the nodes in $\mathcal{V}$. Then, $\overline{L(\mathbf{N})} = \frac{\sum_{i=1}^{N} \overline{L(\mathbf{n}_i)}}{N}$.

Objective. The objectives of CEDAR are 1) to minimize the total delay of the packets in the entire system, which can be represented as:

$$\min \sum_{i=1}^{N} \left(\overline{D(\mathbf{n}_i)} \sum_{k=1}^{K} \lambda_{i,k} \right) \tag{3.14}$$

and 2) to balance the en/decoding load of all the nodes in the network. In this paper we use standard deviation [8] of en/decoding loads, which reflects how much variation exists between each node's en/decoding load and $\overline{L(\mathbf{N})}$, to measure the balance of the en/decoding load of the network. The lower value of the standard deviation, the higher fairness of the en/decoding load of all the nodes. Then, the objective can be formulated as:

$$\min \sqrt{\sum_{i=1}^{N} \left(\overline{L(\mathbf{n}_i)} - \overline{L(\mathbf{N})} \right)^2} \tag{3.15}$$

Consequently, we combine these two objectives and formulate the optimization problem as:

$$\min \quad \gamma_1 \sum_{i=1}^{N} \left(\overline{D(\mathbf{n}_i)} \sum_{k=1}^{K} \lambda_{i,k} \right) + \gamma_2 \sqrt{\sum_{i=1}^{N} \left(\overline{L(\mathbf{n}_i)} - \overline{L(\mathbf{N})} \right)^2} \tag{3.16}$$

$$\text{s.t.} \qquad x_{i,k} \leq y_{i,k},\ \overline{D\left(\mathbf{n}_i\right)} \leq U_k, \tag{3.17}$$

$$1 \leq i \leq N, 1 \leq k \leq K \tag{3.18}$$

where γ_1 and γ_2 represent the weights we set for these two objectives. In this paper, we primarily consider minimizing packet delay and secondarily consider balancing en/decoding load. Thus, we set $\gamma_1 >> \gamma_2$ in our system.

Now we need to consider how to solve the multiple objective optimization problem: the packet delay in v_i (which is composed of prop&tran and queueing delays) is a function of $\mathbf{n}_i$. This will be deduced in the mathematical analysis in the next section. We use $\overline{D_{\text{p\&t}}\left(\mathbf{n}_i\right)}$ to denote the average prop&tran delay of all the packet streams being decoded in v_i, and use $\overline{D_{\text{q}}\left(\mathbf{n}_i\right)}$ to denote the average queuing delay of the packet stream in v_i. Then, the total average delay when one packet crosses v_i is:

$$\overline{D\left(\mathbf{n}_i\right)} = \overline{D_{\text{q}}\left(\mathbf{n}_i\right)} + \overline{D_{\text{p\&t}}\left(\mathbf{n}_i\right)} \tag{3.19}$$

Thus, we need to minimize $\overline{D_{\text{q}}\left(\mathbf{n}_i\right)}$ and $\overline{D_{\text{p\&t}}\left(\mathbf{n}_i\right)}$ in order to achieve the objective of CEDAR in Formula (3.16). To this end, we model the Bit Error Rate (BER) fluctuations of wireless channels and probability of successful decoding. Sections 4.1.1 and 4.2 use this model to formulate the prop&tran delay $\overline{D_{\text{p\&t}}\left(\mathbf{n}_i\right)}$ and the queuing delay $\overline{D_{\text{q}}\left(\mathbf{n}_i\right)}$. Finally, Section 4.3 derives two propositions to minimize $\overline{D_{\text{q}}\left(\mathbf{n}_i\right)}$ and $\overline{D_{\text{p\&t}}\left(\mathbf{n}_i\right)}$, respectively. Guided by the propositions, we design CEDAR in Section 4.5.

3.2.2 Link Scheduling for Throughput

3.2.2.1 Fading resistant link scheduling

First, we formulate the *Fading-R-LS* problem. Its objective is to identify a subset of senders, denoted by $\mathcal{P}$ ($\mathcal{P} \subset \mathcal{S}$), such that the throughput (i.e., the total data rate successfully received by receivers) is maximized in one time slot. In other words, we attempt to use one time slot to its full capacity. Formally, we define the decision version of *Fading-R-LS* as follows:

Instance: A finite set of senders $\mathcal{S}$ and their respective receivers $\mathcal{R}$ in a geometric plane, decoding threshold γ_{th}, acceptable error rate ε, and a constant Λ.

Question: Existence of a subset of senders $\mathcal{P}$, namely a *schedule*, such that the total successful transmission rate is no smaller than Λ, *i.e.*, 1) $\Pr(X_j < \gamma_{\text{th}}) < \varepsilon,\ \forall s_j \in \mathcal{P}$ and 2) $\sum_{s_j \in \mathcal{P}} \lambda_j \geq \Lambda$.

We say a schedule $\mathcal{P}$ is *feasible* if all the senders in $\mathcal{P}$ can successfully transmit the message with probability at least $1-\varepsilon$. Below, we first derive the closed-form expression for the probability of successful transmission $\Pr(X_j \geq \gamma_{\text{th}})$ for any receiver r_j (Theorem 3.2.1). Then, we prove that *Fading-R-LS* is NP-hard (Theorem 3.2.5).

Theorem 3.2.1 *Given an active link (s_j, r_j) and active sender set $\mathcal{P}$, the probability of successful transmission from s_j to r_j is:*

$$\Pr(X_j \geq \gamma_{\text{th}}) = \prod_{s_i \in \mathcal{P} \setminus s_j} \frac{1}{1 + \frac{d_{i,j}^{-\alpha} \gamma_{\text{th}}}{d_{j,j}^{-\alpha}}}. \tag{3.20}$$

Proof The CDF of the quotient $X_j = \frac{Z_{i,j}}{Z_{\mathcal{P}\setminus s_j, j}}$ can be computed as follows:

$$\begin{aligned} F_{X_j}(x) &= \mathrm{P}\left(\frac{Z_{j,j}}{Z_{\mathcal{P}\setminus s_j, j}} \leq x\right) && (3.21)\\ &= P(Z_{j,j} \leq x Z_{\mathcal{P}\setminus s_j, j}) && (3.22)\\ &= \int_0^\infty \int_0^{xz} f_{Z_{j,j}}(y)\mathrm{d}y \cdot f_{Z_{\mathcal{P}\setminus s_j, j}}(z)\mathrm{d}z. && (3.23) \end{aligned}$$

By differentiating, we can obtain

$$\begin{aligned} f_{X_j}(x) &= \frac{\mathrm{d}}{\mathrm{d}x} F_{X_j}(x) && (3.24)\\ &= \int_0^\infty z f_{Z_{j,j}}(xz) f_{Z_{\mathcal{P}\setminus s_j, j}}(z)\mathrm{d}z && (3.25)\\ &= \int_0^\infty \frac{z}{P d_{i,j}^{-\alpha}} e^{-\frac{xz}{P d_{i,j}^{-\alpha}}} f_{Z_{\mathcal{P}\setminus s_j, j}}(z)\mathrm{d}z. && (3.26) \end{aligned}$$

Then, the probability of successful transmission from s_j to r_j equals

$$\begin{aligned} \Pr(X_j \geq \gamma_{\text{th}}) &= \int_0^\infty \int_{\gamma_{\text{th}}}^\infty \frac{z}{P d_{j,j}^{-\alpha}} e^{-\frac{xz}{P d_{j,j}^{-\alpha}}} f_{Z_{\mathcal{P}\ s_j, j}}(z)\mathrm{d}x\mathrm{d}z && (3.27)\\ &= \int_0^\infty e^{-\frac{\gamma_{\text{th}} z}{P d_{j,j}^{-\alpha}}} f_{Z_{\mathcal{P}\ s_j, j}}(z)\mathrm{d}z && (3.28)\\ &= \mathcal{L}_{Z_{\mathcal{P}\setminus s_j, j}}\left(\frac{\gamma_{\text{th}}}{P d_{j,j}^{-\alpha}}\right) && (3.29) \end{aligned}$$

where $\mathcal{L}_{Z_{\mathcal{P}\setminus s_j, j}}(\nu)$ represents the Laplace transform of $f_{Z_{\mathcal{P}\setminus s_j, j}}(x)$. Because the Laplace transform of the exponential distribution with mean $1/\mu$ equals $\mu/(\mu+\nu)$, $\mathcal{L}_{Z_{\mathcal{P}\setminus s_j, j}}(\nu) = \prod_{s_i \in \mathcal{P}\setminus s_j} \frac{1}{1+P d_{i,j}^{-\alpha} \nu}$.

Consequently,

$$\Pr(X_j \geq \gamma_{\text{th}}) = \mathcal{L}_{Z_{\mathcal{P}\setminus s_j,j}}\left(\frac{\gamma_{\text{th}}}{P d_{j,j}^{-\alpha}}\right) \tag{3.30}$$

$$= \prod_{s_i \in \mathcal{P}\setminus s_j} \frac{1}{1+\frac{d_{i,j}^{-\alpha}\gamma_{\text{th}}}{d_{j,j}^{-\alpha}}}. \tag{3.31}$$

According to Theorem 3.2.1, in the following, we formulate the ILP form of the *Fading-R-LS* problem and prove that this problem is NP-hard. First, we take the logarithm on both sides of Equ. (3.20):

$$\ln \Pr(X_j \geq \gamma_{\text{th}}) = -\sum_{s_i \in \mathcal{P}\setminus s_j} f_{i,j}, \tag{3.32}$$

where

$$f_{i,j} = \begin{cases} \ln\left(1+(d_{i,j}/d_{j,j})^{-\alpha}\gamma_{\text{th}}\right) & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases} \tag{3.33}$$

We call $f_{i,j}$ the *interference factor* of s_i on r_j. Accordingly, we use $f_{\mathcal{P}\setminus r_j,r_j}$ to denote the interference factor of $\mathcal{P}\setminus r_j$ on r_j, where

$$f_{\mathcal{P}\setminus r_j,r_j} = \sum_{s_i \in \mathcal{P}\setminus r_j} f_{i,j}. \tag{3.34}$$

Corollary 3.2.1 *Given an active link (s_j, r_j) and the active sender set $\mathcal{P}$, r_j can be informed iff*

$$\sum_{s_i \in \mathcal{P}\setminus s_j} f_{i,j} \leq \gamma_\varepsilon, \tag{3.35}$$

where $\gamma_\varepsilon = \ln\left(\frac{1}{1-\varepsilon}\right)$ is a constant.

By Corollary 3.2.1, we formulate the ILP form of *Fading-R-LS* as follows:

$$\max \sum_{i=1}^{N} \lambda_i x_i \tag{3.36}$$

$$\text{s.t.} \sum_{i=1}^{N} f_{i,j} x_i \leq \gamma_\varepsilon + M(1-x_j) \; j = 1, ..., N, \tag{3.37}$$

$$x_i \in \{0,1\} \;\; i = 1, ..., N, \tag{3.38}$$

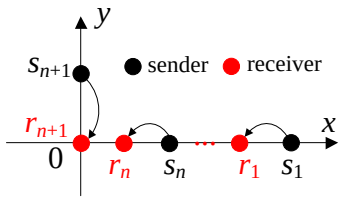


Figure 3.3: Mapping Fading-R-LS to Knapsack.

where M is a constant with a very large value.

Theorem 3.2.2 *The* Fading-R-LS *problem is NP-hard.*

Proof We construct a polynomial time reduction from the well-known Knapsack NP-hard problem [38] to *Fading-R-LS*. The Knapsack problem can be formulated as follows: given n kinds of items, $x_1, ..., x_n$; each item x_j has a value p_j and a weight w_j, and a bag that can carry weight W maximally, the goal is to choose the items to put into the bag such that the sum of the items' values is no smaller than a constant C. We construct a *Fading-R-LS* instance that can be mapped to the Knapsack problem (see Fig. 3.3). We position a sender node s_i in the plane for each x_i, such that the received signal power from s_i at $(0,0)$ is w_i, i.e.,

$$\text{Loc}(s_i) = \left(\left(\frac{e^{\frac{\gamma_\varepsilon w_i}{W}} - 1}{\gamma_{\text{th}}} \right)^{-\frac{1}{\alpha}}, 0 \right), \ \forall 1 \leq j \leq n. \tag{3.39}$$

Then, we set r_i close enough to s_i to guarantee successful reception regardless of other links.

$$\text{Loc}(r_i) = \text{Loc}(s_i) + (\delta, 0), \ \forall 1 \leq i \leq n, \tag{3.40}$$

where $\delta = d_{\min} / \left(\left((e^{\gamma_\varepsilon/(n+1)} - 1)/\gamma_{\text{th}} \right)^{-\frac{1}{\alpha}} + 1 \right)$, and $d_{\min}$ is the minimum distance between any pair of senders. After that, we place one more link l_{n+1}, s.t.

$$\text{Loc}(s_{n+1}) = (0,1), \ \text{Loc}(r_{n+1}) = (0,0). \tag{3.41}$$

Thereafter, we assign a weight to each link:

$$\lambda_i = p_i, \ \forall 1 \leq i \leq n \ \lambda_{n+1} = 2 \sum_{j=1}^{n} p_j. \tag{3.42}$$

The question is whether there exists a schedule to make total data rate (i.e., throughput) at least $2\sum_{j=1}^{n} p_j + C$ for *Fading-R-LS*. Now, we need to prove that the solution of the *Fading-R-LS* instance exists iff the solution of the Knapsack problem exists.

$\Rightarrow$: Suppose that $\exists \mathcal{X}$ *s.t.* $\sum_{x_j \in \mathcal{X}} p_j \geq C$ and $\sum_{x_j \in \mathcal{X}} w_j \leq W$. We activate each sender s_i if $x_i \in \mathcal{X}$. Also, s_{n+1} must be active; otherwise the total data can never reach $2\sum_{j=1}^{n} p_j + C$. First, r_{n+1} can successfully receive the packet because

$$\sum_{s_i \in \mathcal{P} \setminus s_{n+1}} f_{j,n+1} = \sum_{s_i \in \mathcal{P} \setminus s_{n+1}} \ln\left(1 + \frac{d_{i,j}^{-\alpha}\gamma_{\text{th}}}{d_{n+1,n+1}^{-\alpha}}\right) \tag{3.43}$$

$$= \sum_{s_i \in \mathcal{P} \setminus s_{n+1}} \ln\left(1 + \left(\frac{e^{\frac{\gamma_\varepsilon w_i}{W}} - 1}{\gamma_{\text{th}}}\right)\gamma_{\text{th}}\right) \tag{3.44}$$

$$\leq \gamma_\varepsilon \tag{3.45}$$

Then, for each receiver r_j s.t. $x_j \in \mathcal{X}$, we have $\lambda_{total} = \sum_{x_i \in X} p_i x_i + 2W \geq C + 2W$, which implies that exists a schedule such that total data rate is at least $2\sum_{j=1}^{n} p_j + C$ for *Fading-R-LS*.

$\Leftarrow$: Suppose there exists a *Fading-R-LS* schedule $\mathcal{P}$ such that the total data rate is at least $2\sum_{j=1}^{n} p_j + C$, then r_{n+1} must successfully receive the message, and hence $\sum_{s_j \in \mathcal{P} \setminus s_{n+1}} f_{j,n+1} \leq \gamma_\varepsilon$, which implies that $\sum_{s_i \in \mathcal{P}\ s_j} w_j \leq W$ (by Equ. (3.59)) and $\sum_{s_i \in \mathcal{P}\ s_j} p_j \geq C$. Let $\mathcal{X} = \{x_i | s_i \in \mathcal{P}/s_{n+1}\}$, then $\sum_{x_i \in \mathcal{X}} p_i \geq C$ and $\sum_{x_i \in \mathcal{X}} w_i \leq W$.

3.2.2.2 CC Link scheduling

In this section, we formulate two problems, named CLS and OCLS, and prove that the problems are NP-hard.

The CLS problem. For the CLS problem, we determine the set of active links at each time slot. Hence, a CLS schedule can be represented by a link set sequence $\mathcal{I}_{\text{cls}} = \{\mathcal{I}^1, ..., \mathcal{I}^T\}$, where $\mathcal{I}^t$ is the set of active links at time slot t and T is the number of time slots the schedule takes. We say a CLS schedule is *feasible* iff this schedule enables every intended receiver to be informed. The objective of the CLS problem is to find a feasible CLS schedule that takes the minimum number of time slots. Formally, the decision version of CLS is defined as follows:

Instance: A finite set of nodes in a geometric plane V, a set of requests $F = \{f_1, ..., f_N\}$ (each request $f_i \in F$ has a set of links $\mathcal{I}_i$ and a receiver r_i), and constants γ_{th} and T.

Question: Existence of a CLS schedule $\mathcal{I}_{\text{cls}}$ *s.t.*

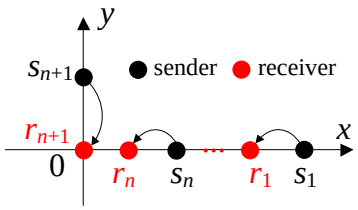


Figure 3.4: An instance of CLS that maps to the Partition problem.

- $\mathcal{I}^t \cap \mathcal{I}^{t'} = \phi \;\; \forall 1 \leq t < t' \leq T$;
- each r_i can be informed by time slot T.

Theorem 3.2.3 *The CLS problem in* SIR *is NP-hard.*

Proof We construct a polynomial time reduction of the well known NP-hard problem, *Partition problem* [38], to CLS. The Partition problem can be formulated as follows: given a finite set of integers $\mathcal{X} = \{x_1, x_2, ..., x_n\}$, find $\mathcal{X}_1 \subset \mathcal{X}$ *s.t.*

$$\sum_{x_j \in \mathcal{X}_1} x_j = \frac{1}{2} \sum_{x_j \in \mathcal{X}} x_j = \frac{\sigma}{2} \tag{3.46}$$

We construct the following CLS instance (see Fig. 3.4) maps to the Partition problem. There are two data flows f_1 and f_2 in the network, where

$$f_1 = (\{l_{s_1,r_1}, ..., l_{s_n,r_1}\}, r_1) \text{ and } f_2 = \left(\{l_{s_{n+1},r_2}\}, r_2\right). \tag{3.47}$$

The locations of these nodes are set by

1) $\text{Loc}(r_1) = \text{Loc}(r_2) = (0, 0)$;
2) $\text{Loc}(s_j) = (x_j^{-1/\alpha}, 0), \;\; j = 1, ..., n$;
3) $\text{Loc}(s_{n+1}) = (0, \sigma/2)$.

Let decoding threshold γ_{th} and time constraint T equal to 1. The question is whether there exists a schedule to make both r_1 and r_2 be informed by the end of slot 1. Now, we need to prove that the solution of the CLS instance exists iff the solution of the Partition problem exists.

$\Rightarrow$: Suppose that $\exists \mathcal{X}_1$ *s.t.* $\sum_{x_j \in \mathcal{X}_1} x_j = \sigma/2$, we use l_{s_j,r_1} along with l_{s_{n+1},r_2}. Then,

$$\mathrm{SIR}_{r_1} = \frac{\sum_{x_j \in \mathcal{X}_1} (x_j^{-1/\alpha})^{-\alpha}}{((\sigma/2)^{1/\alpha})^{\alpha}} = \frac{\sum_{x_j \in \mathcal{X}_1} x_j}{\sigma/2} = 1 \tag{3.48}$$

and

$$\mathrm{SIR}_{r_2} = \frac{1}{\mathrm{SIR}_{r_1}} = 1, \tag{3.49}$$

which implies both r_1 and r_2 can be informed.

$\Leftarrow$: Suppose that the above CLS instance has a solution, which implies that $\exists \mathcal{X}_1$ *s.t.*

$$\mathrm{SIR}_{r_2} = \frac{\sigma}{2X} \geq 1 \Rightarrow X \leq \sigma/2 \tag{3.50}$$

$$\mathrm{SIR}_{r_1} = \frac{\sigma}{2(\sigma - X)} \geq 1 \Rightarrow X \geq \sigma/2, \tag{3.51}$$

where $X = \sum_{x_j \in \mathcal{X}_1} x_j$. From Equ. (3.50) and Equ. (3.51), we can get that $X = \sigma/2$. Hence, the Partition problem has a solution.

The OCLS problem. In contrast to the CLS problem, which aims to inform all the receivers using the minimum number of time slots, the objective of the OCLS problem is to pick a subset of links, denoted by $\mathcal{I}_{\mathrm{ocls}}$, such that the number of receivers to be informed is maximized. In other words, we attempt to use one slot to its full capacity. Formally, we define the decision version of the OCLS problem as follows:

Instance: A finite set of nodes in a geometric plane V, a set of requests $F = \{f_1, ..., f_N\}$ (each request $f_i \in F$ has a set of links $\mathcal{I}_i$ and a receiver r_i), and constants γ_{th} and M.

Question: Existence of a subset of links $\mathcal{I}_{\mathrm{ocls}}$ *s.t.* at least M receivers can be informed.

Theorem 3.2.4 *The OCLS problem in* $\mathrm{SIR}_{\mathrm{G}}$ *is NP-hard.*

Proof Note that the CLS instance constructed in the proof of Theorem 3.2.3 is also an instance of OCLS, where $f_1 = (\{l_{s_1,r}, ..., l_{s_n,r}\}, r_1)$ and $f_2 = \big(\{l_{s_{n+1},r}\}, r_2\big)$, $\gamma_{\mathrm{th}} = 1$, and $M = 2$. Hence, we can construct a polynomial time reduction of the *Partition problem* to the OCLS problem, which implies that OCLS is NP-hard.

3.2.2.3 Vehicle Link Scheduling

Below, we first formulate the *Vehicle Link Scheduling (VLS) problem*, and then prove that the problem is NP-hard (Theorem 3.2.5). Formally, the VLS problem is defined as follows:

Instance: A finite set of senders $\mathcal{S}$ and their respective receivers $\mathcal{R}$ in a geometric plane, decoding threshold γ_{th}, and a constant Λ.

Question: Using Λ channels, whether there exists a schedule (which allocates a channel to each vehicle sender), such that the SINR received by each vehicle receiver is higher than γ_{th}?

Theorem 3.2.5 *The VLS problem is NP-hard.*

Proof We construct a polynomial time reduction from the well-known Partition problem [38] to VLS. The Partition problem can be formulated as follows: given n kinds of items, $\mathcal{N} = \{1, ..., n\}$ and each item i has a value p_i. The goal is to find $\mathcal{N}' \subset \mathcal{N}$ such that

$$\sum_{i \in \mathcal{N}'} p_i = \frac{1}{2} \sum_{i \in \mathcal{N}} p_i \tag{3.52}$$

We construct an instance of the VLS problem to map to the Partition problem. We position a vehicle sender s_i in the plane for each item i, such that the received signal power from s_i at $(0, 0)$ is p_i, i.e.,

$$x_{s_i} = \left(\left(\frac{P}{p_i} \right)^{\frac{1}{\alpha}}, 0 \right), \ \forall 1 \leq j \leq n. \tag{3.53}$$

Then, we set r_i close enough to s_i to guarantee successful reception regardless of other links.

$$x_{r_i} = x_{s_i} + \delta, \ \forall 1 \leq i \leq n. \tag{3.54}$$

where δ is close to 0. After that, we place one more link l_{n+1}, s.t.

$$x_{s_{n+1}} = 1, \ x_{r_{n+1}} = 0. \tag{3.55}$$

Set Λ by 2. Then, the question is whether there exists a schedule such that all the receivers can receive the packet in two channels. Now, we need to prove that the solution of the VLS problem exists iff the solution of the Partition problem exists.

$\Rightarrow$: Suppose that $\exists \mathcal{N}'$ *s.t.*

$$\sum_{i\in\mathcal{N}'} p_i = \frac{1}{2}\sum_{i\in\mathcal{N}} p_i \tag{3.56}$$

In the first channel, we activate each sender s_i if $i \in \mathcal{N}'$ and s_{n+1}. Each r_i can receive the packet since the distance between the sender and the receiver (i.e., δ) is small enough. Also, r_{n+1} can successfully receive the packet because the SINR received by receiver r_{n+1} is equal to

$$\frac{\left(\left(\frac{P}{\frac{1}{2}\sum_{i\in\mathcal{N}} p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P}{\sum_{i\in\mathcal{N}'}\left(\left(\frac{P}{p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P} = \frac{\frac{1}{2}P^{-1}P\sum_{i\in\mathcal{N}} p_i}{\sum_{i\in\mathcal{N}'}\frac{p_i}{P}P} \tag{3.57}$$

$$= \frac{\frac{1}{2}\sum_{i\in\mathcal{N}} p_i}{\sum_{i\in\mathcal{N}'} p_i} = 1. \tag{3.58}$$

Then, in the second channel, we activate each sender s_i if $i \in \mathcal{N}\backslash\mathcal{N}'$ and s_{n+2}. Similarly, each r_i can receive the packet and r_{n+2} can successfully receive the packet because the SINR received by receiver r_{n+2} is equal to

$$\frac{\left(\left(\frac{P}{\frac{1}{2}\sum_{i\in\mathcal{N}} p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P}{\sum_{i\in\mathcal{N}'}\left(\left(\frac{P}{p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P} = \frac{\frac{1}{2}\sum_{i\in\mathcal{N}\backslash\mathcal{N}'} p_i}{\sum_{i\in\mathcal{N}'} p_i} = 1. \tag{3.59}$$

$\Leftarrow$: Suppose there exists a schedule such that all the packets can be received within the two channels. First, r_{n+1} and r_{n+2} cannot receive the packet in the same channel since they will interfere with each other. Without loss of generality, let r_{n+1} and r_{n+2} be scheduled in the first channel and second channel, respectively. Let $\mathcal{N}'$ and $\mathcal{N}\backslash\mathcal{N}'$ be the indices of senders allocated in the first channel and second channel, respectively. Then, since r_{n+1} and r_{n+2} can receive the packet, we have

$$\frac{\left(\left(\frac{P}{\frac{1}{2}\sum_{i\in\mathcal{N}} p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P}{\sum_{i\in\mathcal{N}'}\left(\left(\frac{P}{p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P} \geq 1 \Rightarrow \frac{1}{2}\sum_{i\in\mathcal{N}} p_i \geq \sum_{i\in\mathcal{N}'} p_i \tag{3.60}$$

$$\frac{\left(\left(\frac{P}{\frac{1}{2}\sum_{i\in\mathcal{N}} p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P}{\sum_{i\in\mathcal{N}\setminus\mathcal{N}'}\left(\left(\frac{P}{p_i}\right)^{-\frac{1}{\alpha}}\right)^{\alpha} P} \geq 1 \Rightarrow \frac{1}{2}\sum_{i\in\mathcal{N}} p_i \geq \sum_{i\in\mathcal{N}\setminus\mathcal{N}'} p_i. \tag{3.61}$$

which implies that

$$\sum_{i\in\mathcal{N}'} p_i = \frac{1}{2}\sum_{i\in\mathcal{N}} p_i. \tag{3.62}$$

Chapter 4

Relay Selection for Packet Recovery

Recall that in Section 3.1.1, we have presented a Markovian wireless channel model to capture the variations in wireless error conditions due to non-stationary wireless noise and calculate BER of a packet when it goes through several channels. Using this model, in this section, we analyze the relationship between the number of hops a packet goes through and the probability of its successful decoding. This relationship leads us to calculate the prop&tran delay and queuing delay, respectively. By minimizing the two delays, we can find the locations of intermediate nodes in a route for decoding. Finally, we formulate the problem of minimizing the sum of the delays as a non-linear integer programming problem. The analytical results and the formed problem lay the foundation for the design of an optimized strategy for choosing intermediate nodes for the CEDAR packet recovery.

