INTRA- AND INTER-CLUSTER LINK SCHEDULING IN AD HOC NETWORKS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MUSTAFA LEVENT EKSERT

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
COMPUTER ENGINEERING

DECEMBER 2020

Approval of the thesis:

INTRA- AND INTER-CLUSTER LINK SCHEDULING IN AD HOC NETWORKS

submitted by **MUSTAFA LEVENT EKSERT** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** _______________

Prof. Dr. Halit Oğuztüzün
Head of Department, **Computer Engineering** _______________

Prof. Dr. Ertan Onur
Supervisor, **Computer Engineering, METU** _______________

Examining Committee Members:

Prof. Dr. Volkan Atalay
Computer Engineering, METU _______________

Prof. Dr. Ertan Onur
Computer Engineering, METU _______________

Prof. Dr. Ahmet Coşar
Computer Engineering, Ankara Bilim University _______________

Prof. Dr. İbrahim Körpeoğlu
Computer Engineering, Bilkent University _______________

Assist. Prof. Dr. Hande Alemdar
Computer Engineering, METU _______________

Date:

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Mustafa Levent Eksert

Signature :

ABSTRACT

INTRA- AND INTER-CLUSTER LINK SCHEDULING IN AD HOC NETWORKS

Eksert, Mustafa Levent
Ph.D., Department of Computer Engineering
Supervisor: Prof. Dr. Ertan Onur

December 2020, 98 pages

While clustering improves the scalability of ad hoc networks in comparison to flat topologies, it introduces additional challenges for resource scheduling when contention-free medium access is employed. This thesis addresses intra- and inter-cluster link scheduling problem in multi-channel ad hoc networks. A clustered network model as well as a novel inter-cluster link scheduling mechanism over Control and User Plane Separation structure are proposed. A resource distribution method that provides an effective and efficient solution is presented for preventing inter-cluster interference. A link scheduling strategy responsive to instant communication demands and available resources as a non-linear optimization problem, which is then reduced to linear form, is formulated. Link scheduling optimization is solved as integer linear programming and the implementation results are discussed and analyzed. The simulation results of the optimizer show that the run time cost of the optimization function drastically increases by the parameter size growth. An iterative hıstory-based solution to decrease the running time is proposed. The adaptation of the iterative history-based approach to the original optimization process makes the solution feasible and ensures near-optimal satisfaction and efficiency.

Keywords: Centralized MAC, Distributed MAC, Contention-free TDMA MAC, Hybrid TDMA MAC, Disjoint time slot, Collisions

ÖZ

TASARSIZ AĞLARDA ÖBEK İÇİ VE ÖBEKLER ARASI BAĞLANTI ÇİZELGELEME

Eksert, Mustafa Levent
Doktora, Bilgisayar Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Ertan Onur

Aralık 2020, 98 sayfa

Öbekleme, merkezi ve düz hiyerarşilere kıyasla, tasarsız ağlarda haberleşme performansını ölçeklenebilirlik açısından geliştirirken bağlantı çizelgeleme konusunda bazı kısıtlamalar getirmektedir. Bu tezde, çok kanallı tasarsız ağlarda öbek içi ve öbekler arası bağlantı çizelgelemeye değinmektedir. Öbekli ağ modeli ve Kontrol ve Kullanıcı Katmanı Ayrımı yapısı üzerinden özgün bir öbekler arası bağlantı çizelgeleme mekanizması orataya konmuştur. Etkin ve verimli bir çözüm sağlayan bir kaynak dağıtım algoritması da sunulmaktadır. Anlık iletişim istekleri ve uygun kaynaklara cevap verebilen bir bağlantı çizelgeleme yöntemi doğrusal olmayan ve sonradan doğrusal forma indirgenen bir eniyileme problemi olarak formüle edilmiştir. Bağlantı çizelgeleme eniyilemesi doğrusal tamsayı programlama ile çözülmüş ve gerçekleştirme sonuçları tartışılmış ve çözümlenmiştir. Eniyilemenin gerçekleme sonuçları göstermiştir ki, eniyileme işlevinin çalışma zamanı parametre büyüklüğünün artışıyla birlikte şiddetli bir biçimde artmaktadır. Çalışma süresini düşürmek adına bir geçmiş tabanlı döngülü çözüm sunulmuştur. Geçmiş tabanlı döngülü yaklaşımın eniyileme sürecine uyarlanması çözümü olurlu hale getirmekte ve eniyiye yakın doyum ve ve-

rimi garantilemektedir.

Anahtar Kelimeler: Merkezi MAC, Dağıtık MAC, Çekişmesiz TDMA MAC, Hibrit TDMA MAC, Ayrık zaman dilimi, Çarpışmalar

To my beloved family, who believes me and grants me their priceless and limitless support.

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Prof. Ertan Onur for his advice, criticism, encouragement, and insight throughout the research. In the long and challenging journey of my graduate education, I owe most of my achievements to his guidance, patience, support, and effort. His contribution to my academic knowledge and professional life is extremely valuable and truly exceptional.

I would like to express my sincere gratitude to my department, METU Computer Engineering (CENG). From my first encounter as a new adult to the present, I am very proud to be part of CENG as a student and a former teaching assistant for many years. I would like to thank my professors for equipping me with a top-tier computer science education, my former colleagues for their friendship and support, and the students for their effort in their courses.

The author would also like to thank Assoc. Prof. Dr. Hamdullah Yücel and my researcher colleague Mr. Doğanalp Ergenç for their contribution to the study.

ASELSAN, which is the company that the author currently employed, granted the opportunity of the research project that forms the backbone of this thesis. I would also like to thank the contact people for the research project, ASELSAN Project Team led by Dr. Tolga Numanoğlu. The technical assistance of Dr. Tolga Numanoğlu and his team through the research project is gratefully acknowledged.

This work is partially funded by ASELSAN under grant number HBT-TE-2017-012.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

ABBREVIATIONS

QoS	Quality of Service
CP	Control Plane
UP	User Plane
CDMA	Code Division Multiple Access
FDMA	Frequency Division Multiple Access
TDMA	Time Division Multiple Access
CSMA	Carrier Sense Multiple Access
TDD	Time Division Duplexing
MAC	Medium Access Control
CUPS	Control and User Plane Separation
ICLS	Intra- and Inter-Cluster Multi-Channel Link Scheduling
SDMA	Spatial Division Multiple Access
CDF	Cumulative Distribution Function
CHRA	Cluster-based Hybrid Routing Algorithm
OLSR	Optimized Link State Routing Protocol
AODV	Ad-hoc On-demand Distance Vector Routing
DMAC	Distributed Mobility Adaptive Clustering
LEACH	Low Energy Adaptive Cluster Hierarchy
HEED	Hybrid Energy-Efficient Distributed
PCA	Density-aware Probabilistic Clustering Algorithm
HCC	Highest Connectivity Cluster
DEECF	Distributed Energy Efficient Cluster Formation

VANET	Vehicular Ad Hoc Network
AMACAD	Adaptable Mobility-Aware Clustering Algorithm based on Destination Positions
GPS	Global Positioning System
DMCNF	Distributed Multi-hop Clustering algorithm for VANETs based on Neighborhood Follow
LTE	Long Term Evolution
WCA	Weighted Clustering Algorithm
WBACA	Weight Based Adaptive Clustering Algorithm
CLPSO	Comprehensive Learning Particle Swarm Optimization
MANET	Mobile Ad Hoc Network
MOPSO	Multi-Objective Particle Swarm Optimization
AWCP	Adaptive Weighted Clustering Protocol
NSGA-II	Non-dominated Sorted Genetic Algorithm Version 2
DCA	Dependability-based Clustering Algorithm
WSN	Wireless Sensor Network
HNCNN	Hysteretic Noisy Chaotic Neural Network
CC-TDMA	Coloring- and Coding-based Multi-Channel TDMA
CP-TDMA	Coloring- and Probability-Based TDMA
FATS	Fairness Adaptive TDMA Scheduling Algorithm
ACK	Acknowledgment
LB-FFVSA	Load-Balanced-Fair Flow Vector Scheduling Algorithm
ER-MAC	Hybrid MAC Protocol for Emergency Response Wireless Sensor Networks
FlexiTP	Flexible-Schedule-Based TDMA Protocol
CSMA-CA	Carrier-Sense Multiple Access with Collision Avoidance
ATMA	Advertisement-based TDMA Protocol
GBCA	Game Based Channel Assignment Algorithm

PCS	Parent-Children Set
MC-LMAC	Multi-Channel Lightweight Medium Access Control
IWSN	Source Industrial Wireless Sensor Node
STDMA	Single-Channel Spatial TDMA
TC	Transmission Scheduling
ROS	Round-Optimal Schedule
DRAND	Distributed Randomized TDMA Scheduling
SD-MUCS	Semi-Distributed Multi-Channel Scheduling Algorithm
CBT	Cluster-based TDMA
VC	VANET Coordinator
SAM	Slot-Allocation Map
ALOHA	Additive Links On-Line Hawaii Area
RR-ALOHA	Reliable Reservation
FI	Frame Information
A-ADHOC	Adaptive-ADHOC
TC-MAC	TDMA Cluster-Based MAC
CCH	Control Channels
SCH	Service Channel
DATS	Decentralized Adaptive TDMA Scheduling Strategy
ASAS	Adaptive TDMA Slot Assignment Strategy

CHAPTER 1

INTRODUCTION

Wireless communication is a vital instrument of the modern world, constituting the basis of tremendous technologies ranging from health care to the entertainment industry. Wireless Ad Hoc Networks supply connection service in case cost efficiency and portability is desired, when particularly traditional wireless networks fail to conform to these requirements. A challenging aspect of wireless ad hoc networks is managing the communication resources of the network to sustain connection and guarantee a certain level of Quality of Service (QoS). The study differs from its counterparts that scheduling clustering ad hoc networks regarding several aspects such as resource optimization and receiver entities of the defined links. This thesis proposes a method for communication of the network elements, by regulating link scheduling relying on a clustered ad hoc network structure. This chapter introduces the fundamental concepts of the study prior to the details of the proposed solution, followed by a literature review, methodology, results, and conclusions chapters.

1.1 Scope

Ad hoc networks are basically a group of self-organized communicating devices without a central controller or preinstalled and static infrastructure assistance, unlike infrastructure-based networks aided by base stations or access points[1]. Ad hoc networks have numerous application areas such as millitary applications [1], public safety [2], and health care systems [3], etc.

[1] Wireless ad hoc network is a type of an ad hoc network composed of a collection of wireless devices, that communicate over a common wireless medium. In this thesis, both “ad hoc network” and “wireless ad hoc network” terms refer to wireless ad hoc network and are used interchangeably as the target network type of this thesis is wireless ad hoc network.

Wireless ad hoc networks can be deployed to any location in which static terminals may not be available. However, management tasks such as resource sharing become more challenging in the absence of centralized controllers. Therefore, a management technique is required to establish an effective end-to-end communication scheme and orchestrate how network entities access the shared wireless channel. Clustering is an effective solution where nodes are grouped based on some properties. It is commonly employed for achieving scalability, coordination, and maintainability in ad-hoc networks [4, 5].

Since random access protocols are prone to collisions that hamper channel efficiency in wireless ad hoc networks, contention-free channel access protocols, for instance, time-division multiple access (TDMA), are generally preferred when the QoS is desired. In TDMA, each pair of transmitting and receiving nodes in the network has to follow a (predetermined) link schedule to access the medium for reliable and concurrent data transfer. If the medium consists of more than one channel, then this resource allocation problem is called multi-channel link scheduling that is NP-Hard [6].