4.1 Probability of Successful Decoding

CEDAR is developed based on the error recovery mechanism in the ACE Communication Model [55]. Thus, we first introduce ACE before we present the mathematical models. Specifically, during τ_i, a transmitter encodes data symbols z_i with parity codes x_i (referred as type-I parity code) to create a codeword $C_i\,(z_i, x_i)$. It transmits a packet $M_i = (C_i(z_i, x_i), y_i)$, where y_i denotes

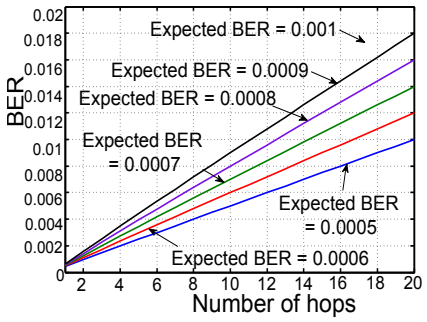


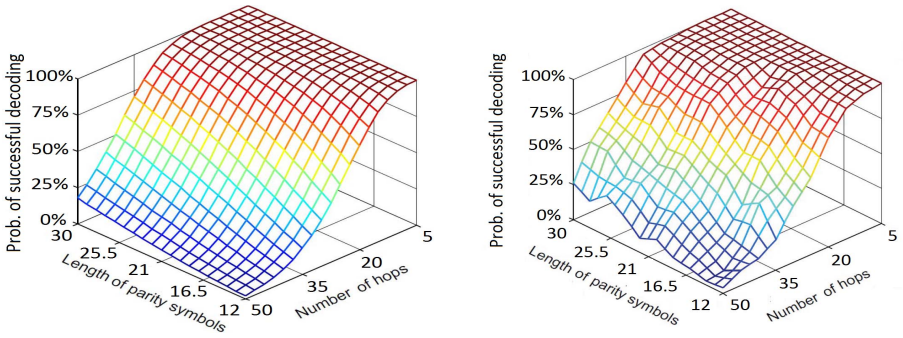
Figure 4.1: The curve of Equation (3.1).

the additional parity (hereafter type-II parity) symbols for recovering previously received corrupted packets at the receiver. We also use x_i, y_i and z_i to denote the number of their symbols. The receiver utilizes x_i to decode C_i. If the decoding operation fails, the receiver stores C_i in its buffer and issues a request along with ACK_i for more parity symbols. The transmitter then sends additional parity $y_j (j > i)$ along with M_j. I use $m_i = x_i + y_i$ to denote the total number of parity symbols of M_i.

First, consider a simple cascade model ($v_0 \rightarrow v_1 \rightarrow ... \rightarrow v_n$) in which a packet stream goes through a series of nodes $v_0, v_1, v_2, ..., v_n$ and is encoded and decoded at v_0 and v_n, respectively. Fig. 4.1 shows an example of the curve of Equation (3.1), where n is varied from 1 to 10, and the expected value of $\bar{\epsilon}$ is varied from 0.0005 to 0.001. We assume the channel has two states: noisy and not noisy, and each state can transfer to the other state at the next time slot with the same probability. From the figure, we can find that $\overline{\epsilon^n}$ is approximately proportional to n. Thus, we drew figure to based on Equation (3.1) and found that $\overline{\epsilon^n}$ is approximately proportional to n. I can approximate Equ. (3.1) by Equ. (4.1) to calculate the BER for a routing through n nodes under the Markovian channel model:

$$\overline{\epsilon^n} \approx n\bar{\epsilon} \tag{4.1}$$

As ACE, we take Reed-Solomon codes [63] as an example, which is a kind of non-binary cyclic error-correcting codes, for channel coding. In the Reed-Solomon codes, each symbol is composed of b bits, indicating that the probability of error for each symbol equals: $\overline{\epsilon^{n,b}} = 1 - (1 - \overline{\epsilon^n})^b$. The number of error symbols introduced in one packet M_i with a length of $z_i + m_i$ symbols through n hops can be represented by a random variable E_i following a binomial distribution $E_i \approx Bi\left(z_i + m_i, \epsilon^{n,b}\right)$. If the error estimate is $\hat{\epsilon}^{n,b}$ for one symbol of b bits, the receiver is capable of correcting up to αm_i errors out of $|C_i|$ symbols in packet M_i, where α is a function measuring the expected error-correcting



(a) The surface of $F_\alpha(n, m_i)$ (b) Simulation results

Figure 4.2: Comparison of Equation (4.3) and simulation.

capability of a particular decoder based on $\hat{\epsilon}^{n,b}$. For instance, the error-correcting capability of the Reed-Solomon codes is half as many as redundant symbols (i.e., $\alpha = 0.5$) [63]. The probability of successfully recovering data bits by a parity code with m_i length symbols equals:

$$\Pr[E_i \leq \alpha m_i] = \sum_{k=0}^{\lfloor \alpha m_i \rfloor} \begin{pmatrix} z_i + m_i \\ k \end{pmatrix} \left(1 - \hat{\epsilon}^{n,b}\right)^{z_i + m_i - k} \left(\hat{\epsilon}^{n,b}\right)^k \tag{4.2}$$

From Equ. (4.2), we observe that the probability of successful is a discrete function of two variables:

$$\Pr[E_i \leq \alpha m_i] = F_\alpha(n, m_i). \tag{4.3}$$

$F_\alpha(n, m_i)$ is monotonically decreasing function of n (number of hops in a route), and is monotonically increasing function of m_i (number of symbols in parity code). This is observed in Fig. 2. Fig. 2 (a) shows the surface of $F_\alpha(n, m_i)$ when $\alpha = 0.5$, $z_i = 20$ and $\hat{\epsilon} = 1.5 \times 10^{-3}$, when n is varied from 5 to 50, and m_i is varied from 12 to 30. Fig. 2 (b) shows the simulation results of the successful decoding rate under the FSMC model between a source and a destination node with n hops between them (n is ranged from 5 to 50). The consistency between the analysis results and simulation results verifies Equ. (4.2) and Equ. (4.3).

Based on Equ. (4.3), the number of times (i.e., trials) a packet is required to be decoded until it is recovered has a nonhomogeneous geometric distribution (denoted by G) [47] given that the length (i.e., number of symbols) of predetermined parity code equals m_t at the t^{th} trial.

Lemma 4.1.1 *I use $f_G^t(n, m_{i_t})$ to denote the probability of successful decoding on the t^{th} decoding*

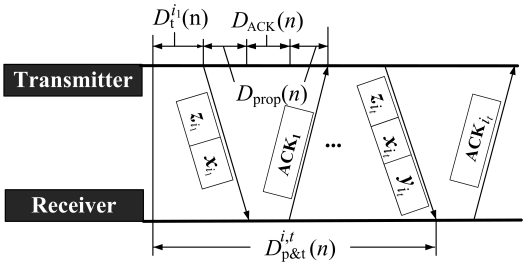


Figure 4.3: Transmission and propagation delay.

trial for a packet M_i going through n hops. Then,

$$f_G^t\left(n, m_{i_t}\right) = F_\alpha\left(n, m_{i_t}\right) \times \prod_{j=1}^{t-1}\left(1 - F_\alpha\left(n, m_{i_j}\right)\right) \tag{4.4}$$

4.1.1 Propagation and Transmission Delay

In this section, we consider the prop&tran delay of each packet M_i. We use $D_{\text{p\&t}}^{i,t}(n)$ to denote the prop&tran delay of M_i when the parity code of M_i has been transmitted for t times through n nodes. Let $D_{\text{p}}(n)$ represent the propagation delay for one packet going through n nodes and $D_{\text{ACK}}(n)$ denote the transmission delay of the ACK packet. Further, let $D_{\text{t}}^{i_k}(n)$ denote the transmission delay of the packet M_{i_k}. The length of this packet is $L_{\text{pac}}^{i_k} = z_{i_k} + m_{i_k}$, where M_{i_k} is the k^{th} packet that carries M_i's parity symbols for the k^{th} time after k-1 times of recovery failures (i.e., type-II parities). Then, as Fig. 4.3 shows, $D_{\text{p\&t}}^{i,t}(n)$ can be calculated as

$$D_{\text{p\&t}}^{i,t}(n) = \sum_{k=0}^{t-1}\left(2D_{\text{p}}(n) + D_{\text{ACK}}(n) + D_{\text{t}}^{i_k}(n)\right) + D_{\text{p}}(n) + D_{\text{t}}^{i_t}(n) \tag{4.5}$$

We use $R_{i,l}$ to denote the bandwidth provided to the route M_i travels in the l^{th} hop. Assume electric signal travels at velocity c in the media and the distance of each hop (d) is an invariable. Then, $D_{\text{ACK}}(n)$, $D_{\text{t}}^{i_k}(n)$ and $D_{\text{p}}(n)$ can be calculated as:

$$D_{\text{ACK}}(n) = \frac{nL_{\text{ACK}}}{R_{i,l}}, \tag{4.6}$$

$$D_{\text{t}}^{i_k}(n) = \frac{nL_{\text{pac}}^{i_l}}{R_{i,l}}, \tag{4.7}$$

$$D_{\text{p}}(n) = \frac{nd}{c} \tag{4.8}$$

Based on Equ. (4.5) and (4.6), we can derive that:

$$D_{\text{p\&t}}^{i,t}(n) = \sum_{l=1}^{n} \left[\sum_{k=0}^{t-1} \left(\frac{2d}{c} + \frac{L_{\text{ACK}} + L_{\text{pac}}^{i_k}}{R_{i,l}} \right) + \frac{d}{c} + \frac{L_{\text{pac}}^{t}}{R_{i,l}} \right] \tag{4.9}$$

Based on Equ. (4.4) in Lemma 4.1.1 and Equ. (4.9), we retrieve Lemma 4.1.2 for the expectation of $D_{\text{p\&t}}^{i}(n)$.

Lemma 4.1.2 *Assuming each packet has the same length, the expected propagation and transmission delay of a packet M_i $D_{\text{p\&t}}^{i}(n)$ can be calculated by:*

$$\overline{D_{\text{p\&t}}(n)} = \sum_{t=1}^{\infty} f_G^t(n, m_{i_t}) D_{\text{p\&t}}^{i,t}(n) \tag{4.10}$$

As shown in Fig. 4.5, given a route from a source node to a destination node, we can divide the route into e segments, each segment having length of $n_1, n_2, ..., n_e$. In each route segment, a packet is encoded at the first node and decoded at the last node. The goal of our scheme is to determine the $n_1, n_2, ..., n_e$ in order to minimize the prop&tran delay of a packet from the source to the destination, i.e., to achieve

$$\min \quad \sum_{j=1}^{e} \overline{D_{\text{p\&t}}(n_j)} \tag{4.11}$$

$$\text{s.t.} \quad \sum_{j=1}^{e} n_j = n \tag{4.12}$$

Note that the above formulated optimization problem still holds with routing delay constraint, since minimizing $\overline{D_{\text{p\&t}}(n_j)}$ is sufficient for satisfying the delay constraint. That is, if the minimum value of $\overline{D_{\text{p\&t}}(n_j)}$ is no larger than the constraint, then the constraint is always satisfied; otherwise, there is no solution to satisfy the constraint.

Error Estimation Code. Recall that in the above method, a receiver requests its sender to send the packet repetitively until it can successfully decode the packet, which may lead to multiple retransmissions for each packet. In order to avoid such retransmissions, we introduce another method that only needs one retransmission.

In this method, after receiving a packet, each receiver first uses Error Estimation Code (EEC) [9] to estimate the number of corrupted symbols in the received packet, and then sends a

request to its sender to ask for the additional parity code, which helps successfully recover the packet.

EEC estimates BER (e.g., checks whether BER is no larger than 1%) of the received packet, but in CEDAR, the receiver needs to estimate the number of corrupted symbols. Then, CEDAR uniformly samples the symbols instead of bits and builds EEC for the sampled symbols. More specifically, for a packet with length m_i, there are $\lfloor \log_2 m_i \rfloor$ levels of EEC bits added in each packet, with s EEC bits in each level. An EEC bit at level $i (1 \leq i \leq \lfloor \log_2 m_i \rfloor)$ is simply the parity bit for $2^i - 1$ randomly chosen symbols in the packet, which has totally $(2^i - 1)b$ bits. Each of these $2^i - 1$ data symbols is chosen uniformly randomly and independently (with replacement) from the original m_i symbols.

In addition to the parity code, each packet M_i also contains EEC codes, which has a length of $\log_2 \lfloor m_i \rfloor s/b$. Note that though this EEC-based method reduces the number of retransmission, it increases the transmission packet size. First we consider an ideal scenario, in which EEC never underestimates the number of corrupted symbols for each packet. Then, the prop&tran delay of a packet M_i can be simply calculated by:

$$D_{\text{p\&t}}^{\text{id}}(n) = D_{\text{p\&t}}^{i,1}(n) + D_{\text{p\&t}}^{i,2}(n). \tag{4.13}$$

However, like any error estimator, EEC may underestimate the number of errors. We use η to denote the probability of underestimation, then the actual expected prop&tran delay of a packet M_i is given by:

$$\overline{D_{\text{p\&t}}(n)} = (1-\eta) D_{\text{p\&t}}^{\text{id}}(n) + \eta \sum_{t=3}^{\infty} h_G^{3,t}\left(n, m'_{i_t}\right) \sum_{l=1}^{t} D_{\text{p\&t}}^{i,l}(n) \tag{4.14}$$

where $h_G^{3,t}\left(n, m'_{i_t}\right) = F_\alpha\left(n, m'_{i_3}\right) \prod_{j=3}^{t-1} \left(1 - F_\alpha\left(n, m'_{i_j}\right)\right)$ and $m'_{i_j} = m_{i_j} + \log_2 \lfloor m_{i_j} \rfloor s/b$.

4.2 Queuing Delay

In a priority queuing model, packets entering a buffer are classified into several different priority categories and added into different queues accordingly. The packets with lower priority can enter the server only when all queues for higher priority queues are empty. In the wireless network, for any single node v_i that is responsible for decoding at most K routes, there will be K poisson streams $(\lambda_{i,1}, \lambda_{i,2}, ..., \lambda_{i,K})$ arriving at this node. Notice if v_i is not responsible for decoding packet for $\mathbf{r}_k$, $\lambda_{i,k} = 0$. v_i needs to decide the order of arriving packets to decode. Thus, by regarding

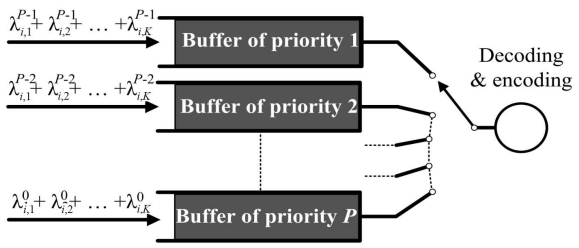


Figure 4.4: The structure of priority queue model.

v_i as the server in the model, we can use the priority queuing model (M/M/1/∞/∞/PR) [23] for analyzing the queuing delay. Note when a packet fails to decode, it will be decoded (i.e., join in a queue) again when it received another type-II parity code along with another packet. In order to balance the queuing delay of each node, we propose a strategy for determining the priority of decoding packets. That is, the more times a packet has failed to be corrected, the higher priority it will be given when it is re-decoded. When a packet suffers P number of failures, it is dropped. We do not consider the stream of retransmission for packets after P failures because the probability of failing more than P times is extremely small.

Poisson process is widely used to describe the data traffic in wireless networks [5, 23, 34, 49], so we also use Poisson process to model the data traffic in this paper. The self-similar model has been proven to be more realistic than Poisson process to describe data traffic in modern LANs and WANs, in which batch arrivals, event correlations and traffic burstiness are key factors [2]. To the best of our knowledge, there is no previous work that has studied the priority queuing system based on the self-similar model. We will use the self-similar model to analyze the packet delay in our future work.

We use priority queuing model to analyze the queuing delay for the packets crossing a given node. Fig. 4.4 gives a sketch of the priority queuing model in our scheme. In the figure, $\lambda_{i,1}^{p}, \lambda_{i,2}^{p}, ..., \lambda_{i,K}^{p}$ denote the arriving rate of the streams whose packets are re-decoded at the $(p-1)^{\text{th}}$ time. Recall that if a packet fails to decode, it is stored in the buffer waiting for the next parity symbol for recovery. As indicated in [8], if traffic stream A follows Poisson distribution, and each packet in A with some probability gets selected to generate a new traffic stream B, then packet stream B will also follow Poisson distribution. Therefore, the re-decoded streams, which are "generated" by failed decoded packets, follow Poisson distribution and their arrival rates satisfy the following

condition:

$$\lambda_{i,k}^{p}\left(n_{i,k}\right)=\lambda_{i,k}^{1}\prod_{t=1}^{p-1}\left(1-F_{\alpha}\left(n_{i,k},m_{t}\right)\right) \tag{4.15}$$

where $n_{i,k}$ $(1\leq k\leq U)$ denotes the number of hops in the route segment where k^{th} traffic stream has traveled through since its last en/decoding in the route. Assuming that $\lambda_{i,k}^{1}$ and m_t have been pre-determined, the value of $\lambda_{i,k}^{p}\left(n_{i,k}\right)$ is determined by $n_{i,k}$. We assign priority p to the packet stream of $\lambda_{i,k}^{p}$ $(p=1,2,...,P)$. The packets in a queue with the highest priority P enter the server (decoding and encoding part) first. If the queue of priority P is empty, then the packets of priority $P-1$ enter the server; and so on.

Assume there are K data streams in the p^{th} $(1\leq p\leq P)$ priority queue, because each packet stream follows Poisson distribution, all of these streams can be combined into one stream $\lambda_{i}^{p}=\sum_{k=1}^{K}\lambda_{i,k}^{p}$. We use ρ_l to represent the *utilization* of a server when the first packet in the buffer with priority l enters the server and use Y_l to represent service time for a packet in a queue with priority l [23]. Recall $\mathbf{n}_i$ is the set of all $n_{i,k}$ $(1\leq k\leq K)$. Then, $\rho_l\left(\mathbf{n}_i\right)$ can be calculated as

$$\rho_{l}\left(\mathbf{n}_{i}\right)=\overline{Y_{l}}\times\sum_{k=1}^{K}\lambda_{i,k}^{l}\left(n_{i,k}\right) \tag{4.16}$$

W_l represents the average delay of packets with priority l packets and W_0 represents the average delay for one tagged waiting packet due to a packet already in service. W_0 can be calculated as:

$$W_{0}\left(\mathbf{n}_{i}\right)=\sum_{l=1}^{P}\frac{\overline{Y_{l}^{2}}}{2\overline{Y_{l}}}\times\rho_{l}\left(\mathbf{n}_{i}\right) \tag{4.17}$$

As a result, the waiting time for each of packets is:

$$W_{p}\left(\mathbf{n}_{i}\right)=\frac{W_{0}\left(\mathbf{n}_{i}\right)+\sum_{l=p+1}^{P}\rho_{l}\left(\mathbf{n}_{i}\right)W_{l}\left(\mathbf{n}_{i}\right)}{1-\sum_{l=p}^{P}\rho_{l}\left(\mathbf{n}_{i}\right)} \tag{4.18}$$

From Kleinrock's conservation theorem in priority queuing model [27], the expected queuing delay for one packet in any node can be calculated as:

$$\overline{W_{\text{que}}\left(\mathbf{n}_{i}\right)} = \sum_{p=1}^{P}\rho_{p}\left(\mathbf{n}_{i}\right)W_{p}\left(\mathbf{n}_{i}\right) \tag{4.19}$$

$$= \frac{\rho\left(\mathbf{n}_{i}\right)W_{0}\left(\mathbf{n}_{i}\right)}{1-\rho\left(\mathbf{n}_{i}\right)} \tag{4.20}$$

where

$$\rho\left(\mathbf{n}_i\right) = \sum_{p=1}^{P} \rho_p\left(\mathbf{n}_i\right) \tag{4.21}$$

Now, we consider the queueing delay for one packet, which might enter the queueing system several times due to re-en/decoding. During time interval T (T is large enough), the total number of packets N_{total} that enter the queueing system equals:

$$N_{\text{total}}\left(\mathbf{n}_i\right) = \sum_{l=1}^{P} \sum_{j=1}^{K} \lambda_{i,j}^{l}\left(n_{i,j}\right) \times T \tag{4.22}$$

The total waiting time can be given by:

$$W_{\text{total}}\left(\mathbf{n}_i\right) = N_{\text{total}}\left(\mathbf{n}_i\right) \times \overline{W_{\text{que}}\left(\mathbf{n}_i\right)} \tag{4.23}$$

Lemma 4.2.1 *The expectation of the total queuing time for one packet when it goes through a node with* $\mathbf{n}_i = [n_{i,1}, n_{i,2}, n_{i,3}, ..., n_{i,K}]$ *can be calculated as*

$$\overline{D_{\text{q}}\left(\mathbf{n}_i\right)} = \frac{W_{\text{total}}\left(\mathbf{n}_i\right)}{\sum_{k=1}^{K} \lambda_{i,k}^{1}\left(n_{i,k}\right)} \tag{4.24}$$

$$= \frac{\sum_{p=1}^{P} \sum_{k=1}^{K} \lambda_{i,k}^{p}\left(n_{i,k}\right) \times \overline{W_{\text{que}}\left(\mathbf{n}_i\right)}}{\sum_{k=1}^{K} \lambda_{i,k}^{1}} \tag{4.25}$$

If each receiver uses EEC to estimate the number of corrupted symbols, then the packet only needs to be retransmitted at most twice, i.e., $P = 2$. Hence, the expectation of the total queuing time equals:

$$\overline{D_{\text{q}}\left(\mathbf{n}_i\right)} = \frac{\sum_{p=1}^{2} \sum_{k=1}^{K} \lambda_{i,k}^{p}\left(n_{i,k}\right) \times \overline{W_{\text{que}}\left(\mathbf{n}_i\right)}}{\sum_{k=1}^{K} \lambda_{i,k}^{1}} \tag{4.26}$$

The above priority queue model assumes that packets have no delay constraints, so it gives a higher priority to a packet that has been retransmitted more times. With the consideration of packet delay constraints (deadlines), we can first give a higher priority to the packet with smaller remaining time period to its deadline; if two packets have the same remaining time periods, we give a higher priority to the packet that has been transmitted more times. Modeling such two-level priority queue to calculate queuing delay is non-trivial, so we leave this task as our future work.

4.3 Minimizing the Delays

We need to minimize $\overline{D_{\mathrm{q}}(\mathbf{n}_i)}$ and $\overline{D_{\mathrm{p\&t}}(\mathbf{n}_i)}$ in order to achieve the objective of CEDAR in Formula (3.16). According to Equ. (4.10), the prop&tran delay of the packet stream for $\mathbf{r}_k$ in v_i is calculated as:

$$\overline{D_{\mathrm{p\&t}}(n_{i,k})} = \sum_{t=1}^{\infty} f_G^t(n_{i,k}) D_{\mathrm{p\&t}}^t(n_{i,k}) \tag{4.27}$$

Consequently, the average prop&tran delay of all the packet stream decoded in v_i is calculated as:

$$\overline{D_{\mathrm{p\&t}}(\mathbf{n}_i)} = \frac{\sum_{k=1}^{K}\left[\lambda_{i,k}\sum_{t=1}^{\infty} f_G^t D_{\mathrm{p\&t}}^t(n_{i,k})\right]}{\sum_{k=1}^{K}\lambda_{i,k}} \tag{4.28}$$

where $\mathbf{n}_i = [n_{i,1}\ n_{i,2}\ ...\ n_{i,K}]$. The average queuing delay of the packet stream in v_i can be derived from Equ. (4.24):

$$\overline{D_{\mathrm{q}}(\mathbf{n}_i)} = \frac{\sum_{p=1}^{P}\sum_{k=1}^{K}\lambda_{i,k}^p(n_{i,k}) \times \overline{W_{\mathrm{que}}(\mathbf{n}_i)}}{\sum_{k=1}^{K}\lambda_{i,k}^1(n_{i,k})} \tag{4.29}$$

By minimizing the above $\overline{D_{\mathrm{q}}(\mathbf{n}_i)}$ and $\overline{D_{\mathrm{p\&t}}(\mathbf{n}_i)}$, we retrieve two propositions presented below.

1) Minimizing Queuing Delay

Proposition 4.3.1 *Suppose* $\mathcal{V} = \{v_1, v_2, ..., v_N\}$ *and* $\mathcal{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_K\}$ *with total arrival rate* $\sum_{k=1}^{K}\lambda_k = \lambda_{\mathrm{total}}$*, each packet is required to be decoded once in its route, and each node can decode the packet in any route, and also* $\mathbf{n}_1 = \mathbf{n}_2 = ... = \mathbf{n}_N$*. To minimize the total queueing delay for all the packets, the packet rate each node is responsible for should be the same. That is:*

$$\overline{\lambda_i} = \frac{\lambda_{\mathrm{total}}}{N} \quad (i = 1, 2, 3, ..., N) \tag{4.30}$$

Proof According to Equ. (4.16), Equ. (4.17), Equ. (4.19) and (4.21), the queuing delay of packets decoded at v_i can be derived as:

$$\overline{D_{\mathrm{q}}^i(\mathbf{n}_i)} = \frac{\rho(\mathbf{n}_i)}{1-\rho(\mathbf{n}_i)} \frac{\sum_{l=1}^{P}\sum_{k=1}^{B}\lambda_k^l(n_{i,k})}{\sum_{k=l}^{B}\lambda_k^1} W_0 \tag{4.31}$$

$$= \frac{AJ\overline{\lambda_i}^2}{1-J\overline{\lambda_i}} \tag{4.32}$$

where A and J are invariants figured by Equ. (4.21), Equ. (4.16) and Equ. (4.15). Then, the following

equation can be derived:

$$\sum_{i=1}^{N} \frac{AJ\overline{\lambda_i}^2}{1-J\overline{\lambda_i}} = -AC\lambda_{\text{total}} + \frac{AN}{J} + \frac{A}{J}\sum_{i=1}^{N} \frac{1}{\left(1-J\overline{\lambda_i}\right)} \tag{4.33}$$

Let $H_i = 1 - J\overline{\lambda_i}$ and then $\sum_{i=1}^{N} H_i = N - J\lambda_{\text{total}}$. According to Cauchy−Schwarz inequality [8], we find that:

$$\begin{aligned}
\sum_{i=1}^{N} \frac{N - J\lambda_{\text{total}}}{H_i} &= \sum_{i=1}^{N} \frac{1}{H_i} \sum_{i=1}^{N} H_i && (4.34)\\
&= \sum_{i=1}^{N} \frac{1}{\left(\sqrt{H_i}\right)^2} \sum_{i=1}^{N} \left(\sqrt{H_i}\right)^2 && (4.35)\\
&\geq \left(\sum_{i=1}^{N} \sqrt{\frac{1}{H_i}}\sqrt{H_i}\right)^2 && (4.36)\\
&= N^2 && (4.37)
\end{aligned}$$

from which, we can derive that

$$\begin{aligned}
\sum_{i=1}^{N} \frac{1}{\left(1-J\overline{\lambda_i}\right)} &= \sum_{i=1}^{N} \frac{1}{H_i} && (4.38)\\
&\geq \frac{N^2}{N - J\lambda_{\text{total}}} && (4.39)
\end{aligned}$$

and

$$\sum_{i=1}^{N} \frac{AJ\overline{\lambda_i}^2}{1-J\overline{\lambda_i}} \geq -A\lambda_{\text{total}} + \frac{AN}{J} + \frac{AN^2}{J\left(N - \lambda_{\text{total}}\right)} \tag{4.40}$$

which reaches its minimum value when $1 - J\overline{\lambda_1} = 1 - J\overline{\lambda_2} = ... = 1 - J\overline{\lambda_N}$, or the values of $\overline{\lambda_i}$ $(i = 1, 2, .., N)$ are equal to each other.

According to Proposition 4.3.1, given several packet streams and number of nodes required to decode these packets, we need to balance the en/decoding load for all of these nodes. For example, in Fig. 4.6, three packet streams λ_1, λ_2 and λ_3 are transmitted through the 1st route, the 2nd route and the 3rd route, respectively. The 2nd route chose v_2 as the key node, and the 3rd route chose v_4 as the key node. For the 1st route, v_1 is the packet source (*sender*) and v_5 is the packet destination (*receiver*). It needs to choose one node among intermediate nodes v_2, v_3 and v_4 to decode packets.