1.2 Problem Definition: Resource Scheduling in Clustered Ad Hoc Networks

In contrast to fully-centralized structures that possess centralized managers and a flat structure that has no hierarchical entity, a clustered structure contains independent local coordinators in charge of control tasks such as resource scheduling and routing [7]. Control and user (data) plane separation (CUPS) in wireless ad hoc networks through clustering introduces a hierarchical model that shows a distributed behavior on a global scale and mimics centralized behavior on a local scale. In CUPS-based networks, control information may flow over a different sub-network than the data plane packets. Furthermore, nodes may have different roles in control and data planes. While the control plane functions such as routing or link scheduling can be run by some designated nodes (we will call them cluster heads later on), data plane functions such as forwarding the packet on an allocated channel at a specific slot can be accomplished by a different set of nodes. In this thesis, we concentrate on the link scheduling task in CUPS-based ad hoc networks.

Some of the significant application domains that require a high level of QoS are emergency communication systems, public safety networks, or military systems where data transmission may be critical imposing some stringent QoS requirements. For instance, in military ad hoc networking applications, a bound on delay for voice communication as well as a minimum throughput for video streaming may be needed in multi-hop communication among military units. TDMA link scheduling can then be used for handling sound and video streaming tasks for guaranteeing a certain level of QoS. However, due to the complexity of multi-channel link scheduling, performance degradation is possible if TDMA is directly applied to a highly-populated network. In this work, we employ a divide-and-conquer approach to reduce the search space by clustering nodes into small groups, each of which is implicitly scheduled for intra-cluster communication. Inter-cluster communication, which has the problem size of two groups combined, is maintained by the proposed inter-cluster TDMA link scheduling mechanism for QoS support, as well. The idea of partitioning the network elements is motivated by the civilian and military human distributions in rural areas. For instance, the standard size of a mountaineers' party or a military squad is around 10 people, which can be grouped as a cluster in an ad hoc network structure [8, 9].

In Figure 1.1, a simple CUPS-based ad hoc network is shown. In this example, we assume nine nodes are grouped into two clusters. The left cluster has nodes 1, 2, 3, 4, and 5. The right cluster has nodes 4, 6, 7, 8, and 9. Node 4 acts as the gateway between two clusters. Nodes 2 and 7 are the cluster heads of left and right clusters, respectively. Let us consider that left and right clusters are assigned to a different set of channels by spatial multiplexing for which we propose an approach in Section 3.2. Assume node 1 wants to send a packet to node 9. In legacy clustered ad hoc networks, cluster member send their packets to the responsible cluster head. The cluster head then runs the control plane functions (i.e., does routing) and then forwards the packet in the data plane over gateways and cluster heads (e.g., over path 1-2-4-7-9). The approach in CUPS-based ad hoc networks is different. In CUPS, node 1 sends its communication request (not the packet) to its cluster head. While nodes have different roles (cluster head, gateway, or cluster member) in the control plane, all nodes are equal in the data plane. The cluster head (node 2 in this example) runs the routing function and determines the route to the destination, i.e., path 1-3-6-9. Concurrent to

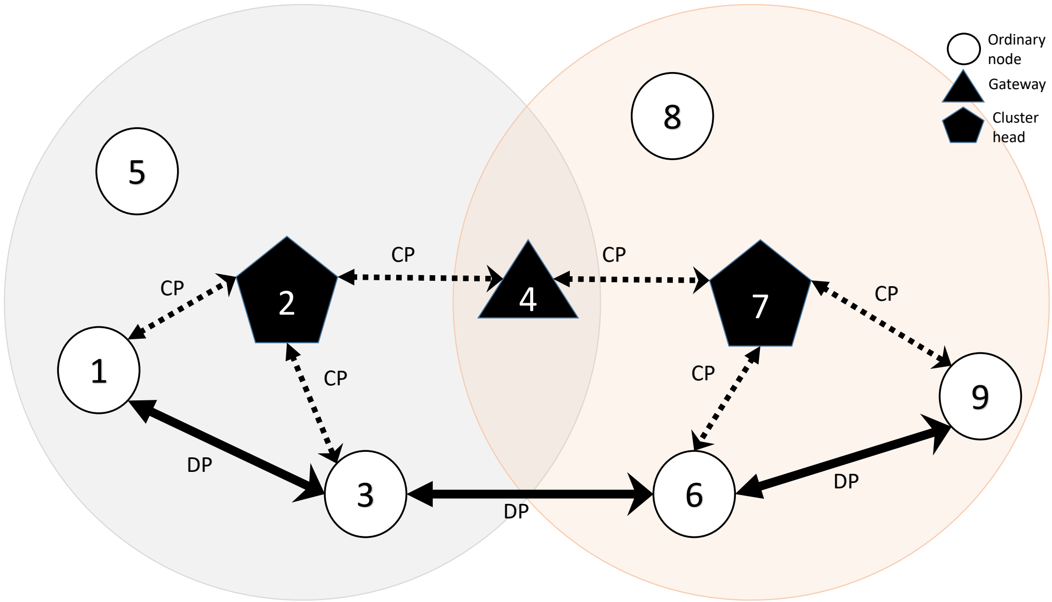


Figure 1.1: An illustration of a CUPS-based ad hoc network. Nine nodes are grouped into two clusters. Left cluster has nodes 1, 2, 3, 4, and 5. Right cluster has nodes 4,6,7,8, and 9. Node 4 (designated with a triangle) acts as the gateway between two clusters. Nodes 2 and 7 are the cluster heads (designated with pentagons) of left and right clusters. Ordinary nodes are designated with circles. Left and right clusters are assigned a different set of channels by spatial multiplexing. CP and DP represent control and data plane, respectively.

the routing task, the cluster heads together with the gateways run the link scheduling function to assign channel-slot pairs (resource blocks) to hops in the data plane. Then, node 1 transmits the packet to node 3 in the allocated time slot over the allocated channel. Node 3 transmits the packet to node 6 which is in another cluster in the allocated resource block. Finally, node 6 delivers the packet to node 9 in the right cluster.

As network division through clustering has the advantage of reducing the problem size, CUPS-based networks introduce inter-cluster coordination issues into the TDMA scheduling problem. For instance, peer contact between the cluster heads as well as mutual agreement on occupied resources for both communicating clusters is required. In this study, we present a solution to this novel coordination problem by the proposed inter-cluster link scheduling mechanism implemented through gateways.

1.3 Motivation

There are many challenges in CUPS-based ad hoc networks such as clustering and assignment of roles to nodes [10], routing [11], or neighbor discovery across clusters. In this thesis, we concentrate on TDMA link scheduling in CUPS based multi-channel ad hoc networks. We propose a scheme where intra-cluster link scheduling such as node 1 to node 3 link in Figure 1.1 and inter-cluster link scheduling such as node 3 to node 6 link in the same example is considered jointly where neighboring clusters employ different radio channels to minimize interference by spatial multiplexing. The problem we define in this study differs from the related work in the following major perspectives:

- The majority of the approaches in the literature concentrate on a single collision domain. In this study, we focus on CUPS-based clustered ad hoc networks that employ spatial multiplexing among clusters. Consequently, clusters generate distinct collision domains. To the best of our knowledge, we are the first to define inter-cluster link scheduling problem in such a scheme.
- A comprehensive multi-channel time-division link scheduling method for CUPS-based networks responsive to link requirements is yet to be proposed for end-to-end forwarding of packets considering intra- and inter-cluster link scheduling jointly.
- We consider not only unicast packets but also multicast and broadcast packets which makes the multichannel link scheduling problem complicated.
- Inter-cluster resource sharing problem is a specific task, which is rarely represented in the previous studies and is not investigated in terms of communication cost.

1.4 Contributions

The main idea behind the proposal in this study is to define resource allocation history as a constraint to the optimization problem that allows us to schedule the links in an iterative fashion. Assigning new links over the existing schedule, the scheduler is able to reduce its time cost in a high traffic network, to handle recent link requests with the presence of previously allocated resource set of the current schedule, and to arrange intra- and inter-cluster link scheduling. The study also proposes an inter-cluster link scheduling mechanism so that two neighboring nodes that belong to two distinct clusters can communicate over a link within a multi-channel TDMA scheme.

The main contributions of this thesis are as follows:

- We define a multi-channel link scheduling problem that extends to the intra- and inter-cluster communication in Section 3.1 that is briefly assigning link requests into available resource blocks conforming to requirements imposed by radio hardware features and time singularity of an active network element in CUPS-based ad hoc networks. The arrangement of the links is expected to prompt a node once at a time as with maximum utilization of the whole network elements. We assume nodes may require messages destined to multiple nodes. In other words, the problem definition covers multicast and broadcast communication requirements that are not extensively addressed in the literature.

- We introduce an efficient plan for connecting two heterogeneous structures in terms of communication: neighboring clusters, by identifying inter-cluster link scheduling steps within CUPS-based architecture in Section 3.1.3.2 where spatial multiplexing is employed.

- We present a two-phase scheduling optimization that is capable of performing control link allocation and repetitive link reservation: the history-based approach in Section 3.1.5. We consider not only unicast link demands but also multicast and broadcast link demands in order to schedule one-to-many node communication.

- Inter-cluster resource sharing algorithm steps are proposed and detailed in Section 3.2 with 3 different execution scenarios.

- Our scheduling method is extensively experimented by Monte-Carlo simulations and the results are presented in Section 4.1.

- We evaluate the basic link scheduler excluding history-based approach in Section 4.1.1 and propose an iterative history-based solution in Section 4.1.2 that is able to accelerate the link scheduling task by applying an optimization process into partitioned input data over a previously accumulated optimized scheduling. The results show that iteratively rescheduling the links over a predetermined history link set reduces the time cost of the scheduler as the number of assigned links almost remains unchanged.

- The inter-cluster resource sharing algorithm is tested and the effect of some cluster parameters are observed in Section 4.2.

- We elaborate on the contributions of this study by making a thorough inspection of the related work in Section 2.2. Moreover, at the end of Section 2.2, we compare the attributes of the proposed method and the related work in Table 2.3 suggesting our study has the most comprehensive and interoperable method by means of channel cardinality, link type, and traffic flow. Resource sharing algorithm alternatives in the literature are given in Section 4.2 with the methods in the graph coloring domain that can be applicable to the original problem for further inspection of all the possible solutions.

1.5 Outline

The outline for the rest of the thesis is listed in the following:

- In Chapter 2, background information about the thesis domain is given in Section 2.1 and the related work is introduced in Section 2.2.

- Chapter 3 proposes CUPS-based multi-channel link scheduling methodology with two sections: Intra- and Inter- Cluster Link Scheduling (ICLS) as Section 3.1 and Inter-Cluster Resource Sharing Algorithms as Section 3.2.

- Chapter 4 presents the simulation results of the algorithms proposed in Chapter 3 and related discussions.

- Chapter 5 concludes the thesis and gives the future work.

CHAPTER 2

LITERATURE REVIEW

This chapter explains further the subjects mentioned in the problem definition in detail in Section 2.1. Eventually, in this chapter, Section 2.1 expands the scope of this thesis with the literature review of the proposed methods that will be presented later in Chapter 3.

2.1 Background

CUPS-based ad hoc network is designed on a clustered ad hoc network architecture and multi-channel TDMA is used as the medium access method in the proposed study. Prior knowledge about the thesis domain contributes to further comprehension of the thesis. This section allows the reader to get familiar with these CUPS-based ad hoc network related concepts and enhance the insight for the presented methods in Chapter 3

2.1.1 Ad Hoc Networks

An ad hoc network consists of a set of devices that has the ability to communicate with each other without infrastructure and is spontaneously formed once the devices initially connect to each other.

Two basic approaches to wireless communication are cellular and ad hoc networks. Cellular networks are high-speed, high-capacity voice and data communication networks distributed over land areas called “cells”, served by fixed-location transceivers providing the cell transmission of voice, data, and other types of content. A cell typically uses a different set of communication resources from neighboring cells, to avoid interference and attempt to achieve guaranteed service quality within each cell.

The basic differences between cellular and ad hoc networks are summarized in Table 2.1. In contrast to cellular networks, ad hoc networks do not require predetermined infrastructure and is a communication platform among identical devices controlled in a distributed manner. Furthermore, messages are transmitted over two or more transceiver entities, called multi-hop communication in ad hoc networks as transmissions occur between users and base stations in cellular infrastructure which manages routing. Ad hoc networks need distributed routing algorithms, e.g., Cluster-based Hybrid Routing Algorithm (CHRA) [11], Optimized Link State Routing Protocol (OLSR) [12], and Ad-hoc On-demand Distance Vector Routing (AODV) [13], for packet forwarding over the transceiver nodes. However, end to and connection may be lost due to mobility. On the other hand, cellular communication is more resilient against handover cases in which a user travels from coverage of the current base station to that of neighboring base stations. Base stations provide guaranteed bandwidth to users in their coverage and utilize the same resource on distinct base station locations in the distance that do not cause interference between each other. In most cases, nodes in ad hoc networks share a common communication resource and fixed frequency assignment is not suitable for collective use unless a few nodes are occupied.

Cellular infrastructure service is not interrupted due to energy shortage unless the power line is damaged unlike ad hoc network stations with finite charge. Therefore, cellular networks are used in applications demanding permanent and continuous access such as mobile networks and wireless Internet infrastructures. However, a cellular infrastructure setup is highly expensive and dependent on the target location. Furthermore, cellular networks may fail in disaster and emergency scenarios or military communication when low-cost, portable, and flexible solutions, which are supported by ad hoc networks, are desired.

Table 2.1: Comparison of cellular and ad hoc networks.

Feature	Cellular Networks	Ad Hoc Networks
Infrastructure	Predetermined	Not available [14]
Typical communication	Single-hop	Multi-hop [14]
Routing	Centralized routing	Distributed routing [15]
QoS	Seamless communication in most cases, relatively easy to provide	Possible routing discontinuity in mobile networks, hard to provide [14]
Communication resources	User-specific	Common [16]
Frequency distribution	Spatial reuse	Dynamic [17]
Energy	Unlimited in base stations	Energy limitations due to limited battery life [17]
Management and maintenance	Costly	Cheap [18]
Architecture	Area-specific	Adaptive [14]
Applications	Mobile network, wireless Internet infrastructure	Disaster and emergency scenarios, military communication [19]

2.1.2 Clustered Ad Hoc Networks

In a clustered topology, nodes are grouped in the proximity of some controller nodes, namely **cluster heads**, in terms of some criteria such as neighborhood, energy, and mobility. Disjoint node groups are called **clusters** composed of a cluster head node and ordinary nodes, namely **cluster members**, which are managed by a cluster head that is dynamically selected. For instance, routing and resource allocation decisions may be taken by cluster heads for cluster members. Moreover, inter-cluster management for network-wide communication can be organized by the cluster heads as well.

In clustering algorithms, the main challenge is cluster head selection for both initial formation and cluster maintenance. Different algorithms focus on various aspects of nodes such as mobility, energy, and topology-based requirements of the scenarios for which they are designed. The evaluation of those criteria reveals which node is more suitable for being a cluster head. For instance, Figure 2.1 shows the importance

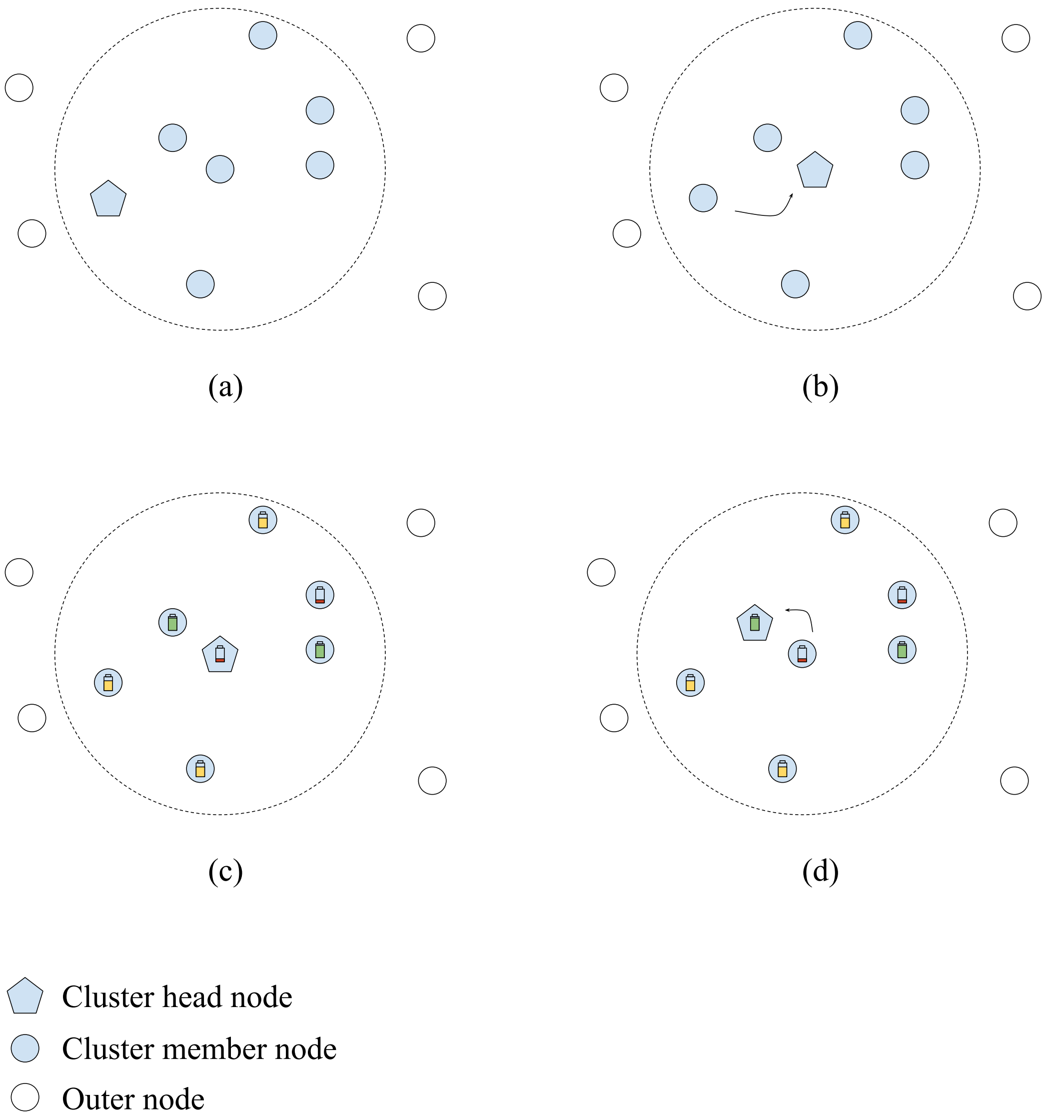


Figure 2.1: Cluster head change scenarios. Cluster head in (a), transfers its role to another member due to the centrality as illustrated in (b). In contrast, the cluster head in the midpoint as seen in (c), altered to the cluster head in (d) for the durability of the cluster.

of the position and residual energy of a node to be selected as a cluster head. In Figure 2.1a and Figure 2.1b, the node located in the center of a group of nodes tends to be a selected cluster head since accessing such node is easier for other member nodes. In other words, the more central node strongly covers its neighborhood. On the other hand, the battery life is considered to select cluster heads in Figure 2.1c and

Figure 2.1d. Instead of the node with low residual energy in Figure 2.1c, the node with higher residual energy (or battery life) becomes cluster head to decrease the risk of quickly-draining battery life of the cluster head in Figure 2.1d.

There is another important point that directly affects the overall clustered structure of the network, **cluster dependability**. Cluster dependability, in this work, indicates how stable, reliable, and popular (in terms of node and traffic density) a cluster is. Keeping their dependability high, the clusters potentially ensure the stability and reliability of a network. It also creates a cluster-comparison criterion for nodes that need to join one of the neighbor clusters as shown in Figure 2.2. In Figure 2.2a and Figure 2.2b, an outer node (i.e., not a member of any cluster) decides to join the cluster with higher-connectivity since high node-connectivity inside a cluster catalyzes the communication. In contrast, Figure 2.2c and Figure 2.2d illustrate joining the cluster with a fewer number of nodes to avoid higher interference, harder resource allocation, etc. due to excessive number of nodes in a cluster. In this sense, Figure 2.2 shows two different perspectives for cluster dependability that conceives a better communication opportunity and more versatile resource use for nodes.

2.1.3 Time Division Multiple Access

In wireless communication systems, which is the focus of this thesis, the wireless channel is generally shared among different users, introducing the broadcast property and the multiple access property [20]. The broadcast property means that a message that is transmitted by a user can be received simultaneously by multiple receivers. On the other hand, the multiple access property means that multiple active transmitters can cause interference at a common receiver. As the broadcast property increases the efficiency of message transportation, the multiple access property can cause interference and disrupt the receiving operation of the other users.

In a network, multiple entities require to access a shared communication resource to serve users or applications running on some device(s). Common resource utilization is possible if the resources are fragmented into operable components. Network users can communicate by either reserving or competing for these resource fragment.

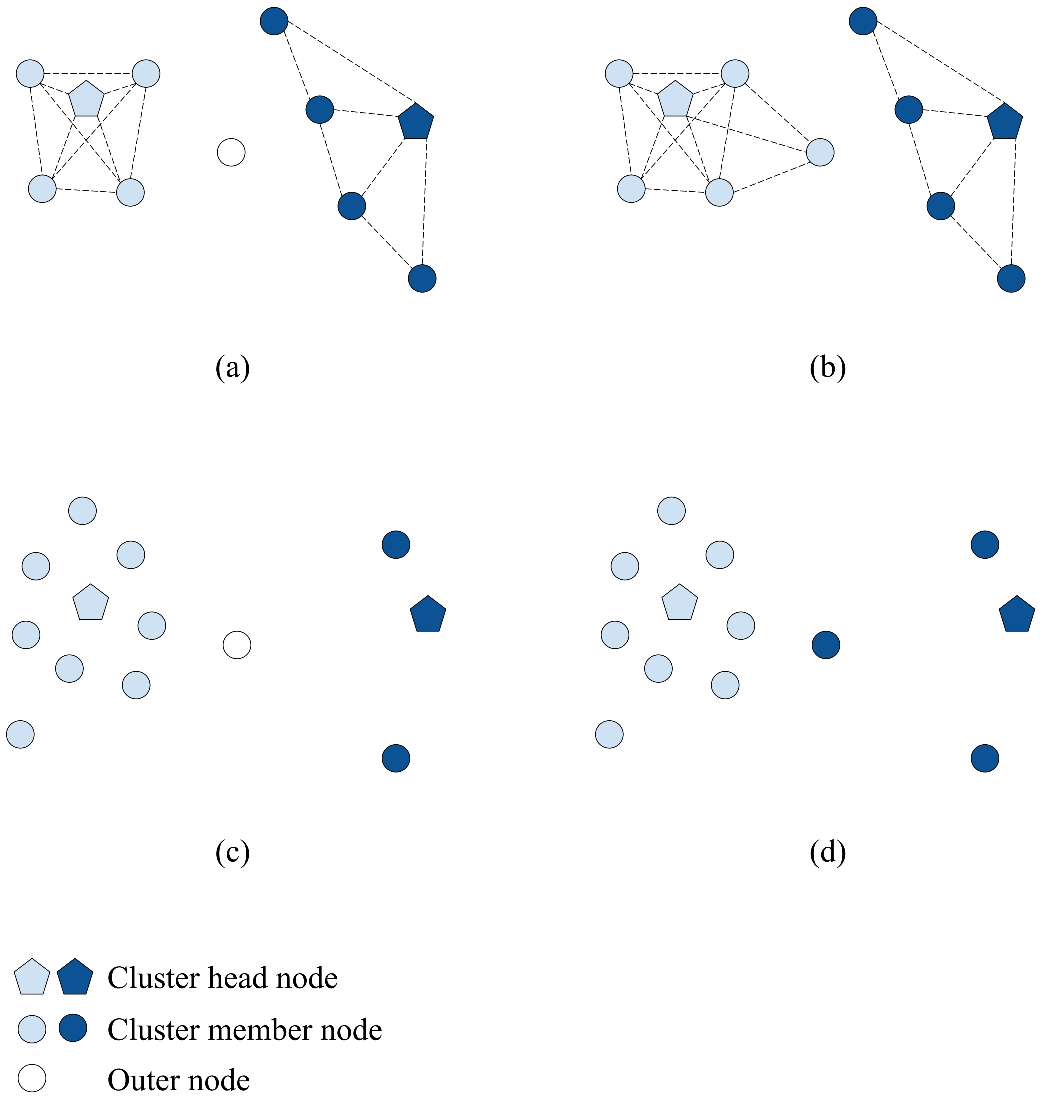


Figure 2.2: Affiliation scenarios of a foreign node to a dependable cluster. An outer node in (a) joins to the cluster on the left side, as illustrated in (b), due to higher interconnectivity. On the other hand, the outer node in (c) tends to join to the right-hand side in (d), as, in this case, node capacity have a greater influence on dependability.