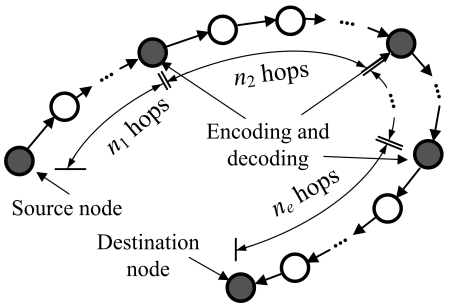


Figure 4.5: Route segment.

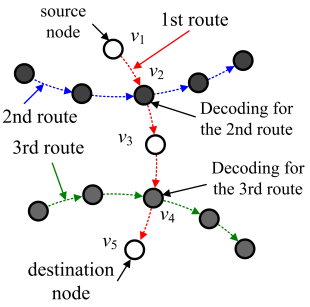


Figure 4.6: Example of balancing en/decoding load.

To achieve load balance and hence reduce queuing delay, its best choice should be v_3 since v_2 and v_4 are already responsible for en/decoding for the 2^{nd} and 3^{rd} routes, respectively.

2) Minimizing Prop&Tran Delay

Consider a route with n hops that is not affected by the interference from any other routes. Our objective is to divide it into several route segments with the size n_1, n_2, ..., n_e respectively in order to minimize the total delay of packet stream transmission. We consider one of these route segment that has n_k hops, and use $\overline{D_{\text{p\&t}}^{\text{ave}}}$ to denote the average prop&tran delay for each hop in this route segment. That is:

$$\overline{D_{\text{p\&t}}^{\text{ave}}(n_i)} = \frac{\overline{D_{\text{p\&t}}(n_i)}}{n_i} \ (i = 1, 2, 3, ..., e) \tag{4.41}$$

where $\overline{D_{\text{p\&t}}^{\text{ave}}(n_i)}$ is a function of n_i. We use $\overline{D_{\text{p\&t,min}}^{\text{ave}}}$ to represent the minimized value of $\overline{D_{\text{p\&t}}^{\text{ave}}(n_i)}$, and we need to search n_{opt} that satisfies $\overline{D_{\text{p\&t}}^{\text{ave}}(n_{\text{opt}})} = \overline{D_{\text{p\&t,min}}^{\text{ave}}}$,

Proposition 4.3.2 *To divide one route into several route segments, the optimal length for each route segment should be n_{opt} in order to minimize the prop&tran delay for the packet delivery.*

Proof The sum of prop&tran delay for the e route segments equals:

$$\sum_{i=1}^{e} \overline{D_{\text{p\&t}}(n_i)} = \sum_{i=1}^{e} \frac{\overline{D_{\text{p\&t}}(n_i)}}{n_i} \times n_i \tag{4.42}$$

$$\geq \overline{D_{\text{p\&t,min}}^{\text{ave}}} \times \sum_{i=1}^{e} n_i \tag{4.43}$$

$$= \overline{D_{\text{p\&t,min}}^{\text{ave}}} \times n \tag{4.44}$$

When $n_1 = ... = n_e = n_{\text{opt}}$, $\sum_{i=1}^{e} \overline{D_{\text{p\&t}}(n_i)}$ reaches its minimum value.

4.4 Balancing of En/decoding Load

In some wireless network applications (e.g., wireless sensor networks), balanced en/decoding load distribution among all nodes greatly affects the network performance including the connectivity and lifetime of the network. Thus, besides minimizing the total packet delay of the system, we have the secondary objective, which is to minimize the difference of the en/decoding load of all the nodes. Recall that "en/decoding load" of a node, say v_i, is defined as the number of packets v_i need to en/decode in a time unit. According to the queuing model built in the Section 4.2, v_i needs to en/decode all the packets waiting the priority queue from different routes, we can derive the value of $\overline{L(\mathbf{n}_i)}$ from Equ. (4.15):

$$\overline{L(\mathbf{n}_i)} = \sum_{k=1}^{K}\sum_{p=1}^{P}\lambda_{i,k}^{p}\left(n_{i,k}\right) \tag{4.45}$$

$$= \sum_{k=1}^{K}\sum_{p=1}^{P}\lambda_{i,k}^{1}\prod_{t=1}^{p-1}\left(1-F_{\alpha}\left(n_{i,k},m_t\right)\right) \tag{4.46}$$

We use $\overline{L\left(\mathbf{N}\right)} = \frac{\sum_{i=1}^{N}\overline{L(\mathbf{n}_i)}}{N}$ to represent the average en/decoding load of all the nodes in $\mathcal{V}$. Since standard deviation [8] of en/decoding load (represented by $\sigma(\mathbf{N})$) shows how much variation or "dispersion" exists from the average $\overline{L\left(\mathbf{N}\right)}$, we use $\sigma(\mathbf{N})$ to measure the fairness of en/decoding load of all the nodes: the lower value of $\sigma(\mathbf{N})$, the higher fairness of the en/decoding load of all the nodes. $\sigma(\mathbf{N})$ is defined by

$$\sigma(\mathbf{N}) = \sqrt{\sum_{i=1}^{N}\left(\overline{L(\mathbf{n}_i)} - \overline{L(\mathbf{N})}\right)^2} \tag{4.47}$$

Then, the objective is to minimize $\sigma(\mathbf{N})$.

Proposition 4.4.1 *Suppose there are N nodes and B routes with total arrival rate λ_{total} and each packet is required to be decoded once in its route, and each node can decode the packet in any route with $\mathbf{n}_1 = \mathbf{n}_2 = \mathbf{n}_3 = ... = \mathbf{n}_N$. To minimize $\sigma(\mathbf{N})$, the en/decoding load of each node should be the same.*

Proof *When the packet arrival rate that each node v_i is responsible for is $\overline{\lambda_i} = \frac{\lambda_{\text{total}}}{N}$, according to Equ. (4.45), we can derive that $\overline{L(\mathbf{n}_1)} = ... = \overline{L(\mathbf{n}_N)} = \overline{L(\mathbf{N})}$. which indicates that $\sigma(\mathbf{N}) = 0$. Because $\sigma(\mathbf{N}) \geq 0$, we can conclude that when the en/decoding load of each node is same, the value of $\sigma(\mathbf{N})$ is minimized.*

Proposition 4.3.1 and Proposition 4.4.1 imply that minimizing the queuing delay and balancing en/decoding load distribution among decoding nodes in the network can be achieved simultaneously.

4.5 Scalable and Distributed Scheme

The objective function of CEDAR in Formula (3.16) is a non-linear integer programming problem. Solving this problem leads to minimizing the total delay for all the packets in the network. This, however, requires each node to collect a global knowledge of the network including the routes and the arrival rate for each traffic stream, which is nearly impractical in wireless applications such as wireless ad hoc networks. Even though the global knowledge is available, the problem is NP-hard as it is a nonlinear integer programming problem [58]. Thus, we need to design a scalable and distributed scheme for identifying the key nodes for each route.

Simply put, Proposition 4.3.1 and Proposition 4.4.1 indicate that the scheme should try to balance the en/decoding load of each node to minimize the queuing delay; and Proposition 4.3.2 indicates the optimal route segment length (i.e., the positions of key nodes) to minimize the prop&tran delay. If both requirements can be satisfied simultaneously, the scheme will satisfy the objective function. However, these two requirements may conflict with each other. We identify different network traffic load situations that each proposition should be primarily considered, and also propose a method to coordinately consider these two propositions when choosing key nodes.

Case I (light traffic): When a wireless network has light traffic, because the influence from queuing delay is much less significant, and on average en/decoding load of each node is low, we mainly consider the prop&tran delay. As Proposition 4.3.2 indicates, we first search the value of n_{opt} and then set OPT_HOP = n_{opt}, and set FLAG = OPT_HOP. In a routing algorithm [51], every node keeps a routing table, and a source node sends out a message to find the route to a destination for transmitting a packet stream. After a source-destination route has been discovered, each node in the route determines whether it is a key node in a distributed manner by executing the key node identification algorithm. Algorithm 1 presents the pseudo code of this algorithm executed by every node (except source node and destination node), say v, in a route $\mathbf{r}$ in the case of light network traffic. Basically, the source node sends a packet (OPT_HOP, FLAG) through the route and nodes. FLAG is decreased by one in each hop and the node receiving the packet with FLAG = 0 is a key

node. This node then restores FLAG = OPT_HOP before forwarding the packet to the next node. Here, SEN_FIN presents whether v has received ACK from the next node in $\mathbf{r}$; REC_FIN presents whether v has received (OPT_HOP, FLAG) from the previous node in $\mathbf{r}$; DEC presents whether v is responsible for en/decoding;

Algorithm 1: Identify key nodes in route $\mathbf{r}$ executed by each node in $\mathbf{r}$ in a light-traffic network.

```
begin
    Set SEN_FIN, REC_FIN and DEC to 0 ;
    while SEN_FIN = 0 or REC_FIN = 0 do
        Listen to other nodes;
        if it has received ACK_REC from the next node in r then
            SEN_FIN ← 1;
        if it receives (OPT_HOP, FLAG) from the previous node in r then
            REC_FIN ← 1;
            Send ACK_REC to the previous node;
            if FLAG = 0 then
                DEC ← 1 // It is a key node;
                FLAG ← OPT_HOP;
            else
                FLAG ← FLAG - 1;
```

Case II (heavy traffic): When a wireless network has heavy traffic, we aim to reduce queuing delay while reducing the prop&tran delay. Also, we need to decrease the difference of en/decoding load overall the network. Fortunately, according to Proposition 4.3.1 and Proposition 4.4.1, decreasing queuing delay and increasing load balancing of the network can be achieved simultaneously.

When a new route is built, the CEDAR scheme first executes Algorithm 1. When executing the algorithm, each node along the route piggybacks its en/decoding load to the packet, and the last node sends the collected en/decoding load information to the source node. The source node then knows the series of nodes identified as "potential key nodes" and their en/decoding load, and calculates the average en/decoding load through the route, denoted as AVE_LOAD. It then checks whether the en/decoding load of each identified key node is larger than a pre-defined threshold (AVE_LOAD + BOUND), where BOUND is a predetermined value. An overloaded potential key node is replaced by the node closest to itself in the route that has load below the threshold. We use LOAD_i to denote the en/decoding load of i^{th} node in the route. If $\text{LOAD}_i > (\text{AVE\_LOAD} + \text{BOUND})$, then the source node compares LOAD_{i-1} and LOAD_{i+1}; and chooses the node with smaller en/decoding

load if the en/decoding load is smaller than the threshold. If both LOAD_{i-1} and LOAD_{i+1} are larger than the threshold, the source node compares LOAD_{i-2} and LOAD_{i+2}. The source node repeats this process until finding a node with load below the threshold or the next two nodes needs to be compared are out of the range of $[i - \lfloor \text{OPT\_HOP}/2 \rfloor, i + \lfloor \text{OPT\_HOP}/2 \rfloor]^{\text{th}}$. The pseudo code of this algorithm is presented in Algorithm 2. In order to release the key node selection load on the source node, this algorithm can be easily extended to a fully decentralized algorithm. Specifically, the destination node calculates the AVE_LOAD and forwards it back to the source node along the route. Each node checks whether its own load is beyond the threshold. If so, it probes its near-by nodes sequentially until finding a node with load within the threshold or meeting an identified potential key node.

Algorithm 2: Select key nodes with consideration of load balance in a heavy-traffic or normal-traffic network.

```
begin
    Use Algorithm 1 to get the "potential key nodes" in route r;
    Let node i be one node selected as the "potential key node";
    j = 0;
    while j ≤ ⌊OPT_HOP/2⌋ do
        if LOAD_{i+j} ≤ LOAD_{i−j} then
            if LOAD_{i+j} ≤ (BOUND + AVE_LOAD) then
                return i + j;
        else
            if LOAD_{i−j} ≤ (BOUND + AVE_LOAD) then
                return i − j;
        j = j + 1;
    return 0;
```

In dynamic wireless networks, the network topology and the packet arrival rates change over time, which require nodes to recalculate the key nodes periodically. Based on Algorithm 1 and Algorithm 2, we see that such dynamics only affect the route length and the en/decoding load of each node in the route that are needed in key node calculation. Thus, to deal with the dynamics, we let the sender periodically send a packet through the route to probe this information. As a result, the key nodes calculated by a packet sender are the correct key nodes for the current network environment and CEDAR is adaptive to the network dynamics.

Chapter 5

Link Scheduling

In this chapter, we will introduce how to solve the different link scheduling problems introduced in Section 3.2.2. In particular, in Section 5.1, we propose two centralized algorithms (the Link Diversity Partition method and the Recursive Link Elimination method) and one decentralized algorithm (the Decentralized Link Scheduling method) for the fading-resistant link scheduling problem. In Section 5.2, we propose two Link Length Diversity based algorithms for CLS and O-CLS, respectively. In Section 5.3, we propose a decentralized algorithm (the Fast and Lightweight Autonomous link scheduling algorithm) for the cooperative communication link scheduling problem.

5.1 Fading Resistant Link Scheduling

Since *Fading-R-LS* is a NP-hard problem, there are no polynomial time solutions to determine the optimal schedule. To solve this problem, in this section, we propose the Link Diversity Partition algorithm (LDP) (Section 5.1.1). We further propose a constant approximation ratio algorithm, namely Recursive Link Elimination algorithm (RLE) (Section 5.1.2) for the case when the data rate of each link is the same.

5.1.1 Link Diversity Partition Algorithm

The LDP algorithm is developed based on the approximation algorithm using the signal to interference ratio (SIR) model proposed in [18], where the SIR model neglects the noise power compared with the SINR model. As previously explained, the deterministic SIR model does not

consider the fluctuating fading in transmissions, which makes the algorithm susceptible to fading environment. Instead, LDP is advantageous by taking into account Rayleigh fading, which however is a non-trivial task. Below, we first briefly introduce the algorithm in [18], explain the faced challenge and advantages of the LDP design, and then present LDP.

The algorithm in [18] builds disjoint link classes by classifying the links with similar lengths to one class. For each link class, it partitions the entire network region into squares and set neighboring squares to different colors (Figure 5.5(a)). Such color setting makes the transmissions in the same-color squares always have a certain distance between each other. The size of the squares is calculated based on the SIR model to ensure the successful transmissions of a selected link from each same-color square when all these selected links transmit simultaneously. Then, all the selected links from the same-color squares form a feasible schedule. This algorithm selects the schedule with the highest data rate among all the feasible schedules.

To extend this algorithm for the Rayleigh fading model is challenging because calculating the closed form of successful transmission probability in Rayleigh fading model is much more complex than that in the deterministic SIR model, which makes the size of each square in the grid difficult to estimate. Fortunately, in Corollary 3.2.1, we have derived a linear formula (Formula (3.35)) to judge a successful transmission under Rayleigh fading model. Also, this previous algorithm sets both upper and lower bounds for the link length of each class when building link classes. We further improve this algorithm by only upper bounding the link length of each class, since the transmission of a shorter-length link will be successful if the transmission of a longer-length link in the same square area is successful. This improvement enhances the throughput as there are more link candidates possibly with higher data rates for a schedule.

Algorithm 3: Pseudo-code for LDP.

input : $S_1, ..., S_{g(L)}$;
output: $\mathcal{P}_{\text{ldp}}$;
for *each* $k \leftarrow 1$ *to* $g(L)$ **do**
    Partition the region into squares with side-length β_k;
    while *inactive sender exist in* S_k **do**
        Pick up the sender s_i that has the maximum data rate in S_k and add it to $\mathcal{P}^k_{\text{ldp}}$;
        Remove each s_j in the squares surrounding the square that r_i is positioned;
$\mathcal{P}_{\text{ldp}} \leftarrow \arg\max\{U(\mathcal{P}^k_{\text{ldp}})|\mathcal{P}^k_{\text{ldp}}, \forall k\}$;
return $\mathcal{P}$;

Definition 5.1.1 *(*Length diversity [18]*) Length diversity set of a set of links L, denoted by $G(L)$, is defined by*

$$G(L) = \{h | \exists l, l' \in L : \lfloor \log(d(l)/d(l')) \rfloor = h\} \tag{5.1}$$

and the link length diversity is defined by

$$g(L) = |G(L)|. \tag{5.2}$$

$G(L)$ lists the magnitudes of transmission link lengths, and $g(L)$ represent the number of these magnitudes. In real applications, $g(L)$ is usually a small constant [18].

For example, suppose link set $L = \{(s_1, r_1), (s_2, r_3), (s_3, r_3)\}$, with link lengths 1, 2, and 4, respectively. Because $\lfloor \log \frac{d_{s_2,r_2}}{d_{s_1,r_1}} \rfloor = 1$, $\lfloor \log \frac{d_{s_3,r_3}}{d_{s_1,r_1}} \rfloor = 2$, and $\lfloor \log \frac{d_{s_3,r_3}}{d_{s_2,r_2}} \rfloor = 1$, then the length diversity set of L is $G(L) = \{1, 2\}$ and the link length diversity of L is $g(L) = |G(L)| = 2$.

In the following, we introduce LDP in detail: LDP starts by building $g(L)$ disjoint sender classes S_1, ..., $S_{g(L)}$ from S, *s.t.* $S_k = \{s_i \in S | \delta_k \leq d_{s_i,r_i} < \delta_{k+1}\}$, where $\delta_k = 2^{h_k}\delta$ and δ is the length of the shortest link in L. That is, each class includes the senders whose links' lengths are no larger than a specific magnitude. Then, the desired signal power from each sender class has a lower bound. Next, for each sender class S_k, LDP constructs a feasible schedule. The pseudocode of this procedure is shown in Algorithm 4. When scheduling S_k, the entire network region is partitioned into a set of squares $\mathcal{A}^k = \{A^k_{a,b}\}$ $(a, b = 1, 2, ...)$, where (a, b) represents the location of the square in the grid. Each square has side-length $\beta_k = \beta\delta_k$, where

$$\beta = (12\zeta(\alpha - 1)\gamma_{\text{th}}/\gamma_\varepsilon)^{\frac{1}{\alpha}} + 1, \tag{5.3}$$

where $\zeta(\alpha - 1)$ is the Riemann zeta function and $\zeta(\alpha - 1)$ is a constant for $\alpha > 2$. Then, LDP iteratively picks the inactive sender that has the largest data rate and eliminates all the inactive senders in the 9 squares surrounding the selected sender's receiver (see Fig. 5.1). By eliminating these inactive senders, LDP guarantees that the distances between the active senders to each active receiver is lower bounded by β_k, and hence the interference is upper bounded. The above process is repeated until no active sender exists. Because there are $g(L)$ classes we can finally get $g(L)$ feasible schedules: $\mathcal{P}^k_{\text{ldp}}$ $(k = 1, ..., g(L))$. As the objective of this link scheduling problem is to maximize the

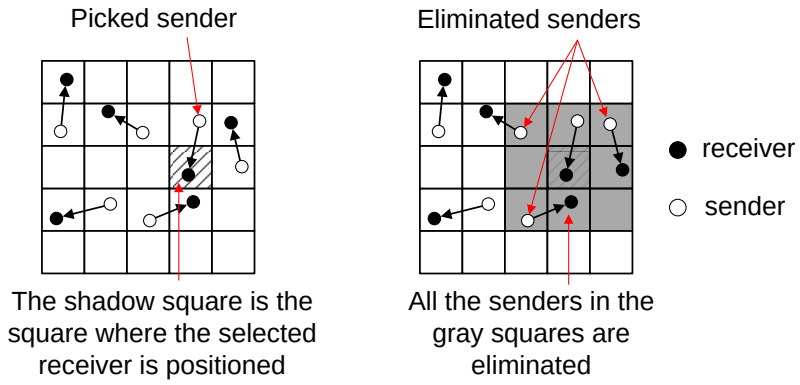


Figure 5.1: Elimination process in each iteration.

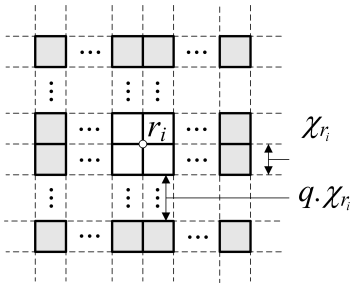


Figure 5.2: Proof of Theorem 5.1.1.

throughput, the schedule with the largest data rate is chosen from these schedules finally:

$$\mathcal{P}_{\text{ldp}} = \arg\max\{U(\mathcal{P}^k_{\text{ldp}})|\mathcal{P}^k_{\text{ldp}}, \forall k\}, \tag{5.4}$$

where $U(\mathcal{P})$ denotes the data rate transmitted by sender set $\mathcal{P}$.

Algorithm 4: Pseudo-code for LDP.

input : $S_1, ..., S_{g(L)}$;
output: $\mathcal{P}_{\text{ldp}}$;
for *each* $k \leftarrow 1$ *to* $g(L)$ **do**
  Partition the region into squares with side-length β_k;
  while *inactive sender exist in* S_k **do**
    Pick up the sender s_i that has the maximum data rate in S_k and add it to $\mathcal{P}^k_{\text{ldp}}$;
    Remove each s_j in the squares surrounding the square that r_i is positioned;
$\mathcal{P}_{\text{ldp}} \leftarrow \arg\max\{U(\mathcal{P}^k_{\text{ldp}})|\mathcal{P}^k_{\text{ldp}}, \forall k\}$;
return $\mathcal{P}$;

Theorem 5.1.1 *LDP provides a feasible schedule.*

Proof Without loss of generality, we examine whether each r_j can successfully receive the packet from s_j ($s_j \in S_k$) if s_j is activated by LDP. We partition the entire network region into squares with side-length $\chi_k = (\beta_k - \delta_k)/\sqrt{2}$, where r_j is positioned at the corner of four squares (see Fig. 5.2). Based on triangular inequality, the distance between any two senders $s_i, s_j \in \mathcal{P}^k_{\text{ldp}}$ cannot be smaller than $\beta_k - \delta_k$, because

$$d_{s_i,s_j} \geq d_{s_i,r_j} - d_{s_j,r_j} \geq d_{s_i,r_j} - \delta_k \geq \beta_k - \delta_k, \tag{5.5}$$

from which we can derive that any two senders in L_k cannot be located in the same square. We use $\mathcal{Q}^k_q$ to denote the set of senders in the squares that is $q\chi_k$ away from r_i. Then, there are at most $4(2q+1)$ senders in $\mathcal{Q}^k_q$. According to the design of LDP, for each $s_i \in \mathcal{P}^k_{\text{ldp}} \backslash s_j$, the distance between s_i and r_j is at least β_k, which is larger than $(\beta_k - \delta_k)/\sqrt{2}$. Therefore, there is no sender in the four squares around r_i, i.e., $\mathcal{Q}^k_0 = \phi$. When $q \geq 1$, the distance between the senders in $\mathcal{Q}^k_q$ and r_j is at least $q\chi_k$, hence the interference factor of any sender $s_i \in \mathcal{Q}^k_q$ on r_j is upper bounded by

$$f_{i,j} = \ln\left(1 + \frac{d^{-\alpha}_{s_i,r_j}\gamma_{\text{th}}}{d^{-\alpha}_{s_j,r_j}}\right) \leq \frac{d^{-\alpha}_{s_i,r_j}\gamma_{\text{th}}}{d^{-\alpha}_{s_j,r_j}} \leq \frac{(q\chi_k)^{-\alpha}\gamma_{\text{th}}}{\delta_k^{-\alpha}}, \tag{5.6}$$

and then the interference factor of $\mathcal{Q}^k_q$ on r_j is upper bounded by

$$f_{\mathcal{Q}^k_q,r_j} = \sum_{s_i \in \mathcal{Q}^k_q} f_{i,j} \leq \frac{4(2q+1)(q\chi_k)^{-\alpha}\gamma_{\text{th}}}{\delta_k^{-\alpha}}. \tag{5.7}$$

Consequently, the interference factor of all senders in $\mathcal{P}\backslash s_j$ on r_j is upper bounded by

$$\begin{aligned}
f_{\mathcal{P}\backslash s_j} &= \sum_{q=0}^{\infty} f_{\mathcal{Q}^k_q,r_j} \leq \sum_{q=1}^{\infty} \frac{4(2q+1)q^{-\alpha}\chi_k^{-\alpha}\gamma_{\text{th}}}{\delta_k^{-\alpha}} && (5.8)\\
&\leq \sum_{q=1}^{\infty} \frac{12q \cdot q^{-\alpha}\chi_k^{-\alpha}\gamma_{\text{th}}}{\delta_k^{-\alpha}} \leq \frac{12\chi_k^{-\alpha}\gamma_{\text{th}}}{\delta_k^{-\alpha}}\zeta(\alpha-1) && (5.9)\\
&= 12(\beta_k/\delta_k - 1)^{-\alpha}\gamma_{\text{th}}\zeta(\alpha-1) && (5.10)\\
&= 12(\beta-1)^{-\alpha}\gamma_{\text{th}}\zeta(\alpha-1) && (5.11)\\
&= \gamma_\varepsilon && (5.12)
\end{aligned}$$

which implies that r_j can be informed.

In the following, we analyze the efficiency of LDP (Theorem 5.1.2). We start by defining $\mathcal{P}_{\text{opt}}$ to be the optimum schedule and defining $\mathcal{P}_{\text{opt}}^k$ to be the optimum schedule comprised by the senders in S_k. Then we have

$$U(\mathcal{P}_{\text{opt}}) = \sum_{k=1}^{g(L)} U(\mathcal{P}_{\text{opt}}^k). \tag{5.13}$$

In addition, we define $B_{\text{opt}}^k(a,b)$ as the subset of $\mathcal{P}_{\text{opt}}^k$ with the receivers located in $A_{a,b}^k$ and define $s_{\text{opt}}^k(a,b)$ as the sender that has the maximum data rate in $B_{\text{opt}}^k(a,b)$:

$$s_{\text{opt}}^k(a,b) = \arg \max_{s \in B_{\text{opt}}^k(a,b)} U(s) \tag{5.14}$$

Finally, we represent the active sender selected by LDP in $A_{a,b}^k$ by $s_{\text{ldp}}^k(a,b)$ and denote the sender that has the maximum data rate in the nine squares around $A_{a,b}^k$ (also including $A_{a,b}^k$) by

$$\hat{s}_{\text{ldp}}^k(a,b) = \arg \max_{c,d \in \{0,\pm 1\}} U(s_{\text{ldp}}^k(a+c, b+d)). \tag{5.15}$$

If there is no sender activated by LDP in $A_{a,b}^k$, then we define $s_{\text{ldp}}^k(a,b) = \phi$ and $U(s_{\text{ldp}}^k(a,b)) = 0$. Before proving Theorem 5.1.2, we first derive Lemma 5.1.1 and Lemma 5.1.2.

Lemma 5.1.1 *Given a link set $L' \subseteq L$ and a square with side-length no more than $g\delta'$, where δ' is the length of the shortest link in L', then the number of active links in L' with the senders in the square cannot exceed eg^2 in any feasible schedule, where $e = 2/\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} - 1\right)^2$ is a constant.*

Proof Suppose $(s_i, r_i), (s_j, r_j) \in L'$ are the two active links in a feasible schedule. For each sender $s_i \in \mathcal{P} \backslash s_j$, its interference factor on r_j ($f_{i,j}$) cannot be larger than γ_ε. Accordingly, $\left(\frac{d_{s_j,r_i}}{d_{s_i,r_i}}\right)^{-\alpha} \leq e^{\gamma_\varepsilon} - 1$. Since $\ln(1+x) \geq \frac{\gamma_\varepsilon x}{e^{\gamma_\varepsilon}-1}$ when $x \in [0, e^{\gamma_\varepsilon} - 1]$, the interference factor of s_i on r_j ($f_{i,j}$) is lower bounded by

$$\begin{aligned}
f_{i,j} &= \ln\left(1 + \frac{d_{s_i,r_j}^{-\alpha}\gamma_{\text{th}}}{d_{s_j,r_j}^{-\alpha}}\right) && (5.16)\\
&\geq \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon}-1}\frac{d_{s_i,r_j}^{-\alpha}\gamma_{\text{th}}}{d_{s_j,r_j}^{-\alpha}} && (5.17)\\
&\geq \frac{\gamma_\varepsilon\gamma_{\text{th}}}{e^{\gamma_\varepsilon}-1}\frac{(d_{s_j,r_j} + d_{s_i,s_j})^{-\alpha}}{d_{s_j,r_j}^{-\alpha}} && (5.18)\\
&\geq \frac{\left(1 + d_{s_i,s_j}/\delta'\right)^{-\alpha}\gamma_\varepsilon\gamma_{\text{th}}}{e^{\gamma_\varepsilon}-1}. && (5.19)
\end{aligned}$$

Since $f_{i,j}$ cannot exceed γ_ε, we have $\frac{\left(1+d_{s_i,s_j}/\delta'\right)^{-\alpha}\gamma_\varepsilon\gamma_{\rm th}}{e^{\gamma\varepsilon}-1} \le \gamma_\varepsilon$, which implies that

$$d_{s_i,s_j} \ge \left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\rm th}\right)^{\frac{1}{\alpha}} - 1\right)\delta'. \tag{5.20}$$

Then, any two active senders cannot be located in a square with size $\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\rm th}\right)^{\frac{1}{\alpha}} - 1\right)\frac{\delta'}{\sqrt{2}}$, which also implies at most $\frac{2g^2}{\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\rm th}\right)^{\frac{1}{\alpha}}-1\right)^2}$ active senders can be located in a square with side-length $g\delta'$.