Multiple access allows the sharing of a common broadcast medium by multiple stations. To achieve that, the transmission medium is partitioned into separate communication channels and then assigned to particular users upon their communication needs as a common practice [21]. In particular, some of the key multiple access methods are code division multiple access (CDMA), frequency division multiple access (FDMA), and time division multiple access (TDMA). These access techniques allocating the available bandwidth and transmission duration in a wireless network are classified as contention-free access methods and used in the generic transmission of data.

Contention-based multiple access lead users to access the shared medium without a reservation policy followed by the network. Example usage of a contention-based multiple access method, i.e., Carrier Sense Multiple Access (CSMA), is initial network setup. Since a stable allocation strategy would cause the users to hold resources even in an inactive state, a contention-free medium access method would waste resources, limit the capacity, and even lead to a shortage of resources. Therefore, connections cannot be established through permanently assigned channels.

In CDMA, a special coding scheme that assigns each user to a distinct orthogonal code for transmission is applied in order to enable stations use the channel concurrently. CDMA spreads across the wide bandwidth improving robustness against fading and noisy environments. However, the overall quality of services decreases, as the number of users increases and CDMA is prone to loss of orthogonality in case of attenuation.

FDMA allows multiple users to transmit messages through a communication channel by dividing the bandwidth into disjoint frequency intervals and reserving each interval to a different user. FDMA has a lower complexity compared to other multiple access techniques and brings relatively short header redundancy. On the other hand, If an FDMA channel becomes idle, the user does not transmit, yet occupies bandwidth and prevents other users to share capacity [22].

TDMA performs channelization of the communication resource in time domain. The entire channel bandwidth serves every user, but the channel access is available to a particular user in dedicated time duration. In other words, the channel is shared in time consecutively among the users in a circular manner. The transmissions are non-continuous and in a buffer-and-burst mode in its allowed time interval.

The advantages of TDMA can be listed as the following:

- TDMA typically applies Time Division Duplexing (TDD) which uses different time resources for transmission and reception. Therefore communicating device does not need to contain a duplexer component.

- Adaptive resource management is possible. TDMA allows the allocation of resources by concatenating or reassigning them to different users on demand.
- TDMA separates users in time, avoiding interference from other users' transmission as the other techniques that using the spectrum simultaneously are at risk in cases the orthogonality of multiple access deteriorates and inter-channel interference increases.
- The noncontinuous transmission leads to low power consumption, as the transmitter is idle and turned off most of the time.

The disadvantages of TDMA are listed below:

- Time synchronization is the essential requirement of a TDMA system. Control message traffic for time synchronization which brings overhead to the system is imminent.
- Receivers with different distances with respect to transmitter experience different propagation delays, leading to changes in synchronization of allocated time resources. In order to mitigate possible interference resulting from time shifts, vacant guard times between allocated time resources are required, introducing additional resource use.
- TDMA is subject to multipath propagation which is briefly the arrival of a radio signal to the receiving antenna by two or more paths with different propagation delays due to the interactions of the surroundings.

2.2 Related Work

This section provides a comprehensive inspection of the studies conducted on the research areas about this thesis. Section 2.2.1 introduces previous clustering algorithms and compares with our solution in [10]. In Section 2.2.2 link scheduling algorithms applicable to clustered ad hoc structures in the literature is presented with the comparison of the proposed link scheduling method in Section 3.1, as well. Finally, resource distribution methods are reviewed in Section 2.2.3 in both graph coloring and ad hoc networks domain in order to cover all possible solutions including the algorithms convertible to the original problem.

2.2.1 Clustering Algorithms

Several clustering methods for wireless networks have been proposed in the literature [23] [24] [25]. Different approaches have been investigated and presented as clustering strategies to improve some target performance measures. Some methods attempt to form stable clusters for reducing overhead while others focus on the energy consumption to prolong the network lifetime. Clustering methods can also be constructed based on some assumptions about the characteristics of applied network domain such as mobility or presence of sink nodes etc. The algorithms are categorized into 3 groups: single metric, weighted, and optimization-based clustering algorithms with respect to their clustering strategy. Single metric clustering algorithms rely on one main metric as weighted clustering algorithms act on scores, a total set of metrics. In contrast, optimization-based clustering algorithms aim to form clusters by optimized network parameters. Clustering algorithms are discussed under those three clustering types in the following part.

2.2.1.1 Single Metric Clustering Algorithms

Clustering algorithms concentrating on a single metric in electing cluster heads as well as grouping the nodes into clusters are classified as single metric clustering algorithms. One of the most fundamental clustering method, Lowest ID Cluster Algorithm, is proposed in [26] which forms clusters around the cluster head nodes that has the lowest ID in one hop neighborhood [27]. In [28], Distributed Mobility Adaptive Clustering (DMAC) algorithm determines cluster heads with respect to a single predetermined weight of each node in a distributed manner. Each node either announces itself as a cluster head or joins the cluster of the cluster head with the greatest weight. The algorithm also supports mobility by introducing procedures for new cluster nodes and link failures. Low Energy Adaptive Cluster Hierarchy (LEACH) [29] is a clustering method proposed for enhancing scalability and robustness in a data transmission flow from wireless microsensor nodes to the base station through the cluster heads. In each round, nodes acquire the chance of being cluster head role consecutively, leading to evenly distributing the load of cluster head role and additional energy consumption among the homogeneous sensors. However, these algorithms do not take into consideration any of the performance parameters of communicating nodes. Hybrid Energy-Efficient Distributed (HEED) clustering [30] is a clustering method designed for quasi-stational sensor networks to aggregate data to the centralized entity through cluster heads. In the algorithm, every node elects itself as a cluster head with a probability depending to the residual energy or joins the cluster head with the strongest received signal strength as a repetitive process. In Density-aware Probabilistic Clustering Algorithm (PCA), node degree is decisive on the probability of being cluster head to balance the number of clusters in dense areas [31]. Highest Connectivity Cluster Algorithm, HCC, [32] the nodes with the highest degree (number of neighbors) among 1-hop neighbors are elected as cluster heads forming the cluster with one-hop neighbors as the cluster members [27]. Similarly, in Distributed Energy Efficient Cluster Formation (DEECF) scheme [33], the node with the highest energy among its neighbors is chosen as the cluster head of the cluster including the neighboring nodes as cluster members [34]. Using an insufficient number of node parameter types is a big disadvantage of these type of algorithms, leading to instability of produced clusters.

Clustering methods are studied for various purposes and in different fields as well. Vehicular Ad Hoc Network (VANET) is a major domain for specialized clustering algorithms which impose additional constraints such as high level of mobility [35]. A mobility-based approach, MOBIC [36], which has a similar mechanism with Lowest ID Cluster Algorithm [26], uses relative mobility as a basic metric for cluster formation instead of unique identifiers. The node which has the least relative mobility among surrounding nodes announce itself as a cluster head where the other nodes in its vicinity become cluster members. Besides, a comprehensive clustering algorithm that provides cluster formation synchronization and maintenance in 4 phases is introduced in [37]. The main motivation of the algorithm is to maintain a quick and complete cluster setup. Unlike the distributed structure of our proposal, the resulting clusters are formed in a hierarchical structure with a diameter of at most four hops which introduce overhead for intracluster communications.

Some methods provide cluster head reelection, cluster maintenance, and cluster discovery mechanisms as well as cluster formation as a whole. For instance, Adaptable Mobility-Aware Clustering Algorithm based on Destination positions (AMACAD) [38] uses Global Positioning System (GPS) and destination data to form clusters and elect cluster heads whereas Distributed Multi-hop Clustering algorithm for VANETs based on Neighborhood Follow (DMCNF) [39] propagation delay for composition and maintenance of n-hop clusters. In [40], a clustering algorithm is proposed to generate and maintain the clusters as well as to suggest a mechanism of merging clusters for IEEE 802.11p and Long Term Evolution (LTE) hybrid network architecture. In [41], residual route time, which is simply a measure of overall neighborhood time duration, is used as clustering metric in VANETs. Although route information has an operational use in forming stable clusters, this approach heavily relies on predetermined directions of nodes and is not able to cluster arbitrarily moving nodes. Compared to the proposed method, these algorithms utilize relatively fewer number of metrics leading to a limited evaluation of communicating nodes. Similarly, cluster evaluation score is not provided with these methods unlike the proposed algorithm involving maintenance mechanism.

2.2.1.2 Weighted Clustering Algorithms

Some clustering techniques are based on a composition of metrics similar to the proposed method. For instance, Weighted Clustering Algorithm (WCA) [42] aims to create a dominating set of network graphs with the cluster heads. WCA elects cluster heads according to the weighted sum of ideal node degree, transmission power, mobility, and battery power metrics of nodes. The reelection of cluster heads is invoked as the relative distance between the cluster head and ordinary nodes changes. In contrast, [43] calculates the weighted average of consumed energy and degree difference among the neighboring nodes periodically and chooses the node with the least values among its one-hop neighbors as the cluster head. Similarly, connectivity, residual energy, and distance between each node and its neighbors are used as weighted metrics for electing cluster heads where the nodes are stationary and each cluster head is assumed to directly communicate to the sink nodes in [44]. Weight Based Adaptive Clustering Algorithm (WBACA) [45] determines cluster members of a cluster as the nodes around the cluster head, which has the least weighted sum of transmission power, transmission rate, mobility, battery power, and node degree metrics among cluster nodes. WBACA triggers affiliation or reclustering process when a node moves out of the cluster head's transmission range or two cluster heads move into each other's neighborhood. Weight-based clustering algorithms basically construct the clusters based on one or a combination of performance parameters of the communicating nodes. The weights of these parameters are determined by the user experiences and the performance results of the algorithms are measured with fixed and predetermined weights. In contrast, our proposal evaluates the effect of the weights as well as the performance measures to assess the algorithm. To achieve this, our approach offers an optimized set of weights that improve target objectives, i.e., stability, control overhead, and QoS. Predicted weights do not guarantee sound and consistent progress in performance measures and objectives.

2.2.1.3 Optimization-based Clustering Algorithms

Another category of clustering methods transform the clustering of a wireless network into an optimization problem and attempt to improve performance measures of a clustering algorithm over possible configurations of node roles or cluster parameters. Swarm or evolutionary optimization algorithms are used to maximize some quality measures of the network predetermined by the clustering methods. In Comprehensive Learning Particle Swarm Optimization (CLPSO) [46], given a Mobile Ad Hoc Network (MANET), a set of cluster formation is generated and optimized with a swarm optimization algorithm for a single objective, the total score of the cluster heads, which is a weighted sum of degree difference, a sum of the distances to member nodes, average speed, and time of being cluster head. In contrast, Multi-Objective Particle Swarm Optimization (MOPSO) is proposed to optimize the number of clusters in a MANET as well as energy consumption in nodes to provide energy efficiency and reduce the network traffic in [47]. Degree difference, energy consumption, mobility, and transmission range are determined as the objectives and the performance of

MOPSO is tested and compared to that of WCA and CLPSO with respect to three performance measures: the number of cluster-heads, energy consumed in a network, and load balance factor. Although MOPSO outperforms WCA and CLPSO by finding a relatively more optimal number of clusters, neither of these optimization approaches do offer an extensive cluster maintenance mechanism as introduced in the proposed method. Various swarm optimization techniques such as ant colony optimization [48] and grey wolf optimization [49] are applied for VANET clustering as well. Similar to MOPSO, both techniques try to optimize the number of cluster heads in the network and neither of them does not have a distributed cluster head reelection, cluster maintenance, and cluster discovery mechanisms. However, another study introduced in [50] differs from the aforementioned optimization methods, presenting a comprehensive clustering method, Adaptive Weighted Clustering Protocol (AWCP) in which cluster head election, cluster maintenance, and cluster merging operations are defined. An evolutionary optimization algorithm, Non-dominated Sorted Genetic Algorithm version 2 (NSGA-II) [51], aims to regulate a set of clustering parameters of AWCP to optimize three objectives: average cluster lifetime, packet delivery ratio, and control packet overhead. The clustering method is run on a network simulator and simultaneously reconfigured by the optimization algorithm operated on an optimization tool. Although this technique devises a set of clustering parameters and weights similar to the proposed method, the study does not investigate the correlation of any of the clustering parameters with the performance measures. Furthermore, stationary scenarios are not tested in network simulation, preventing to observe the influence of node mobility on the weight values and performance measures.

DCA [10], our clustering algorithm proposed as a separate study, relies on the dependability of the clusters in both cluster and cluster head selection. In DCA, clusters are formed and cluster heads are elected by optimizing a set of metrics, i.e., energy, link quality improvement ratio, self-to-cluster degree ratio, clique-to-degree ratio, centrality and capacity utilization for cluster head election and cliqueness, contraction, traffic density, and cluster degree for cluster selection.

2.2.1.4 Evaluation of DCA

Existing clustering methods have a wide range of variety in terms of network structures, target metrics, integrity of method, and application areas. Some methods concentrate on a single or a couple of objectives whereas some others evaluate a combination of node and network parameters as clustering criteria. The proposed method applies the latter approach to provide more stable clusters compared to the former approach as well as introduces a holistic clustering solution with bootstrapping, cluster, and cluster head selection mechanisms to perform a high level of integrity. To our knowledge, an investigation of the correlation between parameters and performance measures within a clustering method is hitherto unavailable. The proposed study simulates the clustering algorithm with a cross-layer implementation, conducts a sensitivity analysis for the metric weights for observing the influence of the parameter on the result, and offers a number of weight set for different performance needs.

Table 2.2 shows the differences between the clustering algorithms in terms of seven metrics demonstrating the features of the algorithms as well as emphasizing the novelty of DCA. *Cluster evaluation* refers to the ability to understand the dependability of clusters rather than individual nodes. *Adaptability* represents the ability to perform effectively to satisfy different objectives. *Optimization* is a comparison metric to show if the multi-metric algorithm is optimized and *metric analysis* represents if decisive metrics of clustering are analyzed individually. *cluster head selection metrics* and *cluster metrics* list the key factors for determining cluster head node and cluster members. Note that, DCA considers a cluster as a concept that needs to be evaluated taking its all members into account. On the other hand, others only consider certain aspects of the cluster head of the related cluster for cluster selection. Only the algorithms indicated with (*) in the cluster selection column have the collective approach to evaluate clusters (i.e., considering all member nodes) while the rest involves only cluster head. Finally, *mobility support* column denotes whether a clustering technique handles possible changes in the cluster integrity caused by the mobility of cluster nodes.

As shown in the table, cluster evaluation and metric analysis are originally introduced in DCA. DCA focuses on the dependability of clusters apart from the eligibil-

Table 2.2: The table emphasizes originality of DCA in terms of four metrics that we stated as our contribution. While ✓ means the algorithm fully satisfy the metric, ∼ represents that related algorithm partially supports it.

Algorithm Type	Algorithm	Cluster Evaluation	Adaptability	Optimization	Metric Analysis	Cluster Head Selection Metrics	Cluster Metrics	Mobility Support
	DCA	✓	✓	✓	✓	**Degree, energy, link quality, and packet traffic**	**Degree, energy, mobility, and packet traffic***	✓
Single Metric Clustering Algorithms	DMAC [28]					Predetermined weight	Predetermined weight	✓
	LEACH [29]		∼			Role duration	Signal strength	
	HEED [30]					Degree	Signal strength	∼
	PCA [31]					Degree	ID	∼
	MOBIC [36]					Mobility	Mobility	✓
	HCA [37]					Distance	Distance	✓
	AMACAD [38]	∼	∼			Mobility	Mobility*	✓
	DMCNF [39]					Mobility	Mobility*	✓
	VMaSC [40]	∼				Mobility	Mobility	✓
	[41]					Degree	Degree	✓
Weighted Clustering Algorithms	WCA [42]	∼	∼			Degree, energy, and mobility	Weight	✓
	WBCA [43]		∼			Degree and energy	Weight	∼
	[44]		∼			Degree, distance, and energy	Weight	✓
	WBACA [45]		∼			Degree, energy, mobility, and signal strenght	Weight	✓
Optimization Based Clustering Algorithms	CLPSO [46]		∼	✓		Degree, distance, and mobility	Weight	✓
	MOPSO [47]		∼	✓		Degree, energy, and mobility	Random	✓
	CACONET [48]		∼	✓		Degree and distance	Random	✓
	GWOCNET [49]		∼	✓		Degree and distance	Random	✓
	AWCP [50]	∼	∼	✓		Degree, distance, mobility, and role duration	Mobility	✓

ity of individual nodes for the maintenance of the clustered structure. It forms stable, energy-efficient, and high service-quality clusters with this approach as presented in our simulation results in [10].

Even though any score-based clustering algorithm is adaptable for different objectives in theory (that is why they are marked as "partially" adaptable), only in this study such objectives clearly defined in terms of various performance measures and DCA is optimized for each of them separately and compared with different algorithms that focus on those performance measures. Moreover, a non-optimized DCA is also designed to validate the effectiveness of our optimization framework.

2.2.2 Scheduling in Clustering Ad Hoc Networks

This section provides an extensive investigation of scheduler algorithms in the literature, considering not only similar settings with the proposed study but also diverse perceptions and conditions such as node scheduling, single-channel MAC, Wireless Sensor Networks (WSNs), etc. Inspected features of the studies are mostly uncorrelated. Therefore, in the following part of this section, algorithms are summarized in Table 2.3 displaying their aspects rather than categorizing with respect to a single property.

Several scheduling algorithms have been proposed for wireless communication [52, 53]. One approach is to schedule the nodes so that each node transmission may be received by all of its neighboring nodes without interference, namely node scheduling [52]. Some of the centralized node scheduling algorithms, in which the scheduling process is carried out by a single entity, optimize resource utilization with some auxiliary methods, e.g., neural networks [54], proposing novel Hysteretic Noisy Chaotic Neural Network (HNCNN), and genetic algorithms [55]. In contrast, [56] is a study presenting a distributed node scheduling method, in which scheduling is performed in each node in the network. TDMA scheduling in a central entity is not feasible since the scheduler node requires a high level of centrality. In fact, node scheduling algorithms, even distributed, are yet incapable of resolving the conflicts in receiver types depicted in Figure 3.5b.

Coloring- and Coding-based Multi-Channel TDMA (CC-TDMA) [57] is a multichannel and graph-based TDMA link scheduling algorithm based on edge coloring and algebraic coding theory. Although the algorithm prevents collisions to some extent, it fails to avoid collisions in some cases. Zhang et al. [58] propose a graph-based TDMA link scheduling algorithm, namely Coloring- and Probability-Based TDMA (CP-TDMA), using distributed edge coloring as well. The algorithm performs well in terms of throughput. However, it has limited slot utilization and conflict avoidance. Apart from the strategies taken into account of topology and neighborhood, some algorithms, namely cross-layer TDMA scheduling algorithms, attempt to optimize different network metrics such as route selection, load balancing, and power consumption. In Fairness Adaptive TDMA Scheduling algorithm (FATS) [59], the overall fairness of sensor networks can be obtained adaptively according to link quality. However, extensive use of Acknowledgment (ACK) messages introduces overhead to network communication. Load-Balanced-Fair Flow Vector Scheduling Algorithm (LB-FFVSA) [60] aims to determine an optimized and load-balanced slot assignment in terms of overall performance and fairness per flow. The algorithm increases frame length in high network traffic conditions, thereby increasing end-to-end delay. Hybrid MAC Protocol for Emergency Response Wireless Sensor Networks (ER-MAC) [61] and Flexible-schedule-based TDMA Protocol (FlexiTP) [62] are hybrid algorithms, which use Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA) in the setup phase and TDMA during network lifetime. ER-MAC is applicable in emergency scenarios and energy-efficient yet it is not scalable for high-density networks. Although FlexiTP is fault-tolerant and energy-efficient since nodes can modify their number of scheduled slots based on local information, the network has a tree structure consisting of improper parent-child pairs leading to fragile schedules. In [63] Advertisement-based TDMA protocol (ATMA), in which a frame is divided into contention-based synchronization and advertisement period followed by a contention-free data period, is introduced. On ATMA, links are reserved in the data period according to successful transmission to advertisement slots. Therefore, link demands are randomly served without an optimization process.

The algorithms focusing on MAC protocols also introduce various multichannel scheduling strategies in wireless sensor networks (WSNs) which employ mostly data gathering tree or forest network structure. Le et al. [64] propose to arrange nodes that communicate frequently into the same channel and separate nodes that do not communicate much into different channels since frequency synthesizer brings extra overhead to stabilize. The protocol is designed for a sink tree structure and is not tested for other network traffic cases. This method may also cause some packets sent to transmitting nodes to drop in case of frequent frequency switch. Game Based Channel Assignment algorithm (GBCA) [65] models a channel assignment game to solve the problem with a suboptimal result by negotiating channel usage among each Parent-Children Set (PCS) in a static and of tree/forest network topology. However, the solution is not effective as the method takes network topology into account rather than the packet traffic. In Multi-Channel Lightweight Medium Access Control (MC-LMAC) [66], time is slotted and several consecutive slots are organized as a frame within a multi-channel MAC protocol and a fully-distributed scheduling mechanism in which each node shares its bitmap of the slot and channel usage with its neighboring nodes in order to prevent interference. The algorithm incurs extra overhead due to control messaging transmission. In [67], a heterogenous frame structure, which involves slots of different lengths, is proposed in order to specialize in serving multiple urgent and small messages simultaneously. Several data transmissions from a Source Industrial Wireless Sensor Node (IWSN) to sink node, named as flows, are scheduled on divisible slots. The operation of scheduling simultaneous flows requires centralized management. Therefore, the method is not applicable to clustered structure and poses problems of centralized ad hoc networks.

Linear programming may generate several numbers of columns that are infeasible to handle and ineffective to reach the optimal solution for some cases. Column generation method which basically decomposes the problem into master and subproblems is an approach applied to integer programming in order to avoid redundant calculations. This method is used for optimizing link scheduling by a number of proposed methods in order to come up with a relatively more scalable scheme. For instance, [68] conducts resource optimization with both node and link scheduling, yet applied to single-channel spatial TDMA (or STDMA). Besides, as the least input load case, the optimizer assigns 26 links from 10 nodes into 17 resource blocks in 6 seconds, which is a long time compared to the Iterative ICLS optimizer performance in 4.4a. [69]

presents a cross-layer resource allocation with a nonlinear column generation technique. Therein, nodes are assumed to communicate with single-channel links and the time performance of the proposed technique is not discussed in the study. [70] compares two approaches of scheduling: Transmission Scheduling (TC)- approach and tree-based approach Round-Optimal Schedule (ROS) for multi-channel TDMA protocol on WSNs and focuses on optimizing sensor coverage of targets on a network with a limited (many-to-one) communication pattern.