Lemma 5.1.2 *For any square $A^k_{a,b} \in \mathcal{A}^k$, there must exist a LDP's sender that is in the nine squares surrounding $A^k_{a,b}$, such that the sender's data rate is no smaller than the data rate of all the senders in $A^k_{a,b}$ selected by the optimal schedule.*

$$\arg\max_{s\in B^k_{\rm opt}(a,b)} U(s) \le \arg\max_{c,d\in\{0,\pm1\}} U(s^k_{\rm ldp}(a+c,b+d)) \tag{5.21}$$

Proof For the sake of contradiction, we assume that

$$\arg\max_{s\in B^k_{\rm opt}(a,b)} U(s) > \arg\max_{c,d\in\{0,\pm1\}} U(s^k_{\rm ldp}(a+c,b+d)) \tag{5.22}$$

Then LDP would select $s^k_{\rm opt}(a,b)$ as an active sender, because all the senders in the squares around $A^k_{a,b}$ (also including $A^k_{a,b}$) selected by LDP have lower data rate than $s^k_{\rm opt}(a,b)$, which is a contradiction.

Theorem 5.1.2 *The approximation ratio of LDP is $O(g(L))$.*

Proof According to Lemma 5.1.1, given an optimal schedule $\mathcal{P}_{\rm opt}$, the maximum number of senders in a square with side-length β_k in $\mathcal{P}^k_{\rm opt}$ cannot exceed $e\beta^2$, then the following bound holds

$$U(\mathcal{P}^k_{\rm opt}) = \sum_{a,b} U(B^k_{\rm opt}(a,b)) \le e\beta^2 \sum_{a,b} U(s^k_{\rm opt}(a,b)) \tag{5.23}$$

By Lemma 5.1.2, Equ. (5.14), and Equ. (5.15), $\forall A^k_{a,b} \in \mathcal{A}^k$ we have $U(s^k_{\rm opt}(a,b)) \le U(\hat{s}^k_{\rm ldp}(a,b))$, then

$$e\beta^2 \sum_{a,b} U(s^k_{\rm opt}(a,b)) \le e\beta^2 \sum_{a,b} U(\hat{s}^k_{\rm ldp}(a,b)). \tag{5.24}$$

Here, notice that $\hat{s}^k_{\rm ldp}(a,b)$ with different locations (a,b) may be the same sender in $\mathcal{P}_{\rm ldp}$, which can

be repeatedly counted by Equ. (5.24) at most nine times. Then, we have

$$\sum_{a,b} U(\hat{s}^k_{\text{ldp}}(a,b)) \leq 9 \sum_{a,b} U(s^k_{\text{ldp}}(a,b)) = 9U(\mathcal{P}^k_{\text{ldp}}). \tag{5.25}$$

According to Equ. (5.23), (5.24) and (5.25), we have

$$U(\mathcal{P}^k_{\text{opt}}) \leq 9e\beta^2 U(\mathcal{P}^k_{\text{ldp}}). \tag{5.26}$$

And consequently, we can obtain that

$$U(\mathcal{P}_{\text{ldp}}) \geq \frac{\sum_{k=1}^{g(L)} U(\mathcal{P}^k_{\text{ldp}})}{g(L)} \geq \frac{\sum_{k=1}^{g(L)} U(\mathcal{P}^k_{\text{opt}})}{9e\beta^2 g(L)} = \frac{U(\mathcal{P}_{\text{opt}})}{9e\beta^2 g(L)} \tag{5.27}$$

which implies that $\frac{U(\mathcal{P}_{\text{opt}})}{U(\mathcal{P}_{\text{ldp}})} \leq 9e\beta^2 g(L)$.

5.1.2 Recursive Link Elimination Algorithm

In this part, we consider a special case of *Fading-R-LS*, in which the transmission rate of each link is the same, i.e., $\lambda_i = \lambda, \forall 1 \leq i \leq N$. We propose a greedy algorithm, namely recursive link elimination algorithm (RLE), for this special case. Algorithm 8 shows the pseudocode for RLE. In each iteration, the algorithm first greedily selects the unpicked sender with the shortest link length, say s_i. The rationale of this strategy is that the signal power received by the receiver with a shorter link is always stronger, and hence the receiver is more likely to successfully receive the packet. Then, all links whose senders are within the radius $c_1 d_{s_i,r_i}$ of the receiver r_i are removed from L, where c_1 is a constant to be set later on (in Formula (5.42)). Second, all senders whose receivers have interference factors above c_2 from the selected senders are removed, where c_2 is a constant smaller than 1. This process is repeated until all links in L have been either active or deleted. Note that though this algorithm has a number of iterations, all identified active links conduct transmissions in one time slot simultaneously.

Below, we prove that the schedule obtained by RLE is both feasible (Theorem 5.1.3) and efficient, i.e., only a constant factor away from the optimal (Theorem 5.1.4). We use r_i to represent a receiver whose desired sender is selected in Algorithm 8, and use $\mathcal{P}_i^-$ and $\mathcal{P}_i^+$ to denote the set of senders added after and before s_i is selected, respectively.

Algorithm 5: Pseudo-code for Recursive Link Elimination algorithm (RLE).

input : $S = \{s_1, ..., s_N\}$
output: $\mathcal{P}$
$\mathcal{P} \leftarrow \phi$;
while $S \neq \mathcal{P}$ **do**
  Pick up the sender s_i that has the shortest link length in L and add it to $\mathcal{P}$;
  Remove each sender s_j, *s.t.* $d_{s_j,r_i} < c_1 d_{s_i,r_i}$ from S;
  Remove each sender s_j, *s.t.* $f_{\mathcal{P},r_j} > c_2\gamma_\varepsilon$ from S;
return $\mathcal{P}$;

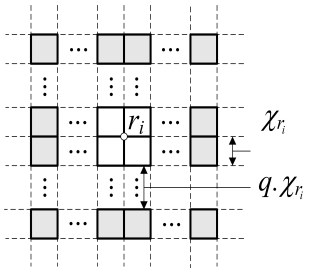


Figure 5.3: Proof of Theorem 5.1.3.

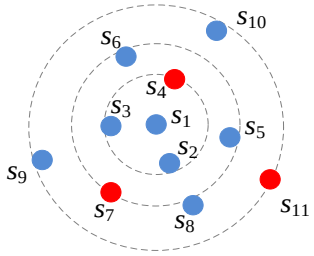


Figure 5.4: z-blue-dominant.

Lemma 5.1.3 *The distance between any two senders in $\mathcal{P}_i^+$ is lower bounded by $(c_1 - 1)d_{s_i,r_i}$.*

Proof For any receiver r_j whose desired sender s_j is in $\mathcal{P}_i^+$, there is no other sender, say s_l, in $\mathcal{P}_i^+$ that has distance smaller than $c_1 d_{s_j,r_j}$ from r_j. Based on this and the triangular inequality, we can calculate the lower bound of the distance between any two senders in $\mathcal{P}_i^+$:

$$\begin{aligned} d_{s_l,s_j} &\geq d_{s_l,r_j} - d_{s_j,r_j} && (5.28)\\ &\geq c_1 d_{s_j,r_j} - d_{s_j,r_j} && (5.29)\\ &\geq d(s_l, r_j) - d_{s_j,s_j} && (5.30)\\ &\geq (c_1 - 1)d_{s_j,r_j} && (5.31)\\ &\geq (c_1 - 1)d_{s_i,r_i}. && (5.32) \end{aligned}$$

Theorem 5.1.3 *In the link elimination algorithm (RLE), all the senders in the active link set $\mathcal{P}$ can successfully transmit the packet.*

Proof When sender s_i is added to the schedule, the interference factor of $\mathcal{P}_i^-$ on r_i must be no larger than $c_2\gamma_\varepsilon$; otherwise it has been deleted in a previous step. Therefore, the interference factor on r_i from concurrent active link set $\mathcal{P}_i^-$ is $f_{\mathcal{P}_i^-,r_i} \leq c_2\gamma_\varepsilon$. It remains to show that $f_{\mathcal{P}_i^+,r_i} \leq (1 - c_2)\gamma_\varepsilon$.

We partition the entire network region into squares with size $\chi_i \times \chi_i$ (see Fig. 5.3), where $\chi_i = (c_1 - 1)d_{s_i,r_i}/\sqrt{2}$. According to Lemma 5.1.3, any two senders in $\mathcal{P}_i^+$ cannot be located in the same square because the distance between the senders in $\mathcal{P}_i^+$ is at least $(c_1 - 1)d_{s_i,r_i}$, and the size of square is $(c_1 - 1)d_{s_i,r_i}/\sqrt{2}$. We use $\mathcal{Q}_q^i$ to denote the set of senders in the squares that is $q\chi_i$ away from r_i. Then, there are at most $4(2q + 1)$ senders in $\mathcal{Q}_q^i$. The distance between the senders in $\mathcal{Q}_q^i$ and r_i is at least $q\chi_i$, hence the interference factor of any sender $s_j \in \mathcal{Q}_q^i$ on r_i is at most

$$\begin{aligned} f_{j,i} &= \ln\left(1 + \frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}\right) && (5.33)\\ &\leq \frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}} && (5.34)\\ &\leq \frac{(q\chi_i)^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}. && (5.35) \end{aligned}$$

The interference factor of $\mathcal{Q}_q^i$ on r_i is then upper bounded by

$$\begin{aligned} f_{\mathcal{Q}_q^i,r_i} &= \sum_{s_j \in \mathcal{Q}_q^i} f_{j,i} && (5.36)\\ &\leq \frac{4(2q+1)(q\chi_i)^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}, && (5.37) \end{aligned}$$

and the interference factor of all active links $\mathcal{P}_i^+ = \cup_q \mathcal{Q}_q^i$ on r_i is upper bounded by

$$\begin{aligned} f_{\mathcal{P}_i^+,r_i} &= \sum_{q=1}^{\infty} f_{\mathcal{Q}_q^i,r_i} && (5.38)\\ &\leq \sum_{q=1}^{\infty} \frac{4(2q+1)q^{-\alpha}\chi_i^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}} && (5.39)\\ &\leq \sum_{q=1}^{\infty} \frac{12q \cdot q^{-\alpha}\chi_i^{-\alpha} \cdot \gamma_{\text{th}}}{d_{i,i}^{-\alpha}} && (5.40)\\ &\leq \frac{12\chi_i^{-\alpha} \cdot \gamma_{\text{th}}}{d_{i,i}^{-\alpha}}\zeta(\alpha - 1). && (5.41) \end{aligned}$$

We set c_1 by (to make $f_{\mathcal{P}_i^+,r_i} \leq (1 - c_2)\gamma_\varepsilon$)

$$c_1 = \sqrt{2}\left(12\zeta(\alpha - 1)\gamma_{\text{th}}/(\gamma_\varepsilon(1 - c_2))\right)^{\frac{1}{\alpha}} + 1, \quad (5.42)$$

we can get that $f_{\mathcal{P}_i^+,r_i} \leq \frac{12\chi_i^{-\alpha}\zeta(\alpha-1)\gamma_{\text{th}}}{d_{s_i,r_i}^{-\alpha}} = (1 - c_2)\gamma_\varepsilon$,

In the following, we then analyze the efficiency of RLE. We first derive Lemmas 5.1.4 - 5.1.6, based on which we prove Theorem 5.1.4.

Lemma 5.1.4 *Let $\mathcal{P}$ be a feasible solution and let $s_i \in \mathcal{P}$. The number of senders in $\mathcal{P}\backslash s_i$ with distance kd_{s_i,r_i} from s_i is at most $\frac{e^{\gamma_\varepsilon}-1}{\gamma_{\text{th}}}(1+k)^\alpha$.*

Proof For each sender $s_j \in \mathcal{P}\backslash s_i$, its interference factor on r_i ($f_{j,i}$) cannot be larger than γ_ε. Accordingly, $\left(\frac{d_{j,i}}{d_{i,i}}\right)^{-\alpha} \leq e^{\gamma_\varepsilon}-1$. Since $\ln(1+x) \geq \frac{\gamma_\varepsilon x}{e^{\gamma_\varepsilon}-1}$ when $x \in [0, e^{\gamma_\varepsilon}-1]$, $f_{j,i}$ is lower bounded by

$$\begin{aligned} f_{j,i} &= \ln\left(1+\frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}\right) && (5.43)\\ &\geq \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon}-1}\frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}} && (5.44)\\ &\geq \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon}-1}\frac{(d_{i,i}+d_{j,i})^{-\alpha}}{d_{i,i}^{-\alpha}}\gamma_{\text{th}} && (5.45)\\ &\geq \frac{\gamma_\varepsilon\,(1+k)^{-\alpha}}{e^{\gamma_\varepsilon}-1}\gamma_{\text{th}}. && (5.46)\end{aligned}$$

Since the interference factor of $\mathcal{P}\backslash s_i$ on r_i cannot exceed γ_ε, there are at most $\frac{e^{\gamma_\varepsilon}-1}{\gamma_{\text{th}}}(1+k)^\alpha = \frac{\varepsilon}{(1-\varepsilon)\gamma_{\text{th}}}(1+k)^\alpha$ such senders.

Definition 5.1.2 (z-blue-dominant [20]) *Let $\mathcal{N}_r$ and $\mathcal{N}_b$ be two disjoint sets of points in a 2D Euclidean space, namely red and blue points, respectively. Let circle $B_d(s_b)$ be the set of points p such that $d(p,s_b) \leq d$. Then, for any positive integer z, a point $s_b \in \mathcal{N}_b$ is* z-blue-dominant *if every circle $B_d(s_b)$ contains z times more blue than red points, or formally*

$$|B_d(s_b) \cap N_b| > z|B_d(s_b) \cap N_r| \quad \forall d \in R^+. \tag{5.47}$$

Fig. 5.4 gives an example for this definition: $\mathcal{N}_r = \{s_4, s_7, s_{11}\}$ and $\mathcal{N}_b = \{s_1, s_2, s_3, s_5, s_6, s_8, s_9, s_{10}\}$. Because every circle centered at s_1 contains 2 times more blue points than red points, s_1 is a 2-blue-dominant.

Lemma 5.1.5 (Blue-dominant centers lemma [20]) *For any positive integer z, if $|\mathcal{N}_b| > 5z|\mathcal{N}_r|$ then there exists at least one z-blue-dominant point s_b in $\mathcal{N}_b$. In addition, given a z-blue-dominant point s_b, for each point s_r in $\mathcal{N}_r$, there exists a subset of $\mathcal{N}_b$ corresponding to s_r, denoted by $G(s_r)$,* s.t.,

1) any point in $G(s_r)$ is farther from s_r than from s_b; 2) for any pair of points $s_r, s'_r \in \mathcal{N}_r$, $G(s_r)$ and $G(s'_r)$ are disjoint; and 3) the number of points in each subset $G(s_r)$ is no smaller than z.

Proof See the proof in Lemma 4.4 in [20].

Lemma 5.1.6 *Denote the set of all senders in the optimal schedule and RLE by $\mathcal{P}_{\text{opt}}$ and $\mathcal{P}_{\text{rle}}$, respectively. Then, $|\mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}| \leq \frac{3^\alpha \times 5\varepsilon}{c_2(1-\varepsilon)\gamma_{\text{th}}} |\mathcal{P}_{\text{rle}}|$.*

Proof For the sake of contradiction, we assume that $|\mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}| > \frac{3^\alpha \times 5\varepsilon}{c_2(1-\varepsilon)\gamma_{\text{th}}} |\mathcal{P}_{\text{rle}}|$. We label the set of senders in $\mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}$ by blue ($\mathcal{N}_b = \mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}$) and those in $\mathcal{P}_{\text{rle}}$ by red ($\mathcal{N}_r = \mathcal{P}_{\text{rle}}$). By Lemma 5.1.5, there is a z-blue-dominant point (sender) $s_i \in \mathcal{N}_b$, where $z = \frac{3^\alpha \varepsilon}{c_2(1-\varepsilon)\gamma_{\text{th}}}$. We shall argue that the sender s_i would have been picked by RLE, leading to a contradiction.

According to Lemma 5.1.5, for any red point $s_j \in \mathcal{N}_r$, there exists a subset of blue points $G(s_j)$ such that all the points in $G(s_j)$ are closer to s_i than to s_j and $|G(s_j)| \geq z$ ($z = \frac{3^\alpha \varepsilon}{(1-\varepsilon)\gamma_{\text{th}}}$). We can derive that $d_{s_i,s_j} > 2d_{s_i,r_i}$; otherwise the number of senders within distance $2d_{s_i,r_i}$ from s_i would be larger than $\frac{(2+1)^\alpha \varepsilon}{c_2(1-\varepsilon)\gamma_{\text{th}}} \geq \frac{3^\alpha \varepsilon}{(1-\varepsilon)\gamma_{\text{th}}}$, which contradicts with the conclusion in Lemma 5.1.4. Based on the triangle inequality, $d_{s_j,r_i} \geq d_{s_i,s_j} - d_{s_i,r_i} > d_{s_i,s_j}/2$. For any point $s_l \in G(s_j)$,

$$
\begin{aligned}
d_{s_l,r_i} &\leq d_{s_l,s_i} + d_{s_i,r_i} && (5.48)\\
&< d_{s_j,s_i} + d_{s_i,r_i} && (5.49)\\
&< d_{s_j,s_i} + d_{s_j,s_i}/2 && (5.50)\\
&= 3d_{s_j,s_i}/2. && (5.51)
\end{aligned}
$$

Hence, the sum interference factor of the blue senders in $G(s_j)$ on r_i is lower bounded

$$
\begin{aligned}
\sum_{s_l \in G(s_j)} f_{l,i} &= \sum_{s_l \in G(s_j)} \ln\left(1 + \frac{d_{l,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}\right) && (5.52)\\
&\geq \sum_{s_l \in G(s_j)} \frac{\gamma_\varepsilon d_{l,i}^{-\alpha}\gamma_{\text{th}}}{(e^{\gamma_\varepsilon}-1)d_{i,i}^{-\alpha}} && (5.53)\\
&> \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon}-1}\frac{e^{\gamma_\varepsilon}-1}{c_2\gamma_{\text{th}}}3^\alpha\frac{3^{-\alpha}}{2^{-\alpha}}\frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}} && (5.54)\\
&= \frac{d_{j,i}^{-\alpha}}{2^{-\alpha}d_{i,i}^{-\alpha}}\frac{\gamma_\varepsilon}{c_2} && (5.55)\\
&> \frac{d_{j,i}^{-\alpha}}{d_{i,i}^{-\alpha}}\frac{\gamma_\varepsilon}{c_2} && (5.56)
\end{aligned}
$$

$$\geq \frac{\gamma_\varepsilon}{c_2} \ln\left(1 + \frac{d_{j,i}^{-\alpha}\gamma_{\text{th}}}{d_{i,i}^{-\alpha}}\right) \tag{5.57}$$

$$= \frac{\gamma_\varepsilon f_{j,i}}{c_2}. \tag{5.58}$$

This relationship holds for any $s_j \in \mathcal{N}_r$, and $G(s_j)$ and $G(s_l)$ are disjoint $\forall s_j, s_l \in \mathcal{N}_r$, then the total interference factor that r_i receives from the senders in $\mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}$ (blue points) is at least $\frac{\gamma_\varepsilon}{c_2}$ times as high as the interference factor it would receive from the senders in RLE (red points). Because the interference factor of $\mathcal{N}_b$ on r_i is at most γ_ε. Therefore, we have $f_{\mathcal{N}_r, r_i} < \frac{f_{\mathcal{N}_b, r_i}}{\frac{\gamma_\varepsilon}{c_2}} \leq \frac{\gamma_\varepsilon}{\frac{\gamma_\varepsilon}{c_2}} = c_2$, which implies that s_i should not have been deleted by RLE, which establishes the contradiction.

Theorem 5.1.4 *The approximation ratio of the link elimination algorithm (RLE) is* $O(1)$.

Proof Denote the number of receivers informed by RLE and the optimal schedule by U_{rle} and U_{opt}, respectively. Then, according to Lemma 5.1.6,

$$\frac{U_{\text{opt}}}{U_{\text{rle}}} \leq \frac{|\mathcal{P}_{\text{opt}}|}{|\mathcal{P}_{\text{rle}}|} = \frac{|\mathcal{P}_{\text{opt}} \backslash \mathcal{P}_{\text{rle}}|}{|\mathcal{P}_{\text{rle}}|} + 1 \leq \frac{3^\alpha \times 5\varepsilon}{c_2(1-\varepsilon)\gamma_{\text{th}}\gamma_\varepsilon} + 1 = O(1). \tag{5.59}$$

5.1.3 Decentralized Link Scheduling

The approximation algorithms described previously are centralized. In this section, we propose a decentralized algorithm, named decentralized link scheduling (DLS), that allows each sender to make its own decision based on limited local information. We assume that receivers periodically provide the SIR information to their senders. A natural way of viewing this setting is as a game, where the senders are the players and they need to determine whether to send a packet. We will define such a mixed game and show that every Nash equilibrium results in an expected number of successful transmissions that is close to the optimal. In the game, each sender is a player and its strategy is either transmitting (represented by 1) or not transmitting (represented by 0). Let q_i denote the probability that sender s_i transmits. A sender receives payoff 0 if it does not transmit, payoff 1 if it transmits and its receiver receives the packet, and payoff -1 if it transmits but its receiver does not successfully receive the packet.

We then show that any mixed Nash equilibrium has the expected throughtput close to the optimal solution. We begin with a few lemmas. For each sender s_i, let p_i be the probability that s_i would be successful if it were to transmit. Then, $M = \sum_i q_i p_i$ is the expected number of successful

transmissions. Let $Q = \sum_{s_i \in S} q_i$ be the expected number of transmissions.

Lemma 5.1.7 *For any Nash equilibrium, for any sender s_i, if $q_i < 1$ then $p_i \leq 1/2$, and if $q_i > 0$ then $p_i \geq 1/2$.*

Proof Suppose $q_i < 1$ and $p_i > 1/2$. Then, the expected payoff to s_i is $E_{s_i} = q_i\ (p_i - (1 - p_i) = q_i(2p_i - 1)$. Since $2p_i - 1 > 0$, s_i's payoff is maximized by setting $q_i = 1$, which contradicts our assumption that $q_i < 1$ is an equilibrium. Similarly, suppose $q_i > 0$ and $p_i \leq 1/2$. Then, when s_i transmits, the probability that its transmission fails is more than $1/2$. Subsequently, s_i receives negative expected payoff $E_{s_i} < 0$ and it will choose not to transmit (i.e., $q_i = 0$), which contradicts our assumption that $q_i > 0$ is an equilibrium.

By Lemma 5.1.7, we can derive that $Q = \sum_{s_i \in S} q_i = 2 \sum_{s_i \in S} \frac{1}{2} q_i \leq 2 \sum_{s_i \in S} p_i q_i = 2M$.

Lemma 5.1.8 *In any feasible schedule, the number of active senders in a ball with radius $k\delta$ is no more than ek^2, where $e = \frac{4\sqrt{2}}{\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} - 1\right)\delta}$ and δ is the length of the shortest link in L.*

Proof Suppose s_i and s_j are two active senders in a feasible schedule, then the interference factor of s_i on r_j ($f_{i,j}$) is lower bounded by

$$
\begin{aligned}
f_{i,j} &= \ln\left(1 + \frac{d_{i,j}^{-\alpha}\gamma_{\text{th}}}{d_{j,j}^{-\alpha}}\right) && (5.60)\\
&\geq \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon} - 1} \frac{d_{i,j}^{-\alpha}\gamma_{\text{th}}}{d_{s_j,r_j}^{-\alpha}} && (5.61)\\
&\geq \frac{\gamma_\varepsilon}{e^{\gamma_\varepsilon} - 1} \frac{(d_{j,j} + d_{i,j})^{-\alpha}}{d_{j,j}^{-\alpha}} && (5.62)\\
&\geq \frac{(1 + d_{i,j}/\delta)^{-\alpha}\,\gamma_\varepsilon\gamma_{\text{th}}}{e^{\gamma_\varepsilon} - 1}. && (5.63)
\end{aligned}
$$

Since $f_{i,j}$ cannot be larger than γ_ε, we have

$$
\frac{(1 + d_{i,j}/\delta)^{-\alpha}\,\gamma_\varepsilon\gamma_{\text{th}}}{e^{\gamma_\varepsilon} - 1} \leq \gamma_\varepsilon \tag{5.64}
$$

and hence we can get the upper bound of the distance between s_i and s_j, say $d_{i,j} \leq \left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} - 1\right)\delta$. Then, any two active senders cannot be located in a square with size $\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} - 1\right)\frac{\delta}{\sqrt{2}}$, which also implies at most $\frac{4\sqrt{2}k^2}{\left(\left(\frac{1-\varepsilon}{\varepsilon}\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} - 1\right)\delta}$ active senders can be located in a square with side-length $2k\delta$.

Finally, Lemma 5.1.8 follows immediately from the fact that any ball with radius $k\delta$ is contained in a square with side-length $2k\delta$. Accordingly, $\left(\frac{d_{s_i,r_j}}{d_{s_j,r_j}}\right)^{-\alpha} \leq e^{\gamma_\varepsilon} - 1$. Since $\ln(1+x) \geq \frac{\gamma_\varepsilon x}{e^{\gamma_\varepsilon}-1}$ when $x \in [0, e^{\gamma_\varepsilon} - 1]$, $f_{i,j}$ is lower bounded by

Let $\mathcal{R}_{\text{opt}}$ be the set of receivers that achieve their SIR requirements in the optimal solution. We then prove the approximation ratio of DLS.