A randomly distributed TDMA scheduling algorithm, Distributed Randomized TDMA Scheduling (DRAND), is presented in [71] operating a state diagram with 4 states: idle, request, grant, and release and the nodes transit among these states in order to be assigned to a slot. The algorithm randomly schedules a single channel TDMA protocol with high messaging complexity rather than adaptive to the communication needs of the nodes. In [72], two centralized TDMA scheduling algorithms consisting of a classical node-based and a novel level-based approach are introduced together with a distributed node scheduling with graph coloring. Similar to the aforementioned WSN schedules, the scheduling algorithm is designed for restricted network traffic, which forms a tree and all packets are destined to a sink node.

Some studies also conducted on scheduling in clustered wireless networks. In [73], centralized network-wide optimized TDMA schedules are aimed to provide high power efficiency, zero conflict, and reduced end-to-end delay. The objective of the scheduling algorithm is to achieve minimum TDMA frame length together via maintaining the slot reuse concept. However, the proposed method is limited to resolve single-channel TDMA scheduling and based upon WSN network traffic pattern. [74] proposes a scheduling method targeting an efficient time slot assignment while reducing power transmission in a clustered network structure. The algorithm adjusts the transmission power level with respect to the node distance. However, inter-cluster communication relies on backbone and slots are statically assigned to cluster members, causing battery drainage and redundant use of resources on certain nodes. Lee et al. [75] propose multi-channel TDMA scheduling algorithms for intra-cluster and inter-cluster communications. The proposed algorithms distribute available non-overlapping channels to proper clusters and reduce the optimal number of TDMA slots by scheduling them over the allocated channels. Although the algorithm manages to shrink frame size, the scheduling problem is addressed slightly since the instructions of handling constraints are not available in the study. In case of conflicts, extra channels (if available) for the clusters with dense packet traffic, which leads to long frame size, are reserved as the solution which is not effective and comprehensive for scheduling clustered wireless networks.

Further, scheduling techniques are presented for various types of networks. A recent work [76] is an extension to TDMA structure in Tactical Data Link [77] and reserves extra interrupt time slots on fixed time slot scheduling, aiming to decrease communication delay. The proposed method is not adaptive and very simple although the performance results show improvements. [78] proposes a vehicular ad hoc network scheduler providing better deadline miss ratios and faster response time. However, the method schedules safety and non-safety data traffic only between mobile vehicles and road side units, thereby limited to 1-to-n communication. In [79] a spatial TDMA is introduced for visible light communication networks with a weighted mixed integer linear programming model. Similar to the study in [78], the network structure is different than our proposed multi-channel TDMA ad hoc networks in which a transmission interferes with all other links in the same channel rather than a network model which can be represented as an undirected graph. Semi-Distributed Multi-Channel Scheduling Algorithm, called SD-MUCS, [80] is another link scheduler using network graphs. A multi-channel TDMA schedule is proposed in favor of resource reuse and connected nodes are grouped into clusters so that link scheduling is handled within clusters and inter-cluster links are scheduled in a centralized manner. The method attempts to minimize the number of required channels ignoring singular communication needs and separates controller elements from the communicating nodes as a feature of 5G architecture which is different than our target structure, ad hoc network.

The scheduling algorithms are summarized in Table 2.3 with their properties. Scheduler type implies whether the strategy of the algorithm is to schedule the nodes or the links. Scheduler deployment states the position of the scheduler entity in the network. The scheduling operation is carried out in a single entity if its scheduler type is centralized, whereas it is distributed in the case in which every node is involved in the scheduling process. If clustered, only cluster heads operate scheduling. Channel car-

Table 2.3: Overall comparison of scheduling algorithms.

Algorithm	Scheduler Type	Scheduler Deployment	Channel Cardinality	Supported Link Types	Optimization	Target Network	Traffic Flow
ICLS	Link	Clustered	Multi-channel	Unicast Multicast Broadcast	✓	Ad Hoc Network	Distributed
HNCNN [54]	Node	Centralized	Single-channel	Broadcast	✓	Ad Hoc Network	Distributed
[55]	Node	Centralized	Single-channel	Broadcast	✓	WiMAX	Distributed
[56]	Node	Distributed	Single-channel	Unicast Multicast Broadcast	✓	Ad Hoc Network	Distributed
CC-TDMA [57]	Link	Distributed	Multi-channel	Unicast		Ad Hoc Network	Distributed
CP-TDMA [58]	Link	Distributed	Single-channel	Unicast		Ad Hoc Network	Distributed
FATS [59]	Link	Centralized	Single-channel	Unicast		WSN	Tree
LB-FFVSA [60]	Link	Centralized	Single-channel	Broadcast		Wireless Multi-hop Network	Distributed
ER-MAC[61]	Link	Distributed	Single-channel	Unicast Broadcast		WSN	Tree
FlexiTP[62]	Link	Distributed	Single-channel	Unicast Broadcast		WSN	Tree
ATMA [63]	Link	Distributed	Single-channel	Unicast	✓	WSN	Distributed
[64]	Link	Distributed	Multi-channel	Unicast	✓	WSN	Tree
GBCA [65]	Link	Distributed	Multi-channel	Unicast	✓	WSN	Tree
MC-LMAC [66]	Node	Distributed	Multi-channel	Unicast Broadcast		WSN	Tree

Overall comparison of scheduling algorithms (continued).

Algorithm	Scheduler Type	Scheduler Deployment	Channel Cardinality	Supported Link Types	Optimization	Target Network	Traffic Flow
ICLS	Link	Clustered	Multi-channel	Unicast Multicast Broadcast	✓	Ad Hoc Network	Distributed
[67]	Link	Centralized	Multi-channel	Unicast		IWSN	Tree
[68]	Node Link	Centralized	Single-channel	Unicast Broadcast	✓	Ad Hoc Network	Distributed
[69]	Link	Centralized	Single-channel	Unicast	✓	Ad Hoc Network	Distributed
TC and ROS [70]	Link	Centralized	Single-channel	Unicast	✓	WSN	Tree
DRAND [71]	Node	Distributed	Single-channel	Broadcast		Ad Hoc Network	Distributed
[72]	Node	Distributed	Single-channel	Broadcast		WSN	Tree
[73]	Link	Centralized	Single-channel	Unicast	✓	WSN	Tree
[74]	Node	Clustered	Multi-channel	Unicast	✓	Ad Hoc Network	Clustered
[75]	Link	Clustered	Multi-channel	Unicast Broadcast	✓	Ad Hoc Network	Clustered
[76]	Link	Centralized	Single-channel	Unicast		Tactical Data Link	Distributed
[78]	Node	Distributed	Single-channel	Unicast Broadcast		VANET	Tree
[79]	Link	Centralized	Single-channel	Unicast Broadcast	✓	Visible Light Communication Network	Distributed
SD-MUCS [80]	Link	Clustered	Multi-channel	Unicast	✓	5G	Distributed

dinality is the count of available channels as supported link types implies the amount of the destination node(s) within an individual link recognizable by the scheduler. Table 2.3 presents if any metric is optimized on the scheduling process as well as its use of the target network. Last, traffic flow shows the destination characteristics of the links. Distributed traffic flow displays no specific pattern as links with tree flow destined to a significant entity, e.g., sink, base station, etc., with a single- or multi-hop. In contrast, clustered flow links are destined to the cluster head of the cluster head and inter-cluster communication takes place among cluster heads, building a backbone in the network topology. Table 2.3 acknowledges that among other methods, ICLS is the only multi-channel link scheduler that supports any link type, performs optimization, and operates on distributed links that do not follow a certain flow pattern. Furthermore, ICLS benefits clustered deployment in which the benefits of centralized and distributed structure coalesce for a more supervised and responsive schedule.

2.2.3 Resource Sharing Algorithms

Several methods from graph coloring as well as resource sharing in wireless networks are investigated in this section. The motivation for introducing graph coloring methods is the similarity between graph coloring and inter-cluster resource sharing problems and direct derivation from one to another. Therefore, some graph coloring algorithms are given as candidate solutions with the corresponding complexity information.

2.2.3.1 Distributed Graph Coloring Algorithms

In the message-passing LOCAL model of distributed computing, a communication network [81] is modeled by an n-vertex undirected unweighted graph $G = (V, E)$. The network is static, and so its topology does not change during the execution of an algorithm. The processors in the network are represented by the vertices of G. For every two vertices $u, v \in V$, there is an edge $(u, v) \in E$ if and only if the two processors corresponding to u and v in the network are connected by a communication link. Each vertex has a unique identity number. These numbers are assumed to belong to the range $\{1, 2, \ldots, n\}$.

A basic coloring solution can be defined as follows: Suppose we have a graph $G = (V, E)$ with maximum degree Δ, which is legally α-colored (or n-colored where n is the number of the vertices in G) by a coloring φ, for some $\alpha \geq \Delta + 1$. The following routine reduces the number of colors to $\Delta + 1$ within $\alpha - (\Delta + 1)$ rounds. In the beginning of each round each vertex sends its current color to all its neighbors. In the first round each vertex v of color α recolors itself in parallel into an available color from the palette $[\Delta + 1]$ (a set of colors $\varphi(v) = i$ such that $i \in \{1, 2, \ldots, \Delta + 1\}$). Repeating this procedure for colors $\alpha - 1, \alpha - 1, \ldots, \Delta + 2$ results in a legal $\Delta + 1$-coloring which is computed within $\alpha - (\Delta + 1)$ rounds.

Acyclic oriented graphs, which has oriented edges and has no cycle within any oriented paths, can be constructed in LOCAL model by transforming the undirected edges into directed ones by assigning direction form v towards u for $(v, u) \in E$ and $Id(v) > Id(u)$. Any acyclic path of G can be $d_i + 1$-colored along an acyclic path p_i with length k_i in $k_i + 1$ steps if the maximum out-degree of the path is d_i. Similarly, an acyclic graph can be $d + 1$-colored at most $k + 1$ steps where d is the maximum out-degree of G and k is the number of vertices within the longest acyclic path on G.

On the other hand, an oriented tree $T = (V, E)$ can be colored in just 2 colors by assigning the first color to the odd-numbered levels and the second color to the even-numbered ones by starting from the root vertex to the leaf vertices. However, this method needs l steps where l is the maximum depth of T which is $\Omega(n)$. Cole and Vishkin algorithm [82] leads to a 3-colored solution with a less number of round complexity complexity via reduction of the bit representation of distinct Ids of $v \in V$ where $Id(v) \in \{1, 2, \ldots, n\}$. This approach can be customized to general graphs by forest decomposition. The worst time complexities of the distributed graph coloring algorithms are illustrated in Table 2.4.

2.2.3.2 Resource Sharing in Wireless Networks

Distributing communication resources to network elements is a popular research topic and a numerous amount of study has been introduced [86, 87, 88]. In contrast to

Table 2.4: Number of colors and time complexities of graph coloring algorithms.

Algorithm	Coloring	Rounds
Cole and Vishkin [82]	3^{Δ}	$\log^* n + O(1)$
	$\Delta + 1$	$3^{\Delta} + \log^* n + O(1)$
Goldberg, Plotkin and Shannon [83]	$\Delta + 1$	$O(\Delta^2) + log * n$
Kuhn-Wattenhofer [84]	$\Delta + 1$	$O(\log\left(\frac{n}{(\Delta+1)}\right)$
Linial [85]	Δ^2	log*n + O(1)
Kuhn-Wattenhofer with Linial	$\Delta + 1$	$O(\Delta \log \Delta) + \log^* n$

clustered and distributed resource sharing algorithms, the problem of distributing resources to clusters is not common and is limited to a few studies. Therefore, in this section, clustered and distributed resource sharing schemes are presented, as well.