Theorem 5.1.5 *Any Nash equilibrium has an expected number of successful transmissions at least* $\Omega(\frac{|\mathcal{R}_{\text{opt}}|}{\Delta^\alpha})$*, where* Δ *is the ratio between the maximum and the minimum distances between nodes.*

Proof Let $\mathcal{R}$ be the set of receivers whose senders are active with probability 1. Each receiver $r_j \in \mathcal{R}_{\text{opt}} \backslash \mathcal{R}$ has a variable b_j (initialized to 0) to record the sum probability of its nearby senders that are active. For each s_i, let $\mathcal{R}_i$ be the $\lfloor \frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| - \eta}{\eta Q} \rfloor$ closest receivers in $\mathcal{R}_{\text{opt}} \backslash \mathcal{R}$ to s_i such that $b_j < 1$, where $\eta = 4/\gamma_\varepsilon$. Then, we increase the b value of each receiver in $\mathcal{R}_i$ by q_i. Since $Q = \sum_{s_i \in S} q_i$ and each sender s_i increases the sum of b values of all receivers by at most $q_i \lfloor \frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| - \eta}{\eta Q} \rfloor$, we can derive that

$$\begin{aligned} \sum_{r_j \in R} b_j &\leq \sum_{s_i \in S} q_i \lfloor \frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| - \eta}{\eta Q} \rfloor & (5.65) \\ &\leq \frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| - \eta}{\eta} & (5.66) \\ &< \frac{|\mathcal{R}_{\text{opt}}|}{\eta}, & (5.67) \end{aligned}$$

which implies that there is a receiver $r_j \in \mathcal{R}_{\text{opt}} \backslash \mathcal{R}$ such that the sum probability from nearby senders is less than $1/\eta$. Let $\mathcal{M}_j$ be the set of senders that have contributed to b_j, and $\mathcal{M}_j^c$ be the set of all other senders. Then, $\sum_{s_i \in \mathcal{M}_j} q_i \leq 1/\eta$. For every distance d, let $z(r_j, d) = \sum_{s_i \in \mathcal{M}_j^c : d_{s_i,r_j} \leq d} q_i$ be the sum probability from $\mathcal{M}_j^c$ located inside $B(r_j, d)$ to r_j. Since $r_j \notin \mathcal{R}_i$, any receiver $r_k \in \mathcal{R}_i$ must have $d_{s_i,r_k} \leq d_{s_i,r_j}$, or else r_j would be in $\mathcal{R}_i$. Then, for some sender $s_i \in \mathcal{M}_j^c$, by the triangle inequality we know that

$$d_{r_k,r_j} \leq d_{s_i,r_k} + d_{s_i,r_j} \leq 2d_{s_i,r_j}. \quad (5.68)$$

Thus, any sender s_i at distance at most d from r_j must have its entire $\mathcal{R}_i$ at distance at most $2d$ from r_j.

We then calculate the upper bound of $z(r_j, d)$. Since each sender s_i in $\mathcal{M}_j \cap B(r_j, d)$ contributes $q_i \lfloor \frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| - \eta}{\eta Q} \rfloor$ probability mass, and each receiver that s_i contributes to must be in

$B(r_j, 2d)$, the sum probability mass of receivers in $B(r_j, 2d)$ is at least $z(r_j, d)\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor$. Since a receiver's b value increases only when $b < 1$ by at most 1, we know that the b value is at most 2. Thus, the number of receivers from $\mathcal{R}_{\text{opt}}$ in $B(r_j, 2d)$ is at least $\frac{z(r_j,d)}{2}\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor$. By Lemma 5.1.8, we get $ek^2 \geq \frac{z(r_j,k\delta)}{2}\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor$ and hence

$$z(r_j, k\delta) \leq \frac{2ek^2}{\left\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\right\rfloor}. \tag{5.69}$$

Now that we have a bound on the probability mass inside a circle around r_j, we want to bound the probability mass in an annulus of thickness δ around r_j. To do this, we note that the interference at r_j is maximized if every circle around r_j actually meets the above bound. We aim to calculate the upper bound of the total interference, which needs to calculate the upper bound of each circle that is shown in Formula (5.69). It implies that the sum of the probability mass of the senders between distance $k\delta$ and $(k+1)\delta$ from r_j is at most $2e((k+1)^2 - k^2)/\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor \leq 6ke/\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor$. For a sender at distance $k\delta$ from r_j, its interference factor is upper bounded by

$$\begin{aligned} f_{i,j} &= \ln\left(1 + \frac{(k\delta)^{-\alpha}\gamma_{\text{th}}}{d_{s_j,r_j}^{-\alpha}}\right) && (5.70)\\ &\leq \frac{(k\delta)^{-\alpha}\gamma_{\text{th}}}{d_{s_j,r_j}^{-\alpha}} && (5.71)\\ &\leq \frac{\gamma_{\text{th}}\Delta^{\alpha}}{k^{\alpha}}. && (5.72) \end{aligned}$$

It implies that the expected interference factor at r_j caused by senders at distance between $k\delta$ and $(k+1)\delta$ from r_j is at most $\frac{6e\Delta^{\alpha}k^{1-\alpha}\gamma_{\text{th}}}{\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor}$. Using linearity of expectations, we can sum over the annuli to get that the expected interference factor of the senders with distance larger than δ from r_j in $\mathcal{M}_j$ is at most $\frac{6e\zeta(\alpha-1)\Delta^{\alpha}\gamma_{\text{th}}}{\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor}$. For the senders between distances 0 and δ from r_j, since the interference factor caused by each sender s_i is at most $f_{i,j} = \ln\left(1 + d_{s_i,r_j}^{-\alpha}\gamma_{\text{th}}/d_{s_j,r_j}^{-\alpha}\right) \leq d_{s_i,r_j}^{-\alpha}\gamma_{\text{th}}/d_{s_j,r_j}^{-\alpha} \leq \Delta^{\alpha}\gamma_{\text{th}}$, their expected interference factor on r_j is at most $\frac{2e\Delta^{\alpha}\gamma_{\text{th}}}{\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor}$. Since $\sum_{r_i\in\mathcal{M}_j^c} p_i \leq 1/\eta$, we get that they cause at most $1/\eta$ expected interference. Thus, the total expected interference is upper bounded by $\frac{8e\zeta(\alpha-1)\Delta^{\alpha}\gamma_{\text{th}}}{\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor} + \frac{1}{\eta}$.

We now have a bound on the expected interference. By using Markov's inequality, we get that the probability that r_j hears interference at least twice the expected interference is at most $1/2$. Since each r_j has $q_j < 1$, and Lemma 5.1.7 implies that $p_j \leq 1/2$, then $\frac{16e\zeta(\alpha-1)\Delta^{\alpha}\gamma_{\text{th}}}{\lfloor\frac{|\mathcal{R}_{\text{opt}}\setminus\mathcal{R}|-\eta}{\eta Q}\rfloor} + \frac{2}{\eta}$

must be no smaller than γ_ε. Accordingly, we can get $Q \geq \Omega(\frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}|}{\Delta^\alpha})$, and hence $M \geq \Omega(\frac{|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}|}{\Delta^\alpha}$. If $|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| = o(|\mathcal{R}_{\text{opt}}|)$, then a constant fraction of senders transmit with probability 1, which implies that the expected number of successful transmissions in the Nash is at least $\Omega(|\mathcal{R}_{\text{opt}}|)$. If $|\mathcal{R}_{\text{opt}} \backslash \mathcal{R}| = \Omega(|\mathcal{R}_{\text{opt}}|)$, then the above relationship implies $M \geq \Omega(\frac{|\mathcal{R}_{\text{opt}}|}{\Delta^\alpha})$, which proves the theorem.

5.2 Cooperative Communication Link Scheduling

In this section, we propose a Link Length Diversity (LLD) based algorithm for both CLS (Section 5.2.2) and OCLS (Section 5.2.3), with a bounded performance guarantee $O(g(\mathcal{K}))$. In addition, we propose a constant approximation ratio algorithm for OCLS, when the link set size for each request is upper bounded by a constant (Section 5.2.4). Before presenting these algorithms, we first introduce some definitions and notations (Section 5.2.1).

5.2.1 Definitions And Notations

Definition 5.2.1 *(*Relative interference (RI)*) Given a receiver r_i and its active desired link set $\mathcal{I}_i$, the RI of link $l_{s,r}$ ($r \neq r_i$) on r_i is the increase caused by $l_{s,r}$ in the inverse of the SIR at r_i, scaled by γ_{th}*

$$RI_{l_{s,r}}(r_i, \mathcal{I}_i) = \gamma_{\text{th}} \frac{d_{s,r_i}^{-\alpha}}{\sum_{l \in \mathcal{I}_i} d(l)^{-\alpha}}. \tag{5.73}$$

Similarly, the RI of a set of links $\mathcal{I}'$ ($\mathcal{I}' \cap \mathcal{I}_i = \phi$) on r_i is the sum RI of the links in $\mathcal{I}'$ on r_i

$$RI_{\mathcal{I}'}(r_i, \mathcal{I}_i) = \sum_{l \in \mathcal{I}'} RI_l(r_i, \mathcal{I}_i). \tag{5.74}$$

Property 5.2.1 *Suppose all links in a link set $\mathcal{I}'$ are activated simultaneously, then a receiver r_i, with active desired link set $\mathcal{I}_i$, can be informed iff $RI_{\mathcal{I}'}(r_i, \mathcal{I}_i) \leq 1$.*

Lemma 5.2.1 *Given a set of disjoint link sets $\mathcal{I}_1$, ..., $\mathcal{I}_n$ and a receiver r_i, which has active desired link set $\mathcal{I}_i$ ($\mathcal{I}_i \cap \mathcal{I}_j = \phi, \forall 1 \leq j \leq n$), the RI of the union $\mathcal{I} = \cup_{j=1}^n \mathcal{I}_j$ on a receiver r_i is the sum RI of all link sets $\mathcal{I}_1, ..., \mathcal{I}_n$ on r_i:*

$$RI_{\mathcal{I}}(r_i, \mathcal{I}_i) = \sum_{j=1}^{n} RI_{\mathcal{I}_j}(r_i, \mathcal{I}_i). \tag{5.75}$$

Proof By Definition 5.2.1,

$$RI_{\mathcal{I}}(r_i, \mathcal{I}_i) = \sum_{j=1}^{n} \sum_{l \in \mathcal{I}_j} RI_l(r_i, \mathcal{I}_i) \tag{5.76}$$

$$= \sum_{j=1}^{n} RI_{\mathcal{I}_j}(r_i, \mathcal{I}_i). \tag{5.77}$$

Definition 5.2.2 *(*Key link*) Given a receiver r_i and its link set $\mathcal{I}_i$. The* key link *of r_i, denoted by $\kappa(r_i)$, is defined as the shortest link in $\mathcal{I}_i$:*

$$k(r_i) = \arg\min\{d(l) | l \in I_i\}. \tag{5.78}$$

In the following, we use $\mathcal{K} = \{\kappa(r_1), ..., \kappa(r_N)\}$ to denote the set of all key links.

Definition 5.2.3 *(*Receiver density*) Given a set of receivers $\mathcal{R}$ and an area A (*e.g., *a square), the receiver density of $\mathcal{R}$ in A is defined as the number of receivers in $\mathcal{R}$ that reside in A.*

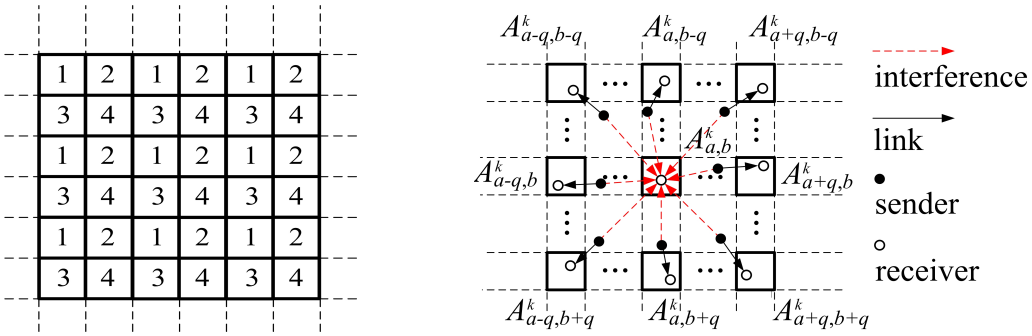
5.2.2 LLD Based Algorithm For CLS

The LLD based algorithm for CLS (LLD-CLS for short) consists of three steps: 1) Calculate the key link for each receiver; 2) Build disjoint link classes according to the links' length; 3) For each link class, construct a feasible schedule using a greedy strategy. In the following we introduce this algorithm in detail.

As we stated in the introduction part, CC can help each receiver decode the message from its desired link set. However, it also generates more interference to other links that transmit the message simultaneously. Hence, we set a link size constraint Δ for each request in LLD-CLS. The algorithm starts by calculating the key link set $\mathcal{K}$ and its link length set $G(\mathcal{K}) = \{h_1, ..., h_{g(\mathcal{K})}\}$. Then, we build $g(\mathcal{K})$ disjoint link classes L_1, ..., $L_{g(\mathcal{K})}$ from L, *s.t.*

$$L_k = \{l \in L | 2^{h_k} \cdot \sigma \leq d(l) < 2^{h_k+1} \cdot \sigma\} \tag{5.79}$$

where σ is the length of the shortest link in L. Next, each link set L_k is scheduled separately (see Algorithm 6). When scheduling L_k, the whole region is partitioned into a set of squares $\mathcal{A}^k = \{A^k_{a,b}\}$, where (a, b) represents the location of the square in the grid and each square has size $\beta_k = 2^{h_k+1} \cdot \sigma\beta$,



(a) Partition and coloring

(b) Proof of Theorem 5.2.1

Figure 5.5: LLD based algorithm for CLS.

where

$$\beta = \left(\frac{8\Delta(\alpha-1)\gamma_{\text{th}}}{\alpha-2}\right)^{\frac{1}{\alpha}}. \tag{5.80}$$

Then, all the squares in $\mathcal{A}^k$ are colored regularly with 4 colors (see Fig. 5.5 (a)). Links whose receivers belong to different cells of the same color are scheduled simultaneously (lines 6-12).

Notice that each receiver's key link must be in one of these classes. Hence, we can partition the receiver set R into $g(\mathcal{K})$ disjoint receiver classes $R_1, ..., R_{g(\mathcal{K})}$ based on the link classes the receivers' key links belong to, *i.e.*,

$$R_k = \{r_i | \kappa(r_i) \in L_k, r_i \in R\}. \tag{5.81}$$

In Algorithm 6, the goal of scheduling each link class L_k is actually to make all receivers in R_k be informed. In Theorem 5.2.1, we show that the schedule calculated by LLD-CLS is feasible, *i.e.*, any receiver $r_i \in R_k$ can be informed by the active links in L_k.

Theorem 5.2.1 *(*Feasibility*) LLD-CLS provide a feasible schedule.*

Proof Without loss of generality, we examine any receiver $r_i \in R_k$. Because $\kappa(r_i) \in L_k$, $2^{h_k}\sigma \leq \kappa(r_i) < 2^{h_k+1}\sigma$, which implies that the signal power received at r_i from its active desired link set $\mathcal{I}_i$ is at least

$$P_{\mathcal{I}_i, r_i} \geq \frac{P}{2^{\alpha(h_k+1)} \cdot \sigma^{\alpha}}. \tag{5.82}$$

Now, we consider the interference caused by the transmission from other requests. Suppose r_i is located in square $A^k_{a,b}$, since links are scheduled concurrently iff their receivers reside in the square

with the same color, the interference can only be caused by the senders whose receivers are in $A^k_{a\pm 2q,b\pm 2q}$, $A^k_{a\pm 2q,b\mp 2q}$, $A^k_{a,b\pm 2q}$, and $A^k_{a\pm 2q,b}$, where $q \in N$ (see Fig. 5.5 (b)). We represent the set of all active links whose receivers are in the $8q$ squares by $\mathcal{Q}^k_q$. For any link $l \in \mathcal{Q}^k_q$, because the distance between r_i and l's sender is at least $2q\beta_k - 2^{h_k+1}\sigma$, the RI of l on r_i is at most

$$\begin{aligned} RI_l(r_i, \mathcal{I}_i) &\leq \frac{P \times (2q\beta_k - 2^{h_k+1}\sigma)^{-\alpha}}{P_{\mathcal{I}_i, r_i}} \cdot \gamma_{\text{th}} && (5.83)\\ &\leq \frac{(2q\beta_k - 2^{h_k+1}\sigma)^{-\alpha}}{2^{-\alpha(h_k+1)}\sigma^{-\alpha}} \cdot \gamma_{\text{th}} && (5.84)\\ &= (2q\beta - 1)^{-\alpha} \cdot \gamma_{\text{th}}. && (5.85) \end{aligned}$$

Since there are at most $8q\Delta$ links in $\mathcal{Q}^k_q$, the RI of $\mathcal{Q}^k_q$ on r_i is upper bounded by

$$\begin{aligned} RI_{\mathcal{Q}^k_q}(r_i, \mathcal{I}_i) &= \sum_{l \in \mathcal{Q}^k_q} RI_l(r_i, \mathcal{I}_i) && (5.86)\\ &\leq \frac{8q\Delta \cdot \gamma_{\text{th}}}{(2q\beta - 1)^{\alpha}}, && (5.87) \end{aligned}$$

and the RI of all active links $\mathcal{Q}^k = \cup_q \mathcal{Q}^k_q$ on r_i is upper bounded by

$$\begin{aligned} RI_{\mathcal{Q}^k}(r_i, \mathcal{I}_i) &= \sum_{q=1}^{\infty} RI_{\mathcal{Q}^k_q}(r_i, \mathcal{I}_i) \quad \text{(By Lemma 5.2.1)} && (5.88)\\ &\leq \sum_{q=1}^{\infty} \frac{8q\Delta\gamma_{\text{th}}}{(2q\beta - 1)^{\alpha}} && (5.89)\\ &\leq \sum_{q=1}^{\infty} \frac{8q\Delta\gamma_{\text{th}}}{q^{\alpha}\beta^{\alpha}} && (5.90)\\ &\leq \frac{8\Delta\gamma_{\text{th}}}{\beta^{\alpha}} \cdot \frac{\alpha - 1}{\alpha - 2} = 1, && (5.91) \end{aligned}$$

which implies that r_i can be informed.

Now, we turn our attention to the approximation ratio of LLD-CLS (Theorem 5.2.2). To prepare the proof of Theorem 5.2.2, we first introduce the following two Lemmas. Table 5.1 lists some notations used in the proofs.

Lemma 5.2.2 *The number of time slots calculated by LLD-CLS, denoted by T_{lld}, is upper bounded by*

$$T_{\text{lld}} \leq 4 \cdot \rho(A^w_{\max}) \cdot g(\mathcal{K}). \quad (5.92)$$

Table 5.1: Notations.

Notation	Description
$\rho(A^k_{a,b})$	The receiver density of R_k in $A^k_{a,b}$.
$A^k_{\max}$	The square that has the highest $\rho(A^k_{a,b})$ in $\mathcal{A}^k$.
$A^w_{\max}$	The square that has the highest $\rho(A^k_{\max})$ over all $A^1_{\max}$, ..., $A^{g(\mathcal{K})+1}_{\max}$. Without loss of generality we assume that $A^w_{\max}$ is in $\mathcal{A}^w$.

Proof Our first observation is that when the link set L_k are scheduled (the loop in lines 4-8 in Algorithm 6), only the receivers in R_k are newly informed in this loop. Otherwise, the receiver must have been informed in some previous loop. It implies that there are at most $\rho(A^k_{\max})$ receivers required to be informed in each square in this loop. Then, the inner repeat loop (lines 7-10) can be repeated at most $\rho(A^k_{\max})$ times. Given that there are 4 colors and $g(\mathcal{K})$ link classes, the number of time slots T_{lld} used in this algorithm is upper bounded by

$$T_{\text{lld}} \leq \sum_{k=1}^{g(\mathcal{K})} 4 \cdot \rho(A^k_{\max}) \tag{5.93}$$

$$\leq 4 \cdot \rho(A^w_{\max}) \cdot g(\mathcal{K}). \tag{5.94}$$

Algorithm 6: Pseudo code for LLD-CLS.

```
input  : {L_1, ..., L_{g(K)}}, {R_1, ..., R_{g(K)}};
output: I_cls = {I^1, I^2, ...I^t};
t ← 0;
for k ← 1 to g(K) do
    Partition the region into squares A^k = {A^k_{a,b}} of size β_k × β_k;
    Color the squares with {1, 2, 3, 4} s.t. no two adjacent squares have the same color
    (see Fig. 5.5 (a));
    for j ← 1 to 4 do
        while R_k has receivers located in squares in j do
            t ← t + 1;
            for each square in j that has receivers in R_k do
                Pick one receiver r_i in the square;
                if |I_i| > Δ then
                    Add the shortest Δ links in I_i to I^t;
                else
                    Add all the links in I_i to I^t;
                Remove r_i from R_k;
return I_cls = {I^1, I^2, ...I^t};
```

Lemma 5.2.3 *Given a pair of receivers $r_1, r_2 \in R_k$ that are located in a square $A_{a,b}^k$. Represent the active desired link sets of r_1 and r_2 by $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively. The RI of $\mathcal{I}_2$ on r_1 is then lower bounded by*

$$RI_{\mathcal{I}_2}(r_1, \mathcal{I}_1) \geq \eta\gamma_{\text{th}} \cdot \frac{P_{\mathcal{I}_2,r_2}}{P_{\mathcal{I}_1,r_1}}, \tag{5.95}$$

where η is a constant $\eta = \left(1 + 2\sqrt{2}\beta\right)^{-\alpha}$, and $P_{\mathcal{I}_1,r_1}$ and $P_{\mathcal{I}_2,r_2}$ are the signal powers that r_1 and r_2 receive from their active desired link sets $\mathcal{I}_1$ and $\mathcal{I}_2$, respectively

Proof Because both r_1 and r_2 reside in the same square $A_{a,b}^k$, the distance between r_1 and r_2, denoted by d_{r_1,r_2}, is upper bounded by $\sqrt{2}\beta_k$. For any link $l_{s,r_2} \in \mathcal{I}_2$, we have $d_{s,r_1} \leq d(l_{s,r_2}) + d_{r_1,r_2}$ (triangular inequality) and $d(l_{s,r_2}) \geq 2^{h_k} \cdot \sigma$, then we can derive

$$\begin{aligned}
\frac{d(l_{s,r_1})^{-\alpha}}{d(l_{s,r_2})^{-\alpha}} &\geq \left(\frac{d(l_{s,r_2}) + d_{r_1,r_2}}{d(l_{s,r_2})}\right)^{-\alpha} && (5.96)\\
&= \left(1 + \frac{d_{r_1,r_2}}{d(l_{s,r_2})}\right)^{-\alpha} && (5.97)\\
&\geq \left(1 + \frac{\sqrt{2}\beta_k}{2^{h_k} \cdot \sigma}\right)^{-\alpha} && (5.98)\\
&= \left(1 + 2\sqrt{2}\beta\right)^{-\alpha}. && (5.99)
\end{aligned}$$

Hence, we can get that

$$\begin{aligned}
\frac{P_{\mathcal{I}_2,r_1}}{P_{\mathcal{I}_2,r_2}} &= \frac{\sum_{l_{s,r} \in \mathcal{I}_2} P \cdot d_{s,r_1}^{-\alpha}}{\sum_{l_{s,r_2} \in \mathcal{I}_2} P \cdot d(l_{s,r_2})^{-\alpha}} && (5.100)\\
&\geq \left(1 + 2\sqrt{2}\beta\right)^{-\alpha} && (5.101)\\
&= \eta.
\end{aligned}$$

Consequently, we can derive

$$\begin{aligned}
RI_{\mathcal{I}_2}(r_1, \mathcal{I}_1) &= \gamma_{\text{th}} \cdot \frac{P_{\mathcal{I}_2,r_1}}{P_{\mathcal{I}_1,r_1}} && (5.102)\\
&= \gamma_{\text{th}} \cdot \frac{P_{\mathcal{I}_2,r_1}}{P_{\mathcal{I}_2,r_2}} \cdot \frac{P_{\mathcal{I}_2,r_2}}{P_{\mathcal{I}_1,r_1}} && (5.103)\\
&\geq \eta\gamma_{\text{th}} \cdot \frac{P_{\mathcal{I}_2,r_2}}{P_{\mathcal{I}_1,r_1}}. && (5.104)
\end{aligned}$$

Theorem 5.2.2 *The approximation ratio of LLD-CLS is $O(g(\mathcal{K}))$.*

Proof We proceed by showing that an optimum solution OPT can inform all the receivers in R_w in $A^w_{\max}$ using at least $T_w = \lceil \rho(A^w_{\max})/m \rceil$ time slots, where m is a constant

$$m = \lceil (\eta\gamma_{\text{th}})^{-1} + 1 \rceil. \tag{5.105}$$

For the sake of contradiction, assume that OPT informs R_w using less than T_{opt} time slots. Therefore, there must exist a time slot t, $1 \le t \le T_w$, such that at least $m+1$ receivers in R_w located in $A^w_{\max}$ are informed simultaneously. Without loss of generality, let r_1, r_2,..., r_{m+1} be the $m+1$ receivers informed at this time slot, which have the active desired link sets $\mathcal{I}_1$, ..., $\mathcal{I}_{m+1}$, respectively, and let

$$P_{\mathcal{I}_1,r_1} = \min\{P_{\mathcal{I}_i,r_i} | k = 1, 2, ..., m+1\} \tag{5.106}$$

where $P_{\mathcal{I}_i,r_i}$ represents the signal power r_i receives from $\mathcal{I}_i$ $(i = 1, 2, ..., m+1)$. Hence, the RI of $\mathcal{I} = \cup_{i=2}^{m+1}\mathcal{I}_i$ on r_1 is given by

$$\begin{aligned} RI_{\mathcal{I}}(r_1, \mathcal{I}_1) &= \sum_{i=2}^{m+1} RI_{\mathcal{I}_i}(r_1, \mathcal{I}_1) \quad \text{(By Lemma 5.2.1)} && (5.107)\\ &\ge \sum_{i=2}^{m+1} \eta\gamma_{\text{th}} \cdot \frac{P_{\mathcal{I}_i,r_i}}{P_{\mathcal{I}_1,r_1}} \quad \text{(By Lemma 5.2.3)} && (5.108)\\ &\ge m\eta\gamma_{\text{th}} > 1 && (5.109) \end{aligned}$$

which implies r_1 cannot be informed. Hence, it needs at least $\lceil \rho(A^w_{\max}/m) \rceil$ time slots for OPT to inform all the receivers in R_w in $A^w_{\max}$. On the other hand, LLD-CLS can inform all receivers within $T_{\text{lld}} \le 4 \cdot \rho(A^w_{\max}) \cdot g(\mathcal{K})$ time slots (by Lemma 5.2.2). Therefore the approximation ratio follows

$$\frac{T_{\text{lld}}}{T_{\text{opt}}} \le \frac{T_{\text{lld}}}{T_w} \le 4m \cdot g(\mathcal{K}) = O(g(\mathcal{K})), \tag{5.110}$$

where T_{opt} denotes the number of time slots that the optimal solution OPT needs to inform all the receivers.

5.2.3 LLD Based Algorithm For OCLS

Similar to LLD-CLS, in the LLD based algorithm for OCLS (or LLD-OCLS for short), we construct $g(\mathcal{K})$ disjoint link classes $L_1, L_2, ..., L_{g(\mathcal{K})}$ according to Equ. (5.79) and schedule each link

class separately (Algorithm 7). For each link class L_k we partition the whole network region into a set of squares $\mathcal{A}^k = \{A^k_{a,b}\}$ and color these squares with 4 colors $j \in \{1,2,3,4\}$, where each square has size $\beta_k \times \beta_k$. Then, we pick up one receiver for each square of color j (if the square has receivers in R_k) and add the receiver's active desired link set to $\mathcal{I}(k,j)$. Note that if the size of desired link set is larger than Δ, we pick the shortest Δ links from the link set. Consequently, we can get $4g(\mathcal{K})$ feasible schedules: $\mathcal{I}(k,j)$ $(k = 1,...,N, j = 1,2,3,4)$. Finally, the schedule with most receivers informed is determined (line 12):

$$\mathcal{I}_{\text{ocls}} = \arg\max\{U(\mathcal{I}(k,j))|\mathcal{I}(k,j), \forall k,j\} \tag{5.111}$$

where $U(\mathcal{I}')$ denotes the number of receivers informed by link set $\mathcal{I}'$. Since we pick one receiver per selected square, the feasibility of the schedule constructed by Algorithm 7 has been proved in Theorem 5.2.1. In the next theorem, we calculate the approximation ratio of this algorithm.