In [89], Ding and Zeng offer a clustered multi-channel network architecture. [89] implements the CDMA principle in cluster level, that is, each cluster head selects a different orthogonal code from that of its neighboring cluster heads. However, the inter-cluster resource distribution procedure is not mentioned in detail. Cluster-based TDMA (CBT) [90] proposes a single-channel TDMA clustered VANET in which each cluster is controlled by a VANET Coordinator (VC). If an intercluster interference is detected, VCs of two neighboring clusters send their Slot-Allocation Maps (SAMs) to each other in a transmit-and-listen manner. Allocation of the first VC that sends SAM remains unchanged while the second VC changes the intra-cluster allocation scheme in order to overcome the interference. The problem of inter-cluster interference is solved on demand. In other words, CBT provides correction rather than avoidance. In [91], RR-ALOHA dynamically establishes single-hop broadcast channels by Frame Information (FI) messages containing the status of (AVAILABLE or RESERVED) of each slot. Although the algorithm prevents hidden terminal and exposed terminal problem, the strategy for choosing frame length is not discussed and contention conflicts are not resolved. Adaptive-ADHOC (A-ADHOC) [92] proposes a frame adjusting method on [91], offering another option of adaptive frame length.

However, both techniques require the transmitting of FIs on a regular basis.

TDMA Cluster-Based MAC (TC-MAC) in [93] utilizes a multi-channel TDMA with frames divided into slots and control channels (CCHs) are composed of equal-sized mini slots in particular. Cluster members transmit on service channel (SCH) slots and CCH mini slots according to their unique local IDs given by the cluster head during the cluster formation phase. The main advantage of the resource assignment is the efficient utilization of channels within the cluster without intra-cluster without conflicts. On the other hand, each cluster head should periodically broadcast an updated list of the cluster members and their local IDs to all cluster members, which brings additional overhead. Moreover, TC-MAC design does not offer a medium access design as well as an interference prevention strategy for inter-cluster communication.

Vergados et. al. [94] proposes a distributed algorithm for bandwidth allocation, namely Local Voting, which targets load balancing, which is basically equalizing the ratio of the queue length and the current packet arrivals, over the number of reserved cells. 6TiSCH IEEE 802.15.4e TSCH network [95], a tree topology, is deployed and the reservation should be refreshed at the beginning of each frame which requires additional computation and transmission. VeMAC [96] provides a distributed resource sharing algorithm that reserves slots on nodes with a special role, namely “providers”, on multi-channel TDMA as in [97], Decentralized Adaptive TDMA Scheduling Strategy (DATS) assigns slots to a node attempting to access the network with respect to the relative position to its closest neighbor in a single-channel TDMA. Both VeMAC and DATS partitions TDMA frame as left, right, and unused according to node direction and slot vacancies but the network structure is not clustered. In contrast, Adaptive TDMA Slot Assignment Strategy (ASAS) [98] forms clusters with the nodes moving to the same direction and applies a similar 3-way frame separation with those of VeMAC and DATS except that contention-based slots are available and cluster members are assigned to specific positions in the frame according to the relative position of the cluster member to cluster head, i.e., being in the “back” or the “front” of the node. Although ASAS devises a simple strategy for inter-cluster interference avoidance by assigning different locations for “back” or “front” regions on the frame of two neighboring clusters, the behavior of this strategy is not clear in case a cluster has multiple neighbors.

In GAH-MAC [99], when more than one nodes occupy the same TDMA slot, the conflict is solved with a game based on Nash Equilibrium [100], leading nodes whether to reserve the original slot or another available one. However, a slot may not be allocated by any competing side in some cases. Jian and Hou [101] evaluate two candidate clustering algorithms, introduce a resource sharing formulation in which SINR of D2D equipment is optimized. Resource sharing among multiple clusters depends on some parameters and coefficients, e.g., power, leading to instability and performance degradation if not properly chosen.

In [102] and [103], SDMA, a design that maps a one-to-one map between the space division and communication resources, is offered. However, SDMA requires location information and GPS device and does not apply to some deployment options where lightweight equipment is used.

CHAPTER 3

CUPS-BASED MULTI-CHANNEL LINK SCHEDULING

Clustered has some advantages in terms of management, scalability, and bandwidth efficiency. CUPS is a promising approach for clustered ad hoc network yet resource management is an open problem to be resolved. Our proposal focuses on two fundamental aspects of communication over CUPS deployment: link scheduling and resource distribution. Link scheduling supporting multi-channel TDMA is handled systematically in Section 3.1 as resource distribution aspects are further discussed in inter-cluster level and a novel method is proposed for in Section 3.2.

3.1 Intra- and Inter-cluster Multi Channel Link Scheduling (ICLS)

We formulate the Intra- and Inter-Cluster Multi-Channel Link Scheduling (ICLS) problem in this thesis. We introduce the overall system model of the applied method with clustered network assumptions and CUPS-based system design details and assumptions imposed by clustered network link scheduling. Next, constraints imposed by clustered link scheduling and TDMA are presented. Eventually, we go into detail about ICLS problem definitions and run-time analysis.

3.1.1 Network Model

Ad hoc networks are modeled in his thesis as a decentralized wireless network composed of almost identical devices by means of communication skills and energy capacity. These homogenous entities, namely nodes, are able to directly communicate with each other without intermediate gateway or central access point and do not rely on a separate centralized controller. That means the network is entirely self-configured and stand-alone. Nodes can move across the territory and routes of network nodes do not have to fit in a mobility pattern or model.

In this work, we assume the network employs a clustered ad hoc structure. Formally,

$$
\begin{aligned}
Cl_i &= \{v_1, v_2, \ldots, v_{N_i}\}, \\
V &= \bigcup_{i=1}^{m} Cl_i, \\
Cl_i \cap Cl_j &= \emptyset \qquad i \neq j,
\end{aligned}
$$

where v_i denotes a node in Cl_i cluster that contains N_i number of nodes and V is the set of all nodes in the network. The whole network denoted by V consists of disjoint clusters, Cl_is. In other words, each node in the network belongs to exactly one cluster Cl_i, $i = 1, 2, \ldots, N_i$ in the data plane. Gateways are associated with multiple clusters in the control plane. On the other hand, some special cluster members in the vicinity of neighboring cluster heads are determined as **gateway** nodes in order to connect both cluster heads. Gateway nodes are involved in the control mechanism providing the inter-cluster communication from a cluster to a node that belongs to its neighboring cluster as well.

3.1.2 The CUPS Architecture

Ad hoc networks are essentially network organizations that accommodate spontaneously communicating nodes. Unlike some ad hoc network models [29, 30], the proposed model in this study does not possess some designated sink nodes to which the traffic of all or a part of nodes is directed. Hence, communication demands do not exhibit a pattern and the network handles the management of irregular demands autonomously.

Each cluster head is the local controller of its own cluster. Cluster heads and gateway nodes constitute a backbone that manages network traffic and resource sharing of the entire network. Cluster heads and gateways are the appointed roles in the **control plane** which is responsible for routing and scheduling tasks among others. To accomplish those tasks, control messages, which contain routing and scheduling related decisions, are transmitted over the control plane. In contrast, data messages can visit any node in the network. Therefore, **data plane**, which is the group of nodes transmitting or receiving data, refers to all the nodes in the network.

In Figure 1.1, the transmission of a data packet through CUPS architecture is illustrated. All the links from sender to receiver node are maintained by the corresponding cluster heads and gateway nodes in the control plane. As a result, a data packet can be transmitted through the shortest route without engaging busy controller nodes, i.e., cluster heads or gateways. Transferring data packets on the shortest path do not only decrease transmission time but also improves load balancing of the data traffic. Furthermore, the control plane can be extended beyond the coverage area of the ad hoc network by employing additional low-cost radio in the control plane. The separation of planes enhances the programmability and flexibility of the network significantly.

3.1.3 Communication in Clustered Structure

There are two main challenges for clustered networks from the resource scheduling perspective: intra- and inter-cluster scheduling. The former can be provided by simply scheduling all link demands with respect to the proposed constraints on the cluster head node as a whole, whereas the latter is more complicated since it requires mutual agreement between the scheduling of two communicating clusters that may employ different channel resources by employing spatial multiplexing.

3.1.3.1 Intra-cluster Scheduling

TDMA divides shared communication resource into the same length time durations named **slots**. Each slot is assigned to zero or one node to transmit a fixed-length message. In a multi-channel TDMA structure, which is employed by the proposed interference model, more than one transmission can be assigned to a slot within separate channels. Each link, which is defined by a sender node and one or more destination nodes, can be reserved to a **resource block**, a channel-slot pair. Slots are divided into the same number of resource blocks and consecutive slots are grouped into the

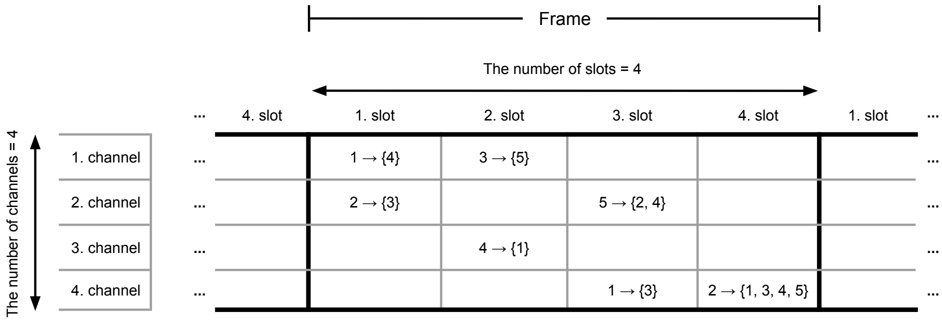
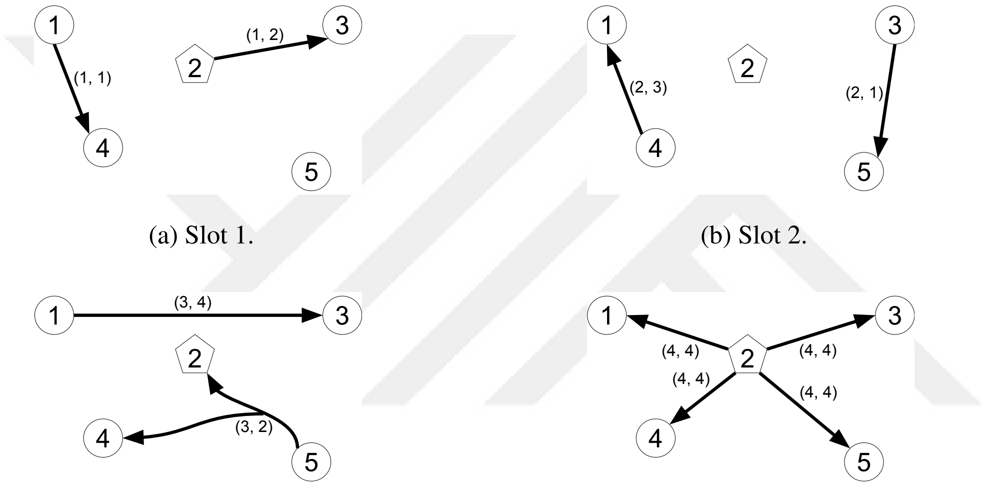


Figure 3.1: A proper schedule example of a cluster.



(a) Slot 1. (b) Slot 2.

(c) Slot 3. (d) Slot 4.

Figure 3.2: A regular transmission scenario with a schedule given in Figure 3.1.

same-length **frames** as shown in Figure 3.1.

The links can be defined as $v_s \rightarrow \mathcal{D}_s$, where $v_s \in \mathcal{V}$ is the source node, $\mathcal{V} = \{1, 2, \ldots, N\}$, $\mathcal{D}_s \subset \mathcal{N}$, $v_s \notin \mathcal{D}_s$, and N is the total number of nodes in the cluster. Each link is assigned to a resource block (i, j), where i is the slot and j is the channel number, respectively. An example of network traffic following the schedule depicted in Figure 3.1 is illustrated in Figure 3.2. In the example, there are 4 channels and 4 slots in a frame and $N = 5$ is the number of nodes in the cluster. Node 2 is the cluster head and the other nodes are the cluster members.