Algorithm 7: Pseudo code for LLD-OCLS.

```
input  : {L_1, ..., L_{g(K)}}, {R_1, ..., R_{g(K)}};
output: I_ocls;
for k ← 1 to g(K) do
    Partition the region into squares A^k = {A^k_{a,b}} of size β_k × β_k;
    Color the squares with {1,2,3,4} s.t. no two adjacent squares have the same color
    (see Fig. 5.5 (a));
    for j ← 1 to 4 do
        for each square in j that has receivers in R_k do
            Pick one receiver r_i in the square;
            if |I_i| > Δ then
                Add the shortest Δ links in I_i to I(k,j);
            else
                Add all the links in I_i to I(k,j);
            Remove r_i from R_k;
I_ocls ← arg max{U(I(k,j))|I(k,j), ∀k,j};
return I_ocls;
```

Theorem 5.2.3 *The approximation ratio of LLD-OCLS is $O(g(\mathcal{K}))$.*

Proof We start the proof by defining $\mathcal{I}_{\max}(k)$ by

$$I_{\max}(k) = arg\max\{U(\mathcal{I}(k,j))|\mathcal{I}(k,j), j = 1,2,3,4\}. \tag{5.112}$$

Since Algorithm 7 returns the schedule of the maximum number of informed receivers over all length classes and colorings, the number of receivers informed by LLD-OCLS is given by

$$U_{\text{lld}} = \max\{U(\mathcal{I}_{\max}(k)), k = 1, 2, ..., g(\mathcal{K})\}. \tag{5.113}$$

We use U_{opt} to represent the number of receivers informed by the optimal solution OPT. Also, we use U_{opt}^k to denote the number of receivers in R_k informed by OPT. Then, we have

$$U_{\text{opt}} = \sum_{k=1}^{g(\mathcal{K})} U_{\text{opt}}^k. \tag{5.114}$$

In Theorem 5.2.2 we have showed that any feasible schedule can inform at most m (defined in Equ. (5.105)) receivers in each square in $\mathcal{A}^k$ at each time slot. Then, the following bound holds:

$$\frac{U_{\text{opt}}^k}{U(\mathcal{I}_{\max}(k))} \leq 4m. \tag{5.115}$$

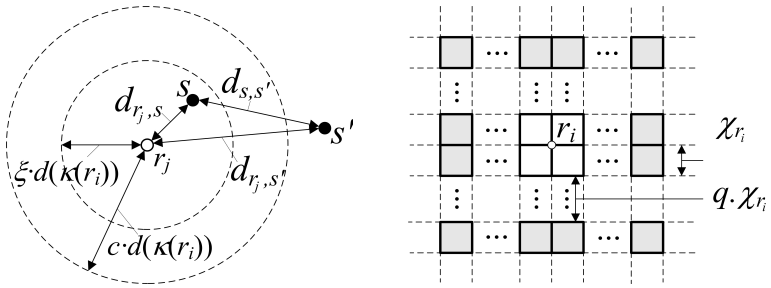
and the approximation ratio follows:

$$\frac{U_{\text{opt}}}{U_{\text{lld}}} = \sum_{k=1}^{g(\mathcal{K})} \frac{U_{\text{opt}}^k}{U_{\text{lld}}} \leq \sum_{k=1}^{g(\mathcal{K})} \frac{U_{\text{opt}}^k}{U(\mathcal{I}_{\max}(k))} \leq 4m \cdot g(\mathcal{K}) = O(g(\mathcal{K})). \tag{5.116}$$

5.2.4 A Greedy Algorithm For OCLS

In this section, we present a greedy algorithm (see Algorithm 8) for a special case of OCLS, in which the desired link set of each receiver is upper bounded by a constant Ω ($\Omega \geq 2$). For example, three-node model for CC [54] assumes that there are at most two senders for each receiver. In each iteration, the algorithm greedily selects the uninformed receiver with the shortest key link in $\mathcal{K}$, say r_i, and activates all the links with lengths no larger than $\xi \cdot d(\kappa(r_i))$ in I_i, where ξ is a constant set by the algorithm. To guarantee that r_i is informed, the algorithm deletes the links that may conflict with the selected links. First, all links whose senders are within the radius $c \cdot d(\kappa(r_i))$ of the receiver r_i are removed from L, where c is a constant

$$c = \sqrt{2}\left(\frac{10\Omega \cdot (\alpha - 1) \cdot \gamma_{\text{th}}}{\alpha - 2}\right)^{\frac{1}{\alpha}} + \xi. \tag{5.117}$$



(a) Proof in Lemma 5.2.4 (b) Proof in Theorem 5.2.4

Figure 5.6: Proof of the approximation ratio of the greedy algorithm.

Second, for any link set I_j, such that the RI of the selected links on r_j rose above 1/2, is removed. This process (lines 3-7) is repeated until all links in L have been either active or deleted. Next, we prove that the obtained schedule from the OCLS algorithm is both feasible (Theorem 5.2.4) and competitive, *i.e.*, is only a constant factor away from the optimum (Theorem 5.2.5).

Algorithm 8: Pseudo code for OCLS's greedy algorithm.

input : $L = \{I_1, ..., I_N\}$
output: $\mathcal{I}_{\text{ocls}}$
$\mathcal{I}_{\text{ocls}} \leftarrow \phi$;
while $L \neq \mathcal{I}_{\text{ocls}}$ **do**
  Pick up the receiver r_i that has the shortest link in L;
  Add the link set $\mathcal{I}_i = \{l \in I_i | d(l) < \xi \cdot d(\kappa(r_i))\}$ to $\mathcal{I}_{\text{ocls}}$;
  Remove $I_i \backslash \mathcal{I}_i$ from L;
  Remove all the links $l_{s,r}$, *s.t.* $d_{s,r} < c \cdot d(\kappa(r_i))$ from L;
  Remove any link set I_j, *s.t.* $RI_{\mathcal{I}_{\text{ocls}}}(r_j, I_j) > 1/2$;
return $\mathcal{I}_{\text{ocls}}$;

Let r_i be any receiver selected in Algorithm 8, which has active desired link set $\mathcal{I}_i$, and let $\mathcal{I}_i^-$ and $\mathcal{I}_i^+$ be the set of links added *after* and *before* $\mathcal{I}_i$, respectively.

Lemma 5.2.4 *The distance between the senders for different receivers in $\mathcal{I}_i^+$ is lower bounded by* $(c - \xi)d(\kappa(r_i))$.

Proof For any receiver r_j whose active desired links are in $\mathcal{I}_i^+$, there is no sender (in different request with r_j) in $\mathcal{I}_i^+$ that has distance smaller than $c \cdot d(\kappa(r_j))$ from r_j. Using this fact and the triangular inequality (see Fig. 5.6 (a)), we can lower bound the distance between two senders in different requests in $\mathcal{I}_i^+$

$$d_{s,s'} \quad \geq \quad d_{s',r_j} - d_{s,r_j} \tag{5.118}$$

$$\geq \quad d_{s',r_j} - \xi \cdot d(\kappa(r_j)) \tag{5.119}$$

$$\geq \quad c \cdot d(\kappa(r_j)) - \xi \cdot d(\kappa(r_j)) \tag{5.120}$$

$$\geq \quad (c-\xi) d(\kappa(r_i)). \tag{5.121}$$

Theorem 5.2.4 *(*Feasibility*) LLD-OCLS provide a feasible schedule.*

Proof When a link set $\mathcal{I}_i$ of r_i is added to the schedule, the RI of $\mathcal{I}_i^-$ on r_i must be no larger than 1/2; otherwise, it has already been deleted in a previous step. Therefore, the RI on r_i by concurrently active link set $\mathcal{I}_i^-$ is $RI_{\mathcal{I}_i^-}(r_i, \mathcal{I}_i) \leq 1/2$. It remains to show that $RI_{\mathcal{I}_i^+}(r_i, \mathcal{I}_i) \leq 1/2$. The transmission power received at r_i from its active link set $\mathcal{I}_i$ is at least

$$P_{\mathcal{I}_i, r_i} \geq \frac{P}{d(\kappa(r_i))^\alpha}. \tag{5.122}$$

We partition the whole network region into squares with size $\chi_{r_i} \times \chi_{r_i}$ (see Fig. 5.6 (b)), where

$$\chi_{r_i} = \frac{\sqrt{2}}{2}(c-\xi) d(\kappa(r_i)). \tag{5.123}$$

According to Lemma 5.2.4, any two senders for different receivers in $\mathcal{I}_i^+$ cannot be located in the same square. We use $\mathcal{Q}_q^i$ to denote the set of links whose senders are in the squares that are $q \cdot \chi_{r_i}$ away from r_i. Then, there are at most $4(q+1) \cdot \Omega$ links in $\mathcal{Q}_q^i$. The distance between the senders in $\mathcal{Q}_q^i$ and r_i is at least $q \cdot \chi_{r_i}$, so the RI of l on r_i is at most

$$RI_l(r_i, \mathcal{I}_i) \quad \leq \quad \frac{P \times \chi_{r_i}^{-\alpha}}{P_{\mathcal{I}_i, r_i}} \tag{5.124}$$

$$\leq \quad \frac{(q \cdot \frac{\sqrt{2}}{2}(c-\xi) d(\kappa(r_i)))^{-\alpha}}{d(\kappa(r_i))^{-\alpha}} \tag{5.125}$$

$$= \quad \left(q \cdot \frac{\sqrt{2}}{2}(c-\xi) \right)^{-\alpha}. \tag{5.126}$$

The RI of $\mathcal{Q}_q^i$ on r_i is then upper bounded by

$$RI_{\mathcal{Q}_q^i}(r, \mathcal{I}_i) \quad = \quad \sum_{l \in \mathcal{Q}_q^i} RI_l(r_i, \mathcal{I}_i) \leq \frac{4(q+1) \cdot \Omega}{(q \cdot \frac{\sqrt{2}}{2}(c-\xi))^\alpha}, \tag{5.127}$$

and the RI of all active links $\mathcal{I}_i^+ = \cup_q \mathcal{Q}_q^i$ on r_i is upper bounded by

$$\begin{aligned}
RI_{\mathcal{I}_i^+}(r_i, \mathcal{I}_i) &= \sum_{q=1}^{\infty} RI_{\mathcal{Q}_q^i}(r_i, \mathcal{I}_i) \quad \text{(By Lemma 5.2.1)} && (5.128)\\
&\leq \sum_{q=1}^{\infty} \frac{4(q+1)\Omega \cdot \gamma_{\text{th}}}{(q \cdot \frac{\sqrt{2}}{2}(c-\xi))^{\alpha}} && (5.129)\\
&\leq \sum_{q=1}^{\infty} \frac{5q\Omega \cdot \gamma_{\text{th}}}{(q \cdot \frac{\sqrt{2}}{2}(c-\xi))^{\alpha}} && (5.130)\\
&= \frac{5\Omega \cdot \gamma_{\text{th}}}{(\frac{\sqrt{2}}{2}(c-\xi))^{\alpha}} \frac{1}{q^{\alpha-1}} && (5.131)\\
&\leq \frac{5\Omega \cdot \gamma_{\text{th}}}{(\frac{\sqrt{2}}{2}(c-\xi))^{\alpha}} \cdot \frac{\alpha-1}{\alpha-2} && (5.132)\\
&= \frac{1}{2}. && (5.133)
\end{aligned}$$

which implies that r_i can be informed.

Lemma 5.2.5 *Let $\mathcal{I}_{\text{ocls}}$ be a feasible solution and let r_i be an informed receiver, which has key link l_{s,r_i}. Denote the active desired link set of r_i by $\mathcal{I}_i$. The number of senders in $\mathcal{I}_{\text{ocls}} \backslash \mathcal{I}_i$ with distance $k \cdot d(\kappa(r_i))$ from s is at most $(k+1)^{\alpha}\Omega/\gamma_{\text{th}}$.*

Proof The RI of each link $l_{s',r'} \in \mathcal{I}_{\text{ocls}} \backslash \mathcal{I}_i$ on r_i is lower bounded by

$$\begin{aligned}
RI_{l_{s',r'}}(r_i, \mathcal{I}_i) &= \frac{d_{s',r_i}^{-\alpha} \cdot \gamma_{\text{th}}}{\sum_{l \in \mathcal{I}_{\text{ocls}}} d(l)^{-\alpha}} && (5.134)\\
&\geq \frac{(d_{s,r_i} + d_{s',s})^{-\alpha} \cdot \gamma_{\text{th}}}{\sum_{l \in \mathcal{I}_{\text{ocls}}} d_{s,r_i}^{-\alpha}} && (5.135)\\
&\geq \frac{(d(l_{s,r}) + d_{s',s})^{-\alpha} \cdot \gamma_{\text{th}}}{|\mathcal{I}_i| \cdot d(l_{s,r})^{-\alpha}} && (5.136)\\
&= \frac{\gamma_{\text{th}}}{|\mathcal{I}_i|} \left(1 + \frac{d_{s,s'}}{d(l_{s,r})}\right)^{-\alpha} && (5.137)\\
&\geq \frac{(1+k)^{-\alpha}\gamma_{\text{th}}}{\Omega} && (5.138)
\end{aligned}$$

Since the RI of $\mathcal{I}_{\text{ocls}} \backslash \mathcal{I}_i$ on r_i cannot exceed one, there are at most $(k+1)^{\alpha} \cdot \Omega$ such senders senders with distance no larger than $k \cdot d(\kappa(r_i))$ from s.

Definition 5.2.4 *(*Blue and red points *[20]) Let $\mathcal{S}_r$ and $\mathcal{S}_b$ be two disjoint sets of points (*red *and* blue*) in a 2D Euclidean space. For any $z \in N$, a point $s_b \in \mathcal{S}_b$ is* z-blue-dominant *if every circle*

$B_\delta(s_b)$ around s_b, comprised by points p such that $d(p, s_b) \leq \delta$, contains z times more blue than red points, or formally

$$|B_\delta(s_b) \cap \mathcal{S}_b| > z \cdot |B_\delta(s_b) \cap \mathcal{S}_r| \quad \forall \delta \in R^+. \tag{5.139}$$

Lemma 5.2.6 *(*Blue-dominant centers lemma *[20]) For any $z \in N$, if $|S_b| > 5z \cdot |S_r|$, then there exists at least one z-blue-dominant point s_b in $\mathcal{S}_b$. In addition, given a z-blue-dominant point s_b, for each point s_r in $\mathcal{S}_r$, there exists a subset of $\mathcal{S}_b$ corresponding to s_r, denoted by $G(s_r)$,* s.t.*,*

- *any point in $G(s_r)$ is farther from s_r than from s_b: $\forall s \in G(s_r)$, $d_{s_r,s} > d_{s_b,s}$;*
- *for any pair of points $s_r, s'_r \in \mathcal{S}_r$, $G(s_r) \cap G(s'_r) = \phi$;*
- *the number of points in each subset $G(s_r)$ is no smaller than z: $|G(s_r)| \geq z$ $\forall s_r \in \mathcal{S}_r$.*

Proof See the proof in Lemma 4.4 in [20].

Lemma 5.2.7 *Denote the set of all senders in the optimal schedule and the greedy algorithm by $\mathcal{S}_{\text{opt}}$ and $\mathcal{S}_{\text{gre}}$, respectively. Then, $|\mathcal{S}_{\text{opt}} \backslash \mathcal{S}_{\text{gre}}| \leq 3^\alpha \times 5\Omega \cdot |\mathcal{S}_{\text{gre}}|$.*

Proof For the sake of contradiction, assume that $|\mathcal{S}_{\text{opt}} \backslash \mathcal{S}_{\text{gre}}| > 3^\alpha \cdot 5\Omega \times |\mathcal{S}_{\text{gre}}|$. Label the set of senders in $\mathcal{S}_{\text{opt}}$ by blue ($\mathcal{S}_b = \mathcal{S}_{\text{opt}}$) and $\mathcal{S}_{\text{gre}}$ by red ($\mathcal{S}_r = \mathcal{S}_{\text{gre}}$). By Lemma 5.2.6, there is a z-blue-dominant point (sender) $s^* \in \mathcal{S}_b$ with sender set $\mathcal{S}^*$, where $z = 3^\alpha \times \Omega$. We shall argue that the link l_{s^*,r^*} (or l^* for simplicity) would have been picked by our algorithm, which leads to a contradiction.

According to Lemma 5.2.6, for any red point $s_r \in \mathcal{S}_r$, there exists a subset of blue points $G(s_r)$ such that all the points in $G(s_r)$ are closer to s^* than to s_r and $|G(s_r)| \geq z$ ($z = 3^\alpha \times \Omega$). We can derive that $d_{s^*,s_r} > 2 \cdot d(l^*)$; otherwise, the number of senders within distance $2 \cdot d(l^*)$ from s^* would be larger than $(2+1)^\alpha \cdot \Omega \geq 3^\alpha \cdot |\mathcal{S}^*|$, which contradicts with the conclusion in Lemma 5.2.5. Based on triangle inequality, $d_{s_r,r^*} \geq d_{s^*,s_r} - d(l^*) > d_{s^*,s_r}/2$. Denote the sum signal power that r^* receives from $\mathcal{S}^*$ by P^*. The RI of the red sender s_r on r^* is then upper bounded by

$$RI_{s_r}(r^*, \mathcal{S}^*) \quad = \quad \frac{d_{s_r,r^*}^{-\alpha} P}{P^*} \cdot \gamma_{\text{th}} \leq \frac{d_{s_r,s^*}^{-\alpha} P}{2^{-\alpha} P^*} \cdot \gamma_{\text{th}}. \tag{5.140}$$

Also, for any point $s_b \in G(s_r)$,

$$d_{s_b,r^*} \quad \leq \quad d_{s_b,s^*} + d_{s^*,r^*} \tag{5.141}$$

$$< \quad d_{s_r,s^*} + d_{s^*,r^*} \tag{5.142}$$

$$< \quad d_{s_r,s^*} + \frac{d_{s_r,s^*}}{2} \tag{5.143}$$

$$= \quad \frac{3d_{s_r,s^*}}{2}. \tag{5.144}$$

Hence, the sum RI of the blue senders in $G(s_r)$ on r^* is lower bounded

$$\sum_{s_b \in G(s_r)} RI_{s_b}(r^*, \mathcal{S}^*) \quad = \quad \sum_{s_b \in G(s_r)} \frac{d_{s_b,r^*}^{-\alpha} P}{P^*} \cdot \gamma_{\text{th}} \tag{5.145}$$

$$> \quad 3^\alpha \cdot \Omega \cdot \left(\frac{3}{2}\right)^{-\alpha} \cdot \frac{d_{s_r,s^*}^{-\alpha} P}{P^*} \cdot \gamma_{\text{th}} \tag{5.146}$$

$$\geq \quad \Omega \cdot RI_{s_r}(r^*, \mathcal{S}^*). \tag{5.147}$$

This relationship holds for any $s_r \in \mathcal{S}_r$, and $G(s_r)$ and $G(s_r')$ are disjoint $\forall s_r, s_r' \in \mathcal{S}_r$, then the total RI that r^* receives from the senders in OPT (blue points) is at least Ω times as high as the RI it would receive from the senders in the greedy algorithm (red points). Because s^* is in $\mathcal{S}_b$, its RI on r is at most 1. Therefore, we have

$$RI_{\mathcal{S}_r}(r^*, \mathcal{S}^*) < \frac{1}{\Omega} \cdot RI_{\mathcal{S}_b}(r^*, \mathcal{S}^*) \leq \frac{1}{2}. \tag{5.148}$$

Since $RI_{\mathcal{S}_r}(r^*, \mathcal{S}^*)$ is less than 1/2, it would not have been deleted by the greedy algorithm, which establishes the contradiction.

Theorem 5.2.5 *The approximation ratio of the greedy algorithm is $O(1)$.*

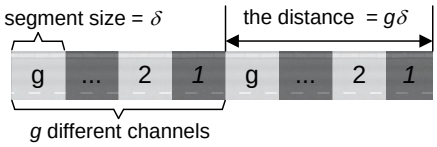
Proof Denote the number of receivers informed by the greedy algorithm and the optimal schedule by U_{gre} and U_{opt}, respectively. Then, according to Lemma 5.2.7,

$$\frac{U_{\text{opt}}}{U_{\text{gre}}} \quad \leq \quad \frac{\Omega \cdot |\mathcal{S}_{\text{opt}}|}{|\mathcal{S}_{\text{gre}}|} \tag{5.149}$$

$$= \quad \frac{\Omega \cdot (|\mathcal{S}_{\text{opt}} \backslash \mathcal{S}_{\text{gre}}| + |\mathcal{S}_{\text{gre}}|)}{|\mathcal{S}_{\text{gre}}|} \tag{5.150}$$

$$\leq \quad (3^\alpha \times 5\Omega + 1)\,\Omega \tag{5.151}$$

$$= \quad O(1). \tag{5.152}$$



(a) Proof in Lemma 5.2.4

Figure 5.7: Link scheduling.

5.3 Vehicle Link Scheduling

In this part, we introduce how to solve the vehicle link scheduling problem introduced in Section 3.2.2.3 using a greed method in the platoon network, called *Fast and Lightweight Autonomous link scheduling algorithm (FLA)*, where each vehicle autonomously determines the time slot for transmission solely based on its distance from the leader vehicle, without the need to collect the location information of other vehicles. The idea of FLA is based on the platoon feature that the distance between vehicles equals the safety distance. Based on this feature, we can conduct the link scheduling to ensure that the distance between vehicles using the same channel is lower bounded by the interference range. Also, each vehicle can autonomously decide which time slots it should use solely based on its distance from the leader vehicle, where the distance with the location of the leader vehicle is broadcasted from the leader vehicle. Here, the leader vehicle needs to periodically broadcast its location to the following vehicles. According to this information and its own location, each following vehicle can derive its distance from the leader vehicle. As Fig. 5.7 shows, we geometrically split the platoon to g segments with length equal to δ so that each segment contains at most one vehicle. Then, we consider every g consecutive segments as a group. Next, we allocate g channels to g segments in each group, and the vehicle in a segment chooses the channel allocated to this segment. As a result, at most one vehicle is contained in each segment and the vehicles sharing the same channel must have distance no less than the interference range, (i.e., $kg\delta$, $k = 1, 2, ...$), which avoids the interference. In the following, we will introduce how to determine g which is the minimum number of channels to avoid interference (Section 5.3.1) and how each node determines its channel in FLA (Section 5.3.2) in detail.

5.3.1 The Minimum Number of Time Slots

Now, we need to determine g, which denotes the least number of channels used to overcome interference. For any segment l, the distance between segment l and each segment that has the same channel as segment l is $kg\delta$ $(k = 1, 2, ...)$. If the distance between two segments is $kg\delta$, then the safety distance between the vehicles in the two segments is $kg\delta - \delta$, which implies that the interference generated from the vehicle in one segment to the vehicle in the other segment is at most $P(kg\delta - \delta)^{-\alpha}$. Consequently, the sum interference received by each vehicle is upper bounded by

$$\sum_{k=1}^{\infty} P\left(gk\delta - \delta\right)^{-\alpha} \leq P\sum_{k=1}^{\infty}\left((g-1)k\delta\right)^{-\alpha} \tag{5.153}$$

$$\leq P(g-1)^{-\alpha}\delta^{-\alpha}\sum_{k=1}^{\infty} k^{-\alpha} \tag{5.154}$$

$$= P(g-1)^{-\alpha}\delta^{-\alpha}\zeta(\alpha) \tag{5.155}$$

where $\zeta(\alpha) = \sum_{k=1}^{\infty} k^{-\alpha}$.

By definition, SIR is actually the quotient of the useful signal power to the sum interference. Because the distance between each pair of sender and receiver, say vehicles s_i and r_i, is upper bounded by the communication range R, the useful signal power received at vehicle r_i $(Pd_{s_i,r_i}^{-\alpha})$ is lower bounded by $PR^{-\alpha}$, i.e., $Pd_{s_i,r_i}^{-\alpha} \geq PR^{-\alpha}$. To guarantee that r_i can successfully receive a packet from s_i, we need to ensure that $SIR_{s_i,r_i} \geq \gamma_{\text{th}}$. That is, we need to find a channel for s_i to upper bound the sum interference from the senders with the same channel with s_i by $\frac{PR^{-\alpha}}{\gamma_{\text{th}}}$. Then, according to Equ. (5.153), we need to ensure

$$P(g-1)^{-\alpha}\delta^{-\alpha}\zeta(\alpha) \leq \frac{PR^{-\alpha}}{\gamma_{\text{th}}} \tag{5.156}$$

from which we derive that

$$g \geq \left\lceil \left(R^{\alpha}\delta^{-\alpha}\zeta(\alpha)\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} + 1 \right\rceil \tag{5.157}$$

We hope we can use as smaller number of channels as possible, then:

$$g = \left\lceil \left(R^{\alpha}\delta^{-\alpha}\zeta(\alpha)\gamma_{\text{th}}\right)^{\frac{1}{\alpha}} + 1 \right\rceil \tag{5.158}$$

That is, g can be pre-defined based on the transmission range of vehicles (R), path loss exponent

(α), decoding threshold γ_{th}, and segment distance δ.

Theorem 5.3.1 *(Feasibility) By setting the number of channels g by Equ. (5.158), the SIR received by each receivers is higher than the decoding threshold* γ_{th}.

Proof Without loss of generality, we examine any vehicle receiver r_i of which $(s_i, r_i) \in L_k$. Because $(s_i, r_i) \in L_k$, $2^{k-1}\delta \leq d_{s_i,r_i} < 2^k\delta$, the signal power received at r_i from its desired sender s_i is at least

$$P_{s_i,r_i} \geq \frac{P}{2^{\alpha k}\delta^{\alpha}}. \tag{5.159}$$

Now we consider the interference caused by the transmission from other requests. Suppose r_i is located in square Seg_m^k, since links are scheduled concurrently iff their receivers reside in the segment with the same color, the interference can only be caused by the links whose receivers are in $\text{Seg}_{m\pm 2q}^k$, where $q = 1, 2, 3, ...$ We represent the set of links whose receivers are in the two segments by $\mathcal{Q}_q^k$. For any link $(s_j, r_j) \in \mathcal{Q}_q^k$, because the distance between r_i and s_j is at least $(2q(g_k - 1) - 2^k)\delta$, the useful signal on r_i is at most

$$P_{s_i,r_i} \leq \frac{P}{(2q(g_k - 1) - 2^k)^{\alpha}\delta^{\alpha}}. \tag{5.160}$$

and

$$\begin{aligned}
\sum_{s_j:(s_j,r_j)\in L_k\setminus(s_i,r_i)} P_{s_j,r_i} &= \sum_{q=1}^{\infty} \sum_{j:(s_j,r_j)\in\mathcal{Q}_q^k} P_{s_j,r_i} && (5.161)\\
&= \sum_{q=1}^{\infty} \frac{2P}{(2q(g_k - 1) - 2^k)^{\alpha}\delta^{\alpha}} && (5.162)\\
&\leq \sum_{q=1}^{\infty} \frac{2P}{q^{\alpha}(g_k - 1)^{\alpha}\delta^{\alpha}} && (5.163)\\
&= \frac{2P\zeta(\alpha)}{(g_k - 1)^{\alpha}\delta^{\alpha}}. && (5.164)
\end{aligned}$$

Then,

$$\begin{aligned}
SIR_{s_i,r_i} &= \frac{P_{s_i,r_i}}{\sum_{s_j:(s_j,r_j)\in L_k\setminus(s_i,r_i)} P_{s_j,r_i}} && (5.165)\\
&\geq \frac{(g_k - 1)^{\alpha}}{2^{\alpha k+1}\zeta(\alpha)} \geq \gamma_{\text{th}}. && (5.166)
\end{aligned}$$

which implies that r_i can successfully receive the packet.

Theorem 5.3.2 *The approximation ratio of FLA is $O(gK)$, where $g = \max_{k=1,\ldots,K}\{g_k\}$.*

Proof Now we show that any optimal solution needs at least $\frac{T_{\text{FLA}}}{gu}$ channels, where u is a constant:

$$u = \left\lceil \frac{2^{\alpha}}{\gamma_{\text{th}}} \right\rceil \tag{5.167}$$

and T_{FLA} is the number of channels required by the FLA algorithm. For the sake of contradiction, we assume that the optimal solution uses less than $ug \cdot T_{\text{FLA}}$ channels. Then, there must exists a segment containing more than u receivers in the same link class in one channel. We pick any link receiver r_i in the same segment and the same channel, and calculate its SIR:

$$\frac{d_{s_i,r_i}^{-\alpha}}{\sum_{s_j:(s_j,r_j)\in L_k\setminus(s_i,r_i)} d_{s_j,r_i}^{-\alpha}} \leq \frac{\left(2^{h_k}\delta\right)^{-\alpha}}{u\left(2^{h_k+1}\delta\right)^{-\alpha}} = \frac{1}{u2^{-\alpha}} < \gamma_{\text{th}} \tag{5.168}$$

which is a contradiction.

5.3.2 Autonomous Channel Determination

In one segment group, as shown in Figure 5.7, each segment has a segment ID ranging from 1 to g. Vehicles in the segment with ID i choose to use channel i among g channels. Below, we introduce our FLA algorithm in which each vehicle autonomously determines its segment ID and then the channel to use.

Definition 5.3.1 *(Distance offset). The distance offset of a follower vehicle receiver r_i, denoted by Δ_i, is defined as the remainder of its distance from the leader vehicle (r_1) divided by $g\delta$:*

$$\Delta_i = d_{r_i,r_1} \mod g\delta \tag{5.169}$$

Property 5.3.1 *Given the distance offset of a receiver r_i, Δ_i, the segment ID of this vehicle is $\left\lceil \frac{\Delta_i}{g\delta} \right\rceil$.*

Table 5.2: The FLA table.

Δ_i	$[0,\delta)$	$[\delta,2\delta)$	...	$[(k-1)\delta,k\delta)$
Channel	1	2	...	g

According to Property 5.3.1, each vehicle's distance offset determines its segment ID, and then determines its channel. Hence, we can build a table (Table 5.2), namely the *FLA* table, which

associates each distance offset with each channel in g channels. A vehicle receives this table from its preceding vehicle after it joins the platoon. This table is kept in each vehicle's storage. Since the partition is static over time, once the table is built, each vehicle does not need to change the FLA table anymore. Using the FLA table, each vehicle only needs to know its distance from the leader vehicle to determine its channel without the need to collect location information of other follower vehicles. To let all the follower vehicles know the leader vehicle's location, the leader vehicle's current location is periodically propagated to all the follower vehicles by piggybacking the location information on the packet periodically sent from a preceding vehicle to its succeeding vehicle. According to the leader vehicle's location, each follower vehicle can calculate its distance from the leader vehicle.

To implement FLA, each vehicle only needs to calculate the distance offset based on its current distance from the leader vehicle by Equ. (5.169). Then, it checks the FLA table by the calculated distance offset, and finds the corresponding channel. For example, suppose the safety distance is 30 meters, and the number of channels is $g = 5$. Vehicle i estimates that the distance between the leader vehicle and itself is 195 meters. Using Equ. (5.169), vehicle i's distance offset equals $195 \mod (30 \times 5) = 45$ meters. Since $45 \in [30, 60)$, it chooses channel 2 based on the FLA table.

Chapter 6

Performance Evaluation

In this chapter, we will evaluate the performance of our relay selection methods (Section 6.1) and link scheduling methods (Section 6.2).

6.1 Relay Selection

This section presents the experimental results of the relay selection methods on NESTbed [12] and simulation with MATLAB. We compared CEDAR with the global optimal solution (OPTIMAL), the traditional link-layer protocols, where packet is decoded hop by hop [43, 55] (HBH), and with another solution where packet is only decoded at the destination (DEST). In order to evaluate the effect of the load balancing algorithm (Algorithm 2) in CEDAR, we also test the performance of CEDAR without this algorithm denoted by CEDAR*. We measured the following metrics: 1) *Packet delay*: the time interval from the time a packet is generated by the source node to the time that the packet is successfully decoded at the destination node. 2) *Throughput*: the total number of data bits successfully decoded at the destinations per time unit (msec) in the entire network. 3) *En/Decoding load*: the number of packets a node en/decodes. 4) *Number of probing messages*: the total number of messages used for probing.

6.1.1 Experiments on Real-World NESTbed

NESTbed is an open testbed for developing wireless sensor systems [12]. It is a collection of 80 TELOSB sensors that are arranged in a grid. The sensors have CC2420 Chip and communicate

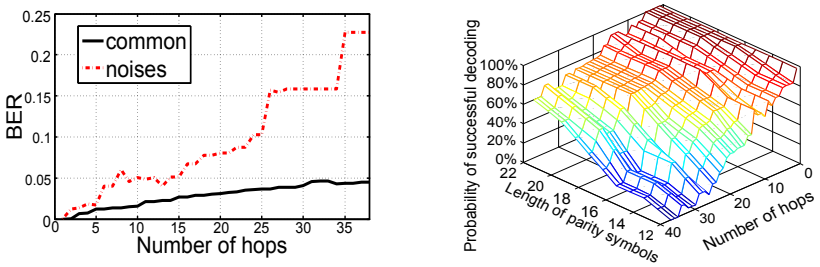
using the IEEE 802.15.4 standard. We verify our mathematical models and evaluate the performance of CEDAR on NESTbed. We created a multi-hop network of TELOSB sensors running Tiny-OS 2.1.0 written in NESC. We use Reed-Solomon codes to detect and fix errors in a packet, if the number of error bits exceeds the capability of Reed-Solomon codes for correction, the receiver ask for retransmission.

Mathematical model verification. We measured BER after the packet traveled for different numbers of hops in two scenarios, *common* and *noises*. Fig. 6.1 (a) shows BER versus the number of hops. We see that BER increases as the number of hops increases in both scenarios. This is because as the number of hops increases, more flipped bits are generated and errors tend to be cumulative and propagated along a routing path in the multi-hop network. Also, *common* produces much lower BER than *noises* as more noises increase BER.

We then measured the probability of successful decoding when the error correction node (responsible of Error Correction Code (ECC)) is away from the source node by different number of hops and when the parity symbols have different lengths. Fig. 6.1 (b) shows the probability of successful decoding in a multi-hop network with varying parity symbol lengths and the number of hops between the error correction node and the source node. The figure illustrates that if the error correction node is further away from the source, the probability of a packet drop increases. Also, when the length of parity symbols increases, the probability of successful decoding increases, which is consistent with Formula (4.2) (Fig. (a)). Further, when the number of hops is large, increasing the length of parity symbols will not guarantee the successful decoding of a packet. The experimental results indicate that in order to maintain a high throughput and low packet drop rate, it is necessary to fix errors in the packet as soon as possible. Thus, arranging the destination to decode the packets is not an effective method. However, performing retransmission or ECC at every hop is not efficient due to queuing delays. An effective method to decode packet is to fix errors in the intermediate nodes, which can successfully decode the packet while increasing the efficiency.

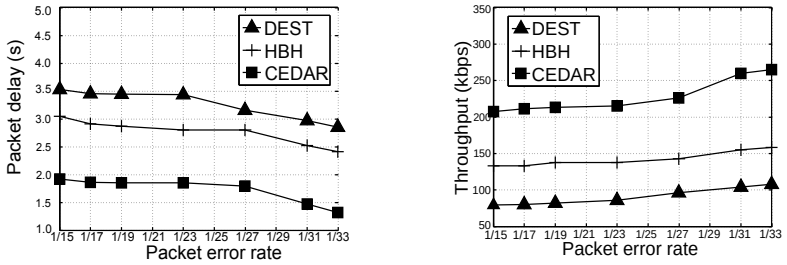
Scheme performance evaluation. In the experiments on NESTbed, we chose 8 sources and 8 destinations and the source-destination path length was 75 hops. Given that data rate $R = 250$kbps, average BER ≈ 0.001 (common scenario in Fig. 6.1 (a)), $L_{\text{pac}} = 50$ bytes, $L_{\text{ACK}} = 28$ bytes, we calculate the optimal number of hops one packet should be decoded once is 9 and 10.

Each source node generated 500 packets at a time interval varying from 80ms to 160ms. We measured the delay of transmissions and the throughput, which is defined as the total size of



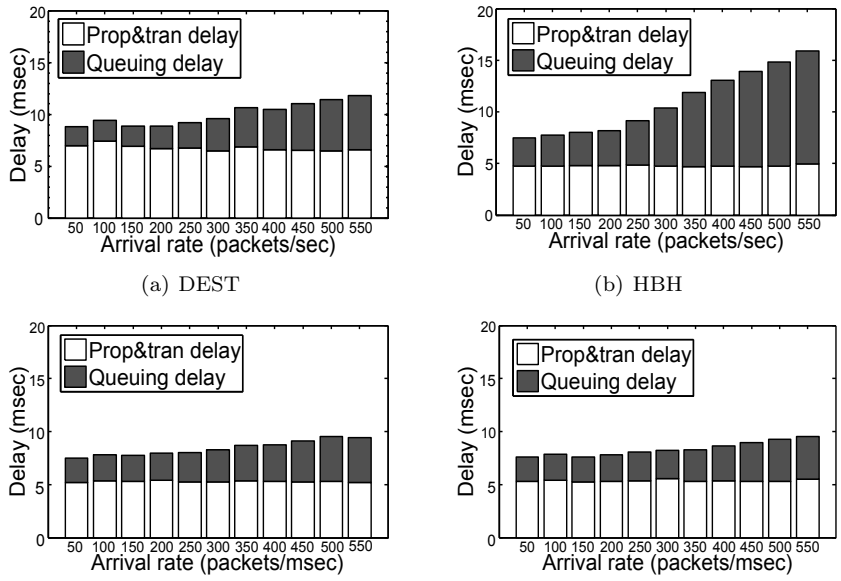
(a) BER in multiple hops (b) Successful decoding rate

Figure 6.1: Comparison of real-world NESTbed results.



(a) Packet delay (b) Throughput

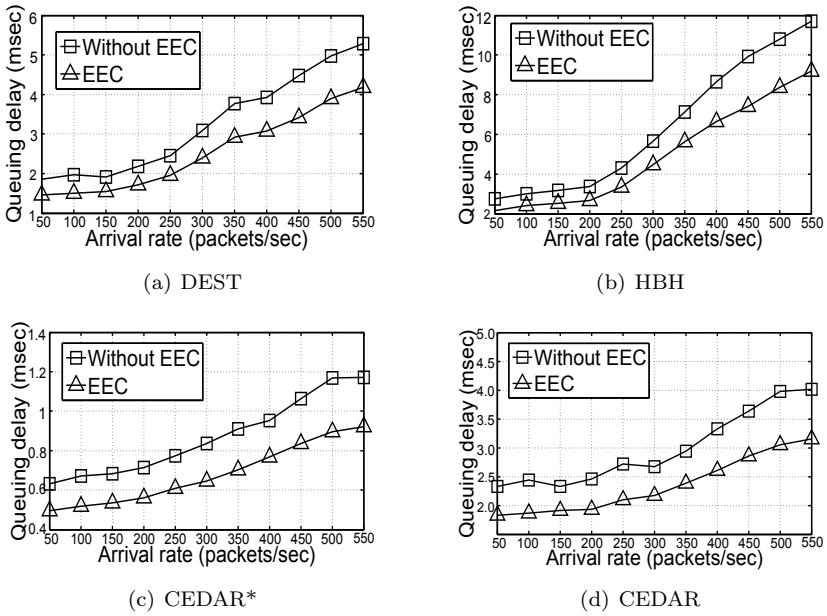
Figure 6.2: Experimental results on real-world NESTbed.



(a) DEST (b) HBH

(c) CEDAR* (d) CEDAR

Figure 6.3: Comparing prop&tran delay and queuing delay.



(a) DEST (b) HBH

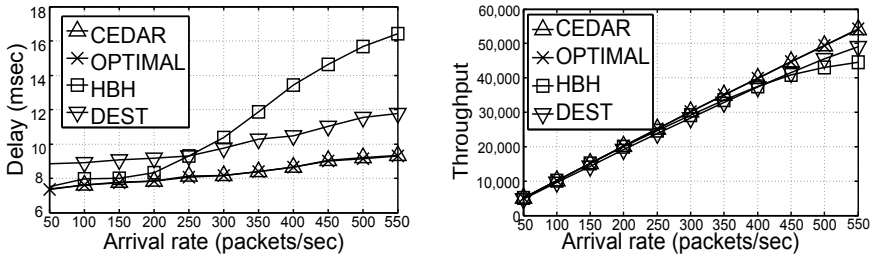
(c) CEDAR* (d) CEDAR

Figure 6.4: Queuing delay with and without using EEC.

all the packets divided by the total time used for transmitting all the packets. The packet error rate is defined as the average percent of unsuccessfully transmitted packets in each hop. To test the performance of the three schemes in different environments, we manually changed the packet error rate from 1/15 to 1/33. Fig. 6.2 (a) shows the packet delay of CEDAR, DEST and HBH. We find that the average packet delay of CEDAR is much lower than that of DEST and HBH. DEST has the highest delay because it assigns the decoding work to each packet's destination rather than the intermediate nodes, which generates much higher probability of packet re-decoding due to higher probability of packet errors, thus increasing the delay. The delay of HBH is higher than that of CEDAR because HBH requires packets to be en/decoded in each hop, which generates high en/decoding load on intermediate nodes, leading to high queuing delay. Fig. 6.2 (b) shows the throughput of three schemes. From the figure, we can find that the throughput follows CEDAR > HBH > DEST. This is because lower packet transmission delay usually leads to higher throughput in the network.

6.1.2 Simulation on Matlab

We conducted simulation on Matlab to evaluate the performance of CEDAR. We built a 9×9 grid network with each node located in one grid and randomly selected 16 pairs of source node



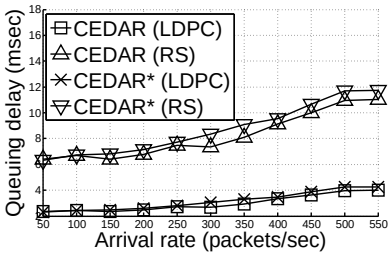
(a) Average packet delay (b) Throughput

Figure 6.5: Packet delay and throughput (with OPTIMAL).

and destination node. Each packet contains 20 data symbols, 5 type-I parity symbols and 5 type-II parity symbols. Each symbol has 5 bits. For the scheme using EEC code, there are $\lfloor \log_2 30 \rfloor = 4$ levels of EEC bits in each packet, with 5 EEC bits in each level. Also, we randomly chose nodes connecting each pair of source node and destination node as the route. We use Low Density Parity Check (LDPC) code [14] for en/decoding packets.

Fig. 6.3 (a), (b), (c) and (d) compare prop&tran delay and queuing delay computed by HBH, DEST, CEDAR* and CEDAR respectively. From the figures we can find that: (1) the queuing delay increases as the generating rate of each data stream increases but the prop&tran delay remains nearly constant; (2) the queuing delay increases more significantly in HBH than in DEST and CEDAR (i.e., it follows CEDAR < DEST < HBH); (3) for prop&tran delay it follows HBH < CEDAR < DEST, (4) CEDAR generates the same prop&tran delay but lower queuing delay than CEDAR*, and (5) the total packet delay follows CEDAR < CEDAR* < DEST < HBH. For (1), this is because queuing delay is determined by the generating rate of the source node but the prop&tran delay is independent of it. For (2), (4) and (5), HBH has higher queuing delay since it generates more en/decoding load on intermediate nodes. In contrast to HBH, DEST only assigns the decoding work to each packet's destination, which increases both prop&tran delay and queuing delay due to higher probability of packet redecoding (as Equ. (4.2) shows). Instead of accumulating decoding work on the destinations, CEDAR* and CEDAR chooses a number of intermediate nodes to be responsible for the en/decoding work to reduce the probability of redecoding. CEDAR performs better than CEDAR* because CEDAR distributes the en/decoding load of the intermediate nodes more evenly, which reduces the queuing delay as indicated in Proposition 4.3.1.

Fig. 6.4 (a), (b), (c) and (d) compare the queuing delays of the four schemes with and without using EEC versus different packet arrival rates. From the figures, we see that using EEC, the

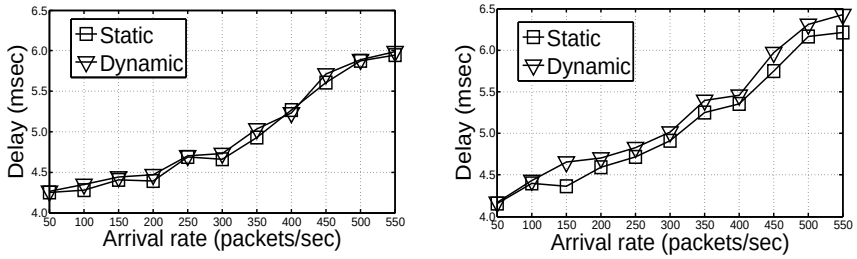


(a) CEDAR

Figure 6.6: RS vs. LDPC.

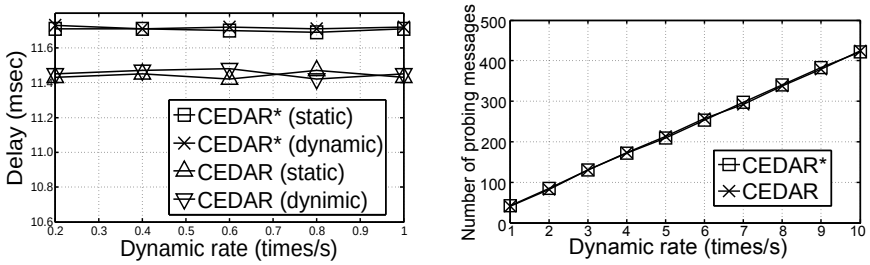
queuing delays of the four schemes are reduced significantly. This result is caused by two reasons. First, EEC decreases the number of retransmissions by estimating BER of the packets. Second, the computing time of EEC is much less than erasure code (e.g., Reed-Solomon code [9]) according to Equ. (4.24) and Equ. (4.26). These experimental results confirm the effectiveness of EEC in enhancing the performance of CEDAR. Fig. 6.5 (a) and (b) compare OPTIMAL with CEDAR, CEDAR*, DEST and HBH in terms of *packet delay* and *throughput*. Considering NP-hard feature of the problem, we only set a small scale network (6 source nodes and 6 destination nodes). The results demonstrate that CEDAR can achieve almost the "best" performance in terms of packet delay even in the distributed manner. Fig 6.6 compares the queuing delay of CEDAR and CEDAR* using Low Density Parity Check (LDPC) code and Reed-Solomon code [14], which is also a widely used code for HARQ protocol. From the figures, we find that the schemes using LDPC has much smaller queuing delay than those using Reed-Solomon code. Though LDPC has a chance of failure, LDPC is much faster than Reed-Solomon code for en/decoding packet (in our simulation on Matlab, to en/decoding a packet, LDPC code is about 1.8 times faster than Reed-Solomon code). Accordingly, the en/decoding time (service time) for LDPC is smaller, which leads the queuing delay of LDPC is smaller (according to Equ. (4.24)).

Fig. 6.7 (a) and (b) compare the packet delay of CEDAR* and CEDAR in the static and dynamic scenarios, respectively. In the dynamic scenario, due to the network topology change, the nodes in a route of each sender were changed to other randomly selected nodes once per second, i.e., dynamic rate was set to 1 times/s. The sender probing frequency was equal to the dynamic rate. From both figures, we find that CEDAR* and CEDAR achieve almost the same performance in static scenario and dynamic scenario. This is because that in both CEDAR* and CEDAR, the senders periodically send the packet to probe the information through the routes, so they always



(a) CEDAR (b) CEDAR*

Figure 6.7: Effect of dynamics with different arrival rates.



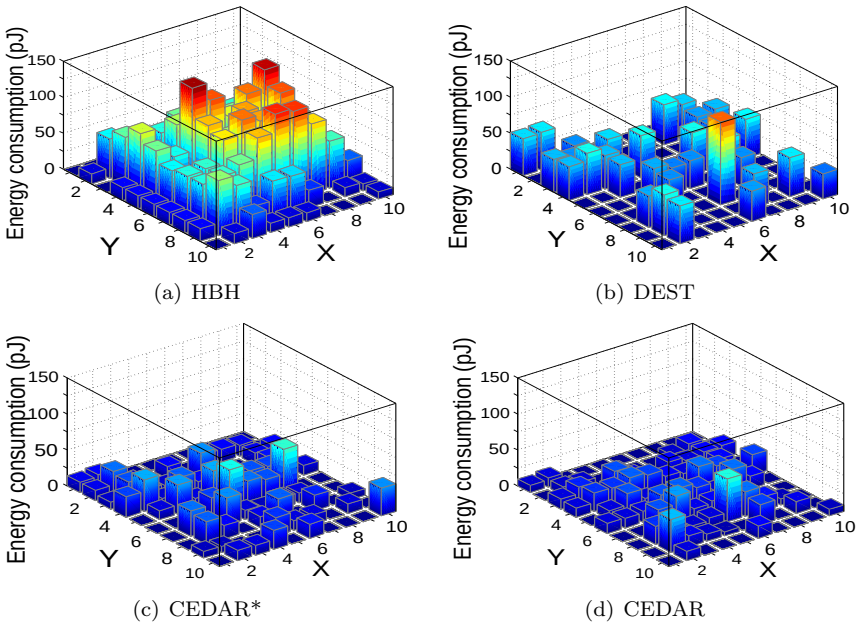
(a) Delay (b) Number of probing messages

Figure 6.8: Effect of dynamics with different dynamic rates.

know the most updated information to determine the en/decoding routing nodes even in the dynamic network. Fig. 6.8 (a) and (b) show the packet delay and the number of probing messages versus the dynamic rate, respectively. From Fig. 6.8 (a), we find that the packet delays of both CEDAR and CEDAR* remain at the same level as the dynamic rate increases. Fig. 6.8 (b) shows that the number of probing messages increases as the dynamic rate increases, since each sender must update information more frequently when the dynamic rate becomes higher.

6.1.3 Load Balancing

Fig. 6.9 (a), (b), (c) and (d) show the en/decoding load of each node in HBH, DEST, CEDAR* and CEDAR, respectively. The coordinate (x, y) $(1 \leq x \leq 9, 1 \leq y \leq 9)$ in the figures shows the position of each node in the network grid. We observe that most nodes in HBH have high decoding load, and the decoding loads among nodes vary greatly. DEST generates lower decoding load on nodes and move balance load distribution than HBH. Most nodes in CEDAR have much lower decoding load than nodes in HBH and DEST. From our experimental result data, we find that the average en/decoding load of HBH, DEST and CEDAR per node is approximately 90, 32, 24 and

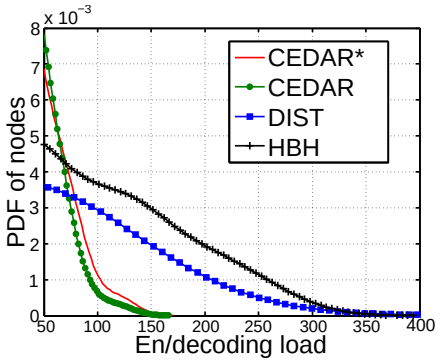


(a) HBH (b) DEST

(c) CEDAR* (d) CEDAR

Figure 6.9: En/decoding load distribution of all the nodes.

24. (HBH > DEST > CEDAR* ≈ CEDAR), respectively, and their standard deviation are 78, 49, 28 and 24 (HBH > DEST > CEDAR* > CEDAR), respectively. Thus, CEDAR can better balance the en/decoding load among nodes compared to the other two schemes and CEDAR*, and also decreases the average decoding load on a node compared to DEST and HBH. The better performance of CEDAR* than HBH and DEST confirms the effectiveness of CEDAR's intermediate node selection algorithm in distributing load among partial intermediate nodes rather than all intermediate nodes or only the destination nodes. The better performance of CEDAR over CEDAR* confirms the effectiveness of CEDAR's load balancing algorithm in balance the en/decoding load among nodes and hence reducing the queuing delay. Unlike HBH that arranges every node through a route to conduce en/decoding regardless of its current en/decoding load, DEST only assigns the destination for decoding, thus reducing decoding load on nodes. CEDAR only arranges the intermediate nodes in a route that have available en/decoding capacity to be decoders, which constrains the en/decoding load of intermediate nodes. CEDAR performs better than DEST in terms of the assigned decoding load per node and load balance. Recall that Equation (4.3) indicates the probability of successful decoding decreases very rapidly if the number of hops between a decoding node and its previous decoding node increases. More decoding failures for a packet in DEST lead to more decoding operations hence higher decoding load. Unlike DEST that biases on the destinations for the decoding



(a) CEDAR

Figure 6.10: Distribution.

operations, CEDAR distributes the decoding load among intermediate nodes that have available capacity for decoding, achieving more balanced load distribution. Actually, CEDAR overcomes the shortcomings of HBH and DEST by decreasing the re-decoding operations via assigning intermediate nodes as en/decoding nodes, and decrease the traffic jam by only selecting the intermediate nodes that have capacity of en/decoding.

Fig. 6.10 shows the Probability Density Function (PDF) of nodes based on en/decoding load in CEDAR, DEST, CEDAR* and HBH. We see that the en/decoding load of nodes in HBH is higher and more unevenly distributed than DEST and CEDAR. The reason for HBH's inferior performance is that it makes all the intermediate nodes be responsible for en/decoding all the packets going through them, which generates high en/decoding load on these nodes and also greatly increases the en/decoding load in the network. Though the mean value of the en/decoding load of DEST is not very high, the load among nodes is not distributed in balance and some nodes have en/decoding load larger than 250. By always conducting decoding in the destination nodes that are randomly selected from the network, DEST distributes the load more evenly than HBH. However, the longer distance between the decoding node and encoding node increases the probability of packet errors and recovery failures, leading to more re-decoding operations. Therefore, DEST generates more en/decoding load than CEDAR.

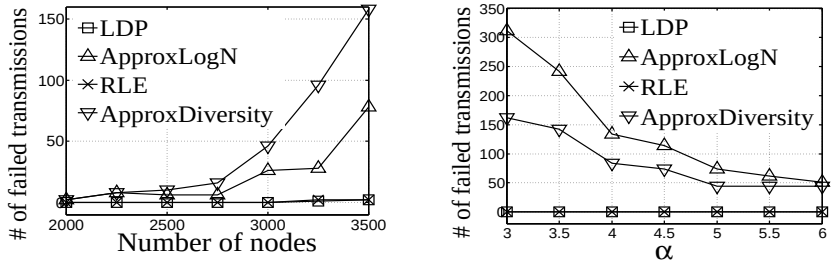
6.2 Link Scheduling

6.2.1 Fading Resistant Link Scheduling

In this section, we present experimental results to better illustrate the practical appeal of the fading resistant scheduling algorithms. In the experiment, each sender was given a random location in a 500×500 square, and each receiver was located from its sender with a distance randomly and uniformly selected from $[5, 20]$ in a random direction. The accepted error rate was set to 0.01, the decoding threshold was set to 1, and the data rate of every link was set to 1. We measured the following two metrics: (1) *throughput* (or the total data rate successfully received by receivers) and (2) *the number of failed transmissions*. We compare our algorithms with two other link scheduling algorithms: ApproxLogN [18] and ApproxDiversity [20]. ApproxLogN partitions the link set into disjoint link classes and schedules the links in each class separately. ApproxDiversity always picks up the shortest link and excludes links conflicted with the picked links in each iteration. Unlike our algorithms, ApproxLogN and ApproxDiversity are not fading-resistant although they are also polynomial time algorithms based on the SIR model.

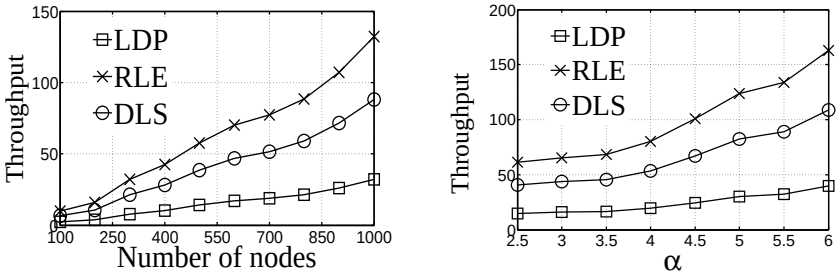
Fig. 6.11(a) and Fig. 6.11(b) show the number of failed transmissions of different algorithms versus the number of links and path loss exponent (α), respectively. Here, we call a transmission is failed if the received SIR is lower than the decoding threshold. We see that LDP and RLE have almost no failed transmissions, because they always select the links that can guarantee successful transmissions with high probability $1 - \varepsilon$ with fading consideration. ApproxLogN and ApproxDiversity assume that the channel is non-fading, which makes them fading-susceptible. Fig. 6.11(a) shows that the number of failed transmissions increases as the number of nodes increases. This is because more nodes cause more transmissions hence severer interference, thus increasing the probability of a transmission failure. An interesting observation from Fig. 6.11(b) is that the number of failed transmissions decreases as α increases. This is because when fading is more severe, the interference factors from all undesired remote nodes are smaller (by Formula (3.33)), which reduces the probability of a transmission failure.

We then measure the performance of the decentralized algorithm DLS and centralized algorithms LDP and RLE. In DLS, we randomly selected 6% links, and let each link conduct transmissions for 200 times. Fig. 6.12 shows that the throughput follows RLE>DLS>LDP with different number of links or different α values. This is because decentralized DLS makes scheduling decision



(a) Number of nodes (b) Path loss exponent

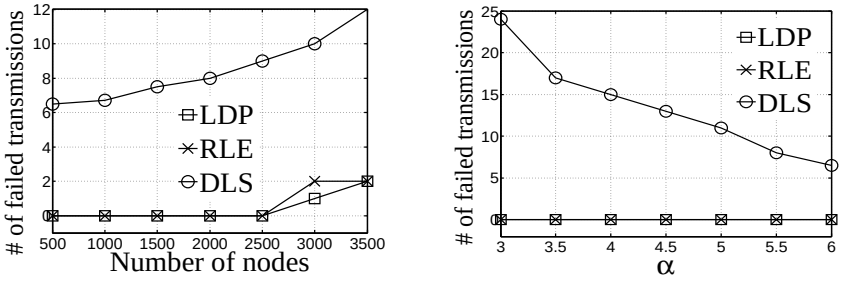
Figure 6.11: Fading-resistant vs. fading-susceptible algorithms: the number of failed transmissions



(a) Number of nodes (b) Path loss exponent

Figure 6.12: Centralized vs. decentralized algorithms: the number of links scheduled

based on local information while RLE is based on all system information. Though LDP is a centralized algorithm, only the links in the same class with the same color can be scheduled at the same time. Though such a mechanism can prevent the conflict among the links, it reduces the number of links that can be scheduled simultaneously. Fig, 6.12(a) shows that the throughput increases as the number of links increases since more transmissions lead to higher throughput. From Fig. 6.12(b), we find that the throughput increases as α increases. For LDP, it is because when α increases, the partitioned square size decreases (by Formula (5.80)), which leads to more partitioned squares and hence more links to be scheduled. For RLE, it is because smaller α makes few nodes eliminated in each iteration (by Formula (5.42)). In DLS, each sender makes the decision based on its historical transmission results, i.e., a higher transmission success probability leads to higher likelihood it will send a packet at the current time slot. Fig. 6.11(b) indicates that higher α value leads to higher transmission success probability, and hence more links are likely to be scheduled in DLS. Fig. 6.13 shows DLS has a larger number of failed transmissions than LDP and RLE. This is because in DLS, each sender makes its strategy only based on the SIR received from its receiver without knowing the decision of its nearby senders.



(a) Number of nodes (b) Path loss exponent

Figure 6.13: Centralized vs. decentralized algorithms: the number of failed transmissions

6.2.2 CC Link Scheduling

In this section, we present simulation results of LLD-CLS, LLD-OCLS, and the greedy algorithm (CC-Greed). All nodes were distributed uniformly at random on a plane field of size 100×100. In the experiment, we measured the following two metrics: (1) *maximum delay*, which is defined as the number of time slots used to inform all receivers, and (2) *throughput*, which is defined as the number of receivers informed in a single time slot. We compared these two metrics of our algorithms with two smart non-cooperative link scheduling algorithms:

- ApproxDiversity [18]. It is a LLD based algorithm, which partitions the link set into disjoint link classes and schedules the links in each class separately.

- ApproxLogN [20]. It is a greedy algorithm, which always picks up the shortest link and excludes links conflicted with the picked-up links in each iteration. ApproxLogN is particularly efficient for the one-shot scheduling problem.

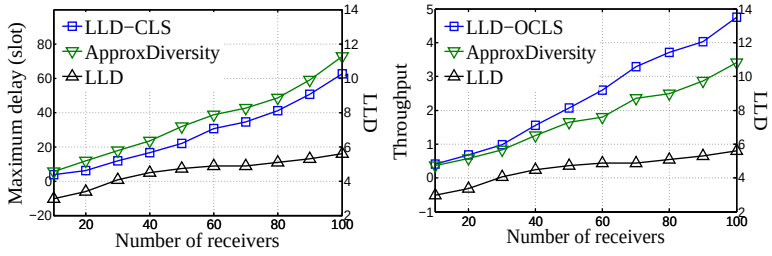
Like ours, both ApproxLogN and ApproxDiversity are polynomial time algorithms for the SIR model. The main difference is that ApproxLogN and ApproxDiversity do not allow CC in transmission. Since ApproxLogN is particularly efficient for the one-shot scheduling problem, we only compare ApproxLogN with our algorithms in terms of throughput.

First, we evaluate the performance of three LLD based algorithms: LLD-CLS, LLD-OCLS, and ApproxDiversity. In Fig. 6.14 (a) and Fig. 6.14 (b), we vary the number of receivers from 10 to 100 with 10 increase in each step, and compare the maximum delay and throughput, respectively. We set the number of senders to 200. As expected, LLD-CLS outperforms ApproxDiversity in maximum delay and LLD-OCLS outperforms ApproxDiversity in throughput. This is becasue LLD-

CLS (LLD-OCLS) allows receivers to combine weak signal powers from senders, which helps increase the opportunities for receivers to recover their messages. In addition, we have two observations from the figures: (1) the maximum delay increases as the LLD increases, and (2) the maximum delay increases as the number of receivers increases. These two observations are caused by the LLD-based algorithms' mechanism, which first partitions the link set into disjoint link classes, and then separately schedules the links in each class in squares. For (1), higher LLD always generates more link classes, leading to more time slots to schedule the whole link set. As for (2), higher receiver density causes more nodes to be in each square, and hence more time slots to schedule each link class.

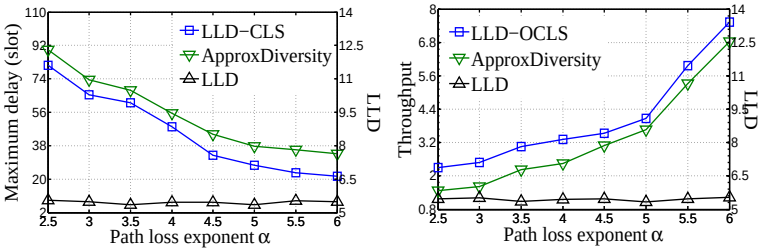
In Fig. 6.15 (a) and Fig. 6.15 (b), we compare different algorithms when the path loss exponent α was varied from 2.5 to 6 with 0.5 increase in each step. The number of senders and receivers are set to be 1000 and 100, respectively. Similar to Fig. 6.14, both figures demonstrate that LLD-CLS and LLD-OCLS outperform ApproxDiversity in terms of maximum delay and throughput, respectively, because of the benefit of CC. Another interesting observation is that with the increase of α, the maximum delay decreases and the throughput increases for both algorithms. This is because when α is smaller, the size of the squares partitioned by the LLD-based algorithms is larger (by Equ. (5.80)), which leads to more receivers located in each square and hence more time slots to schedule each link class.

We then compare the throughput of CC-Greed, LLD-OCLS, ApproxDiversity, and ApproxLogN. In Fig. 6.16 (a), we varied the number of receivers from 40 to 400 and set α to 3. In Fig. 6.16 (b), we varied α from 2.5 to 6 and set the number of receivers to 400. In both figures, each request has exactly two links. From both figures, we can find that CC-Greed is always better than ApproxLogN. Furthermore, we observe that when the number of receivers is low, CC-Greed has no significant better performance than LLD-OCLS and ApproxDiversity. However, as the density of receivers increases, CC-Greed presents increasingly better relative performance. This is because that CC-Greed can achieve constant approximation ratio in throughput (according to the analysis in Section 5.2.4), which enables it to achieve higher throughput than LLD-OCLS and ApproxDiversity when the receiver density of the network is high.



(a) Maximum delay (b) Throughput

Figure 6.14: Different number of receivers.



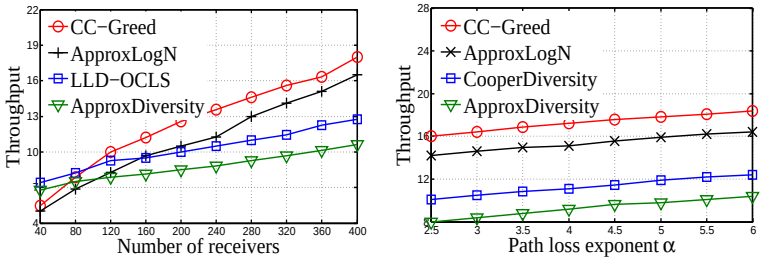
(a) Maximum delay (b) Throughput

Figure 6.15: Different pass loss exponent.

6.2.3 Vehicle Link Scheduling

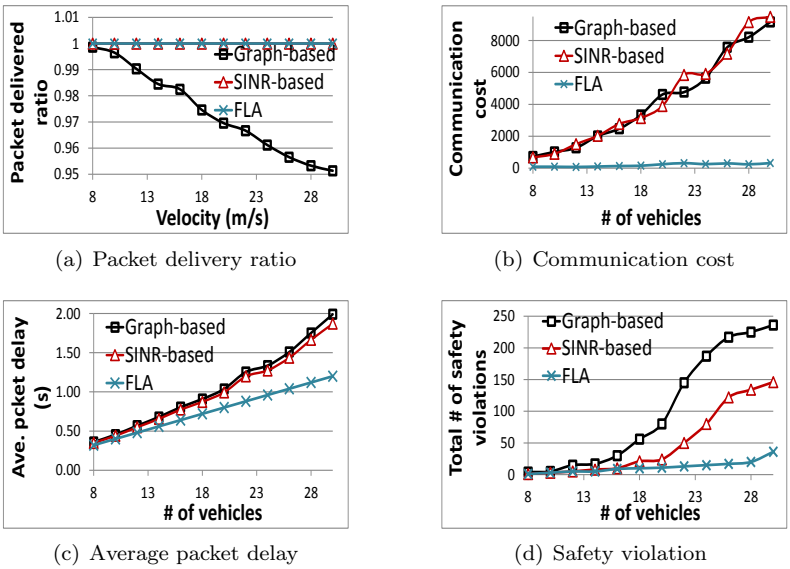
Finally, we evaluate the performance of the vehicle link scheduling method. We define *packet delivery ratio* as the percentage of the packets successfully delivered to their destination vehicles in al the packets sent out during the whole simulation time (i.e, 15 minutes). We define *communication cost* as the total number of packets sent out during the whole simulation time. In both the graph-based and SINR-based methods, the communication cost includes 1) the packets that all the vehicles send to the leader vehicle to inform their initial locations, 2) the packets that the leader vehicle sends to notify each follower vehicle its allocated time slots, and 3) the packets that each vehicle sends to the leader vehicle when its location changes. In FLA, the communication cost only refers to the number of packets that the leader vehicle sends to notify each vehicle the leader vehicle's updated location. For each packet, we define the packet delay as the time duration from the packet being sent to the packet being successfully delivered. And then, we calculate the average packet delay of all the packets sent during the simulation. In this test, in every minute, there is one vehicle entering the platoon and one vehicle leaving the platoon.

Fig. 6.17(a),(b), and (c) compare the packet delivery ratio, the communication cost, and the average packet delay of the three link scheduling methods with different number of vehicles.



(a) Different number of receivers (b) Different α values

Figure 6.16: Throughput of GREEDY, ApproxLogN, CoopDiversity, and ApproxDiversity .



(a) Packet delivery ratio (b) Communication cost

(c) Average packet delay (d) Safety violation

Figure 6.17: Comparison of different link scheduling methods.

Comparing the four figures, we find that FLA 1) produces the average packet delivery ratio almost the same as the SINR-based method but higher than the graph-based method, and 2) generates lower communication cost and average packet delay than both graph-based and SINR-based methods. FLA has much lower communication cost because when the relative location of a vehicle in the platoon changes so that its segment changes, the vehicle can change its own time slots based on its scored FLA table without communicating with the leader vehicle. However, both graph-based and SINR-based methods require all vehicles to send to the leader vehicle their locations. Also, when a vehicle changes its relative location, it needs to send a notification to the leader vehicle. Then, the leader vehicle recalculates the time slots and sends the new time slots to all the follower vehicles, which significantly increases the communication cost. More packets generate more interference and hence

lower packet delivery ratio. Furthermore, when more packets are transmitted, each packet needs to wait longer time before other packets finish, leading to higher packet delay. Vehicles are less likely to adjust their velocities in time with higher packet delay when their neighboring vehicles' relative locations are changed in the platoon, leading to more safety violations. Hence, both graph-based method and SINR based method have higher packet delay an more safety violations than our method. As a result, compared to the SINR-based method, FLA has lower communication cost, average packet delay and lager number of safety violations without compromising the packet delivery ratio, while compared to the graph-based method, our method has better performance in all aspects of communication cost, average packet delay, packet delivery ratio, and total number of safety violations.

Chapter 7

Conclusion

In this dissertation, we propose four methods to solve the challenges in realizing efficient data searching and routing in mobile opportunistic networks.

First, our objective is to find an optimal solution to choose immediate nodes in transmission routes for en/decoding packets in wireless networks in order to minimize the packet delay and increase the throughput. We mathematically analyze the packet delay and model the problem as an integer programming problem, which helps to discover a globally optimal solution. Taking into account the scalability of the network and limitation of the information that each node can collect, we propose a distributed scheme that can achieve performance comparable to the globally optimal solution. The simulation results in MATLAB demonstrates that our scheme performs better than previous packet recovery schemes. Since CEDAR does not have to be tied up with any specific error estimator, e.g., EEC, in our future work, we will try different error estimation codes for CEDAR. Also, we will further consider how to design the method to settle the problem that data rate may vary in each hop for transmissions. Finally, we will combine opportunistic routing with CEDAR to further reduce decoding delay hence packet delay. That is, when a key node overheard a packet, it directly forwards it rather than waiting for the packet sent by its previous receiver to decode it.

Second, by incorporating Rayleigh fading model into the link scheduling problem, we formulated a *Fading-Resistant Link Scheduling problem* (Fading-R-LS) with the objective to maximize the network throughput. The challenge for this problem is its complicated judgement for a successful transmission. As a solution, we derived the closed form of the probability distribution of the SIR received by each receiver, and found that checking transmission success is equivalent to

checking whether the sum interference factor from all the senders to this receiver is lower than a threshold. Based on this finding, we proved *Fading-R-LS* to be NP-hard and proposed two centralized algorithms (LDP and RLE) and one decentralized algorithm (DLS). Both theoretical analysis and experimental results demonstrate that LDP and RLE can substantially improve packet delivery ratio in fading environments compared to previous algorithms.

Third, to study the link scheduling problem in CC networks, we have formulated two problems, namely the CLS problem and the OCLS problem. The goal of CLS is to inform all receivers using as few time slots as possible, while the goal of OCLS is to maximize the number of informed receivers in one time slot. We have proved that both problems are NP-hard. As a solution, we have proposed a *link length diversity* (LLD) based algorithm for both CLS and OCLS problems, with $g(\mathcal{K})$ performance guarantee. Further, we have proposed an algorithm with $O(1)$ approximation guarantee for OCLS in the case that the number of senders in each request is upper bounded by a constant. The experimental results indicate that our cooperative link scheduling algorithms outperform non-cooperative algorithms.

Finally, we applied the link scheduling method to the vehicle platoon network. In particular, we designed the Fast and Lightweight Autonomous channel selection algorithm (FLA), in which each vehicle determines its time slots only based on its distance from the leader vehicle. Our simulation results show that the decentralized platoon network can scale out well with low packet drop rate, low packet delay, and low safety violation. Also, FLA outperforms the previous channel allocation methods for platoons in terms of packet delivery ratio, packet delay, communication cost, and safety violation.

The future work will be four-fold. For the relay selection problem:

1) We will consider how to settle the problem that data rate may vary in each hop for transmission;

2) We will combine opportunistic routing with CEDAR to further reduce packet latency. That is, when a key node overhears a packet, it directly forwards it rather than waiting for the packet sent by its previous receiver to decode it.

For the link scheduling problem:

3) We will take into account more fading models such as the Rician fading models;

4) In our current cooperative link scheduling method, we did not take into account the correlation and interference among different channels for cooperative communication. In our next step,

we will address this shortcoming and reformulate the cooperative link scheduling problem by considering the spatially-correlated interference across channels. We also need to analyze the difficulty of the new problem and propose time-efficient algorithms. Finally, we plan to implement our cooperative link scheduling method in a real testbed..

Bibliography

[1] Marjan Baghaie and Bhaskar Krishnamachari. Delay constrained minimum energy broadcast in cooperative wireless networks. In *Proc. of Infocom*, 2011.

[2] Michela Becchi. From poisson processes to self-similarity: a survey of network traffic models.

[3] E. Biglieri, J. Proakis, and S. Shamai. Fading channels: Information-theoretic and communications aspects. *Tran. on Information Theory*, 1998.

[4] G. Brar, D. M. Blough, and P. Santi. Computationally efficient scheduling with the physical interference model for throughput improvement in wireless mesh networks. In *Proc. of Mobicom*, 2006.

[5] Loc Bui, R. Srikant, and Alexander Stolyar. A novel architecture for reduction of delay and queueing structure complexity in the back-pressure algorithm. *Transaction on Networking*, 2011.

[6] D. Chafekar, V. Kumar, M. Marathe, S. Parthasarathy, and A. Srinivasan. Approximation algorithms for computing capacity of wireless networks with sinr constraints. In *Proc. of Infocom*, 2008.

[7] Deepti Chafekar, VS Anil Kumar, Madhav V Marathe, Srinivasan Parthasarathy, and Aravind Srinivasan. Capacity of wireless networks under SINR interference constraints. *Wireless Networks*, 17(7):1605–1624, 2011.

[8] Saeed Chahramant. *Fundamentals of Probability with Stochastic Process*. Western New England College, Reading, Massachusetts, 1986.

[9] Binbin Chen, Ziling Zhou, Yuda Zhao, and Haifeng Yu. Efficient error estimating coding: Feasibility and applications. In *Proc. of Sigcomm*, 2008.

[10] W. Cheng, X. Cheng, T. Znati, X. Lu, and Z. Lu. Joint TCP congestion control and CSMA scheduling without message passing. In *Proc. of Infocom*, 2009.

[11] R. Cohen and L. Grebla, G.and Katzir. Cross-layer hybrid FEC/ARQ reliable multicast with adaptive modulation and coding in broadband wireless networks. In *Proc. of INFOCOM*, 2009.

[12] Andrew R. Dalton and Jason O. Hallstrom. An interactive, source-centric, open testbed for developing and profiling wireless sensor systems. *International Journal of Distributed Sensor Networks*, March 2009.

[13] J. Dams, M. Hoefer, and T. Kesselheimy. Scheduling in wireless networks with rayleigh-fading interference. In *Proc. of ACM SPAA*, 2012.

[14] D.N.Rowitch and L.B.Milstein. On the performance of hybrid FEC/ARQ systems using rate-compatible low-density parity-check codes. *ITC*, 2000.

[15] T. A. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *Proc. of Infocom*, 2002.

[16] M. Ghaderi, D. Towsley, J. Kurose, Univ. of Calgary, and Calgary. Reliability gain of network coding in lossy wireless networks. In *Proc. of INFOCOM*, 2008.

[17] Andrea Goldsmith. *Wireless Communications*. Stanford University, Andrea Goldsmith, 2004.

[18] O. Goussevskaia, Y. Anne Osfwald, and R. Wattenhofer. Complexity in geometric SINR. In *Proc. of Mobihoc*, 2007.

[19] O. Goussevskaia and R. Wattenhofer. Complexity of scheduling with analog network coding. In *Proc. of FOWANC*, 2008.

[20] O. Goussevskaia, R. Wattenhofer, M. M. Halld orsson, and E. Welzl. Capacity of arbitrary wireless networks. In *Proc. of Infocom*, 2009.

[21] J. Gronkvist and A. Hansson. Comparison between graph-based and interference-based STDMA scheduling. In *Proc. of Mobihoc*, 2001.

[22] Zheng Guo, Jie Huang, and et al. A practical joint network- channel coding scheme for reliable communication in wireless networks. In *Proc. of MobiHoc*, 2009.

[23] Joseph L. Hammond and Peter J.P. O'Reilly. *Performance Analysis of Local Computer Networks*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1988.

[24] S. Haykin and M. Moher. *Modern Wireless Communications*. Prentice Hall, 2005.

[25] J. Huang, V. G. Subramanian, R. Agrawal, and R. Berry. Joint scheduling and resource allocation in uplink OFDM systems for broadband wireless access networks. *JSAC*, 2009.

[26] M.U. Ilyas, Moonseong Kim, and H Radha. Reducing packet losses in networks of commodity IEEE 802.15.4 sensor motes using cooperative communication and diversity combination. In *Proc. of INFOCOM*, 2009.

[27] Villy B. Iversen. *Teletraffic Engineering Handbook*. Technical University of Denmark, 2001.

[28] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *Proc. of SIGCOMM*, 2007.

[29] Predrag R. Jelenkovi and Jian Tan. Dynamic packet fragmentation for wireless channels with failures. In *Proc. of MobiHoc*, 2008.

[30] L. Jiang, D. Shah, J. Shin, and J. Walrand. Distributed random access algorithm: Scheduling and congestion control. *Tran. on Information Theory*, 2010.

[31] L. Jiang and J. Walrand. A distributed csma algorithm for throughput and utility maximization in wireless networks. *Transaction on Networking*, 2010.

[32] C. Joo, X. Lin, and J. Ryu. Distributed greedy approximation to maximum weighted independent set for scheduling with fading channels. In *Proc. of Mobihoc*, 2013.

[33] C. Joo, X. Lin, and N. Shroff. Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks. In *Proc. of Infocom*, 2008.

[34] Changhee Joo and Ness B. Shroff. Performance of random access scheduling schemes in multi-hop wireless networks. *Transaction on Networking*, 2009.

[35] K. Kar, S. Sarkar, A. Ghavami, and X. Luo. Delay guarantees for throughput-optimal wireless link scheduling. *Tran. on Automatic Control*, 2012.

[36] Shirish S. Karande and Hayder Radha. Non-linear integer programming by darwin and boltzmann mixed strategy. *IEEE Transactions On Multimedia*, 2008.

[37] Sachin Katti, Dina Katabi, Hari Balakrishnan, and Muriel Medard. Symbol-level network coding for wireless mesh networks. In *Proc. of Sigcomm*, 2008.

[38] B. Korte and J. Vygen. *Combinatorial Optimization Theory and Algorithms, Third Edition.* Springer, 2005.

[39] Ulas C. Kozat, Iordanis Koutsopoulos, and Leandros Tassiulas. Cross-layer design for power efficiency and qos provisioning in multi-hop wireless networks. *Wireless Communications*, 2006.

[40] A. Krifa and C. Barakat. An optimal joint scheduling and drop policy for delay tolerant networks. *TIT*, 2012.

[41] K. Leung and L. Wang. Integrated link adaptation and power control for wireless ip networks. In *Proc. of VTC*, 2000.

[42] H. S. Lichte, H. Frey, and H. Karl. Fading-resistant low-latency broadcasts in wireless multihop networks: The probabilistic cooperation diversity approach. In *Proc. of Mobihoc*, 2010.

[43] K. C. Lin, N. Kushman, and D. Katabi. Harnessing partial packets in 802.11 networks. In *Proc. of MOBICOM*, 2008.

[44] S. Lin and D. J. Costello. *Error Control Coding: Fundamentals and Applications.* Englewood Cliffs, NJ: Prentice-Hall, 2004.

[45] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *Tran. on Networking*, 2006.

[46] Sarah M Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In *Proc. of Formal Methods*, pages 42–56. Springer, 2011.

[47] M. Mandelbaum, M. Hlynka, and P. H. Brill. Nonhomogeneous geometric distributions with relations to birth and death processes. *Springer J. TOP Business Econ.*, 2007.

[48] I. Maric and R. D. Yates. Cooperative multicast for maximum network lifetime. *IEEE J. Sel. Areas Commun.*, 2005.

[49] Michael J. Neely. Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic? *Transaction on Networking*, 2009.

[50] Guanhong Pei and V.S. Anil Kumar. Low-complexity scheduling for wireless networks. In *Proc. of Mobihoc*, 2012.

[51] Larry L. Peterson and Bruce S. Davie. *Computer network: a system approach.* Morgan Kaufmann, 2007.

[52] A. Reddy, S. Sanghavi, and S. Shakkottai. On the effect of channel fading on greedy scheduling. In *Proc. of INFOCOM*, 2012.

[53] G. Sharma, N. B. Shroff, and R. R. Mazumdar. On the complexity of scheduling in wireless networks. In *Proc. of Mobicom*, 2006.

[54] Sushant Sharma, Yi Shi, Y. Thomas Hou, Hanif D. Sherali, and Sastry Kompella. Cooperative communications in multi-hop wireless networks: Joint flow routing and relay node assignment. In *Proc. of Infocom*, 2010.

[55] S. Soltani, K. Misra, and H. Radha. On link-layer reliability and stability for wireless communication. In *Proc. of MOBICOM*, 2008.

[56] Sohraab Soltani, Kiran Misra, and Hayder Radha. Delay constraint error control protocol for real-time video communication. *IEEE Transaction on Multimedia*, 2009.

[57] E. C. Strinati, S. Simoens, and J. Boutros. Performance evaluation of some hybrid ARQ schemes in IEEE 802.11a networks. In *Proc. of VTS*, 2003.

[58] Peng Tian, Jian Ma, and Dong-Mo Zhang. Non-linear integer programming by darwin and boltzmann mixed strategy. *European Journal of Operational Research*, 1996.

[59] Hong Shen Wang and Nader Moayeri. Finite-state markov channel - a useful model for radio communication channels. *IEEE Transactions on vehicular technolgoy*, 1995.

[60] Liaoruo Wang, Benyuan Liu, Dennis Goeckel, Don Towsley, and Cedric Westphal. Connectivity in cooperative wireless ad hoc networks. In *Proc. of MobiHoc*, 2008.

[61] W. Wang, Y. Wang, X. Li, W. Song, and O. Frieder. Efficient interferenceaware tdma link scheduling for static wireless networks. In *Proc. of Mobicom*, 2006.

[62] X. Wang, Z. Li, and J. Wu. Joint TCP congestion control and csma scheduling without message passing. *TWC*, 2013.

[63] S.B. Wicker and V.K. Bhargava. *Reed-Solomon Codes and Their Applications*. Wiley-IEEE Press, 1994.

[64] Grace Woo, Pouya Kheradpour, Dawei Shen, and Dina Katabi. Beyond the bits: Cooperative packet recovery using physical layer information. In *Proc. of MOBICOM*, 2007.

[65] X. Xu, X. Li, P. Wan, and S. Tang. Efficient scheduling for periodic aggregation queries in multihop sensor networks. *Tran. on Networking*, 2012.

[66] Kyongsu Yi and Young Do Kwon. Vehicle-to-vehicle distance and speed control using an electronic-vacuum booster. *JSAE review*, 22(4):403–412, 2001.

[67] H. Yomo, S. S. Chakraborty, and R. Prasad. PHY and MAC performance evaluation of IEEE 802.11a WLAN over fading channels. *IEEE Transaction on Multimedia*, 2009.