In Figure 3.2, node 1 transmits to node 4 at slot 1 on channel 1, denoted as $1 \rightarrow \{4\}$. Similarly, node 2 sends to node 3 at slot 1 on channel 2 as $2 \rightarrow \{3\}$. At slot 3, occupied by link $5 \rightarrow \{2, 4\}$, node 5 performs multicasting by transmitting to node 2 and node 4 on the same channel (channel number 2) as node 2 broadcasts to all other nodes at slot 4 on channel 4 by link $2 \rightarrow \{1, 3, 4, 5\}$.

A cluster is a part of the network managed in a centralized manner. Similarly, the scheduling mechanism is applied for the whole elements of the cluster and determined by the cluster head.

Data links are created on-demand by collecting link requests of member and gateway nodes into cluster head and by assigning a proper link in the schedule of that cluster. The scheduler organizes links for repetitive and consecutive frames that consist of a fixed number of slots. Prior to the beginning of each frame, each node sends its link demands to its local cluster head. The schedule of that cluster for the following frame is formed and broadcast to every cluster node in a beacon message before the frame starts. Therefore, scheduling of the next frame has to be decided in an instant time period which is at least smaller than a frame duration. That means, run time performance of the scheduling process has a significant role in the overall mechanism. The same procedure is applied in each frame during the lifetime of the network.

The links may be scheduled for one or more frames depending on the communication requirement of the transmitting and receiving nodes. For instance, bursty packets require instant and minimal messaging where sound or video calls are composed of stream packets that need to reside in one or more particular slots during consecutive frames. Besides, the control messages which are vital for the sustainability of the network should be granted higher priority in the schedule for rapid response to control the action. In this study, a set of repetitive link reservations containing data messages spreading over multiple frames and urgent control messages is defined as the **history** of the schedule. Before scheduling current link demands, the history link distribution is moved to the current schedule by reserving the links inherited from the previous frame on the exact resource blocks and is updated by adding recent history links. After that, the link demands collected for the current frame are scheduled on the reserved history links.

The steps followed by a cluster head for intra-cluster scheduling process is given as the following:

1. Collect all the link demands from cluster nodes over some static or dynamic uplinks.
2. Compose a history for the schedule to reserve prior blocks for fixed and primary links.
3. Run an optimization algorithm with the history input in order to provide as many links as possible for collected and internal link demands determined by route discovery.
4. Send the resulting schedule in the first slot of the corresponding frame as a beacon and repeat the steps for the next frame.

3.1.3.2 Inter-cluster Scheduling

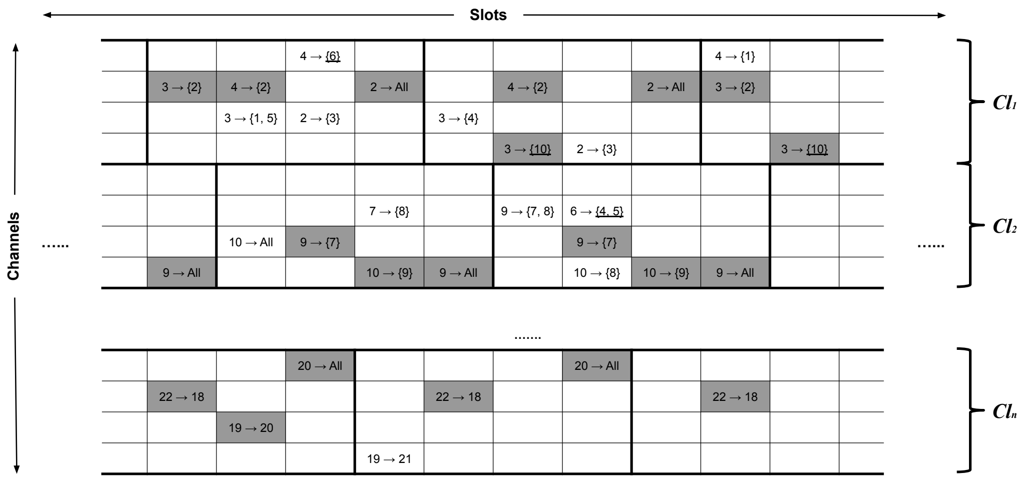
Since intra-cluster communication is scheduled in the cluster head nodes locally, simultaneous use of a common channel may occur within different clusters, resulting in interference in message transfer between two nodes in the same cluster. For reliable intra-cluster communication, it is essential that a cluster does not share the same channel set with the clusters in its communication range. Our technique employs a version of Spatial Division Multiple Access (SDMA) [102] adapted to ad hoc clustered network structure and relies on spatial multiplexing. More specifically, the channel spectrum of communication resources is divided into subsequent and disjoint channel ranges. Each cluster Cl_i utilizes a single range containing one or more channels as the intra-cluster resource. That means each cluster communicates through a multi-channel TDMA scheme within a predefined channel spectrum range for internal messaging of cluster nodes. Dividing the whole channel resources into disjoint segments and distributing these resource blocks among the clusters reduce problem space. Compared to a fully centralized scheduler containing all the variables, operating simultaneous schedulers on each cluster head that manages its cluster nodes and utilizes its dedicated channel range is more practical since each distributed scheduler sustains less number of channels and nodes. On the other hand, in a large network with a significant amount of clusters, assigning each channel to a distinct cluster is not feasible due to the limited channel spectrum. Channels can be shared in intra-cluster communication of non-neighboring clusters, providing **spatial reuse** without causing interference with outer intra-cluster messages. Channel assignment is managed by cluster heads with a distributed resource sharing method introduced in Section 3.2

An overall intra- and inter-cluster scheduling on multiple clusters $Cl_i, i = 1, 2, \ldots, m$, is illustrated in Figure 3.3a. The corresponding network topology of the schedules given in Figure 3.3a is illustrated in Figure 3.3b as well. **All** in Figure 3.3a refers to all nodes in the enclosing cluster. Shaded blocks are the reserved links as history and the rest are the optimized links and vacant resource blocks. In this example, a history may contain schedule broadcast and uplinks to the cluster head for the transmission of link demands. Repetitive links for data transmission may exist in the histories as well. For instance, voice and video calls require permanent links during the call in order to establish a continuous data transmission and high quality of service. It should be noted that the frames of the clusters are not aligned on the slot axis. Frame alignment among clusters is not mandatory due to the self-organized property of intra-cluster scheduling although we assume time synchronization.

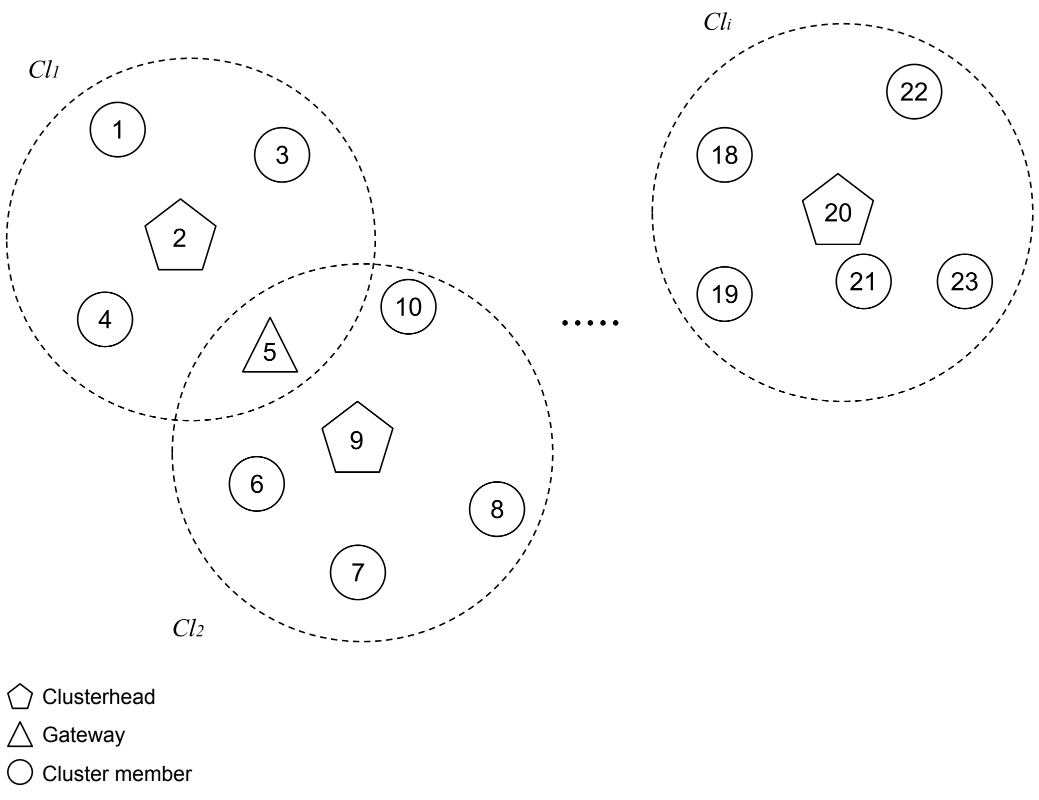
In Figure 3.3a, some reserved resource blocks have underlined destined nodes, which belong to neighboring clusters. For instance, $4 \to \underline{\{6\}}$ at the first frame of Cl_1 denotes to a link that transmits from node 4 to node 6, which is a cluster member of Cl_2. Links with underlined destined nodes are inter-cluster links in the CUPS architecture, which are elaborated in the following sections.

The distributed management concept of the proposed clustered network leads to heterogeneity among neighboring cluster schedules. Consequently, constructing an inter-cluster link between two neighboring clusters is considered a tedious task in terms of compatibility and time complexity. We propose an efficient and feasible mechanism that provides consensus for both sides of the communication. To achieve this, the assistance of gateway nodes is used in the scheduling process.

Intra-cluster communication is held by local scheduling on cluster head nodes in a dedicated collection of channels. However, any two nodes that belong to two separate clusters, and possibly use different channel ranges for intra-cluster communication,



(a) Schedules using disjoint channel ranges simultaneously.



(b) Network topology of the schedules given in Figure 3.3a.

Figure 3.3: Inter-cluster link scheduling example.

should also communicate with each other for the integrity of the network. An example case for an inter-cluster link between node 3 and 6 is shown in Figure 1.1. To accomplish that, inter-cluster links should be maintained between nodes that are the members of two neighboring clusters. Since two neighboring clusters transmit packets through different channel sets, a common channel should be provided. Control plane entities determine inter-cluster links with respect to routing requirements and inform communicating data plane elements. In this study, we assume that physical resources (channels) of the sender's cluster are employed for inter-cluster communication.

We follow a set of procedures that establish an inter-cluster link on control plane. First, the proposed inter-cluster scheduling is initiated by the cluster head on the sender side with a message requesting an inter-cluster link resource from the gateway node communicating to the destination cluster. Then, the gateway node determines the proper resource block for the inter-cluster link and sends its decision to both sender and destination cluster heads. Note that the gateway node is scheduled to listen to schedules of both clusters before the beginning of both cluster frames so that the inter-cluster resource block can be specified as appropriate blocks of each cluster schedule. Therefore, any conflicts between the previously reserved resource blocks of both schedules can be avoided. After the inter-cluster resource block is received by both cluster heads, it is applied to the next schedules of clusters and the inter-cluster message is transmitted and received at the corresponding reserved resource blocks of both sender and receiver cluster schedules. The steps of the inter-cluster scheduling process are displayed in Figure 3.4 as a close-section of the inter-cluster link between node 3 and node 6 shown in Figure 1.1. This process can be handled together with the network layer routing task in a cross-layer fashion to further enhance the performance.

3.1.4 Constraints

Organizing all the link demands in the frame introduces some rules, i.e., resolving some conflicts leading to interference in communicating nodes, as the scheduler should serve as many link requests as possible. The restrictions are basically to assign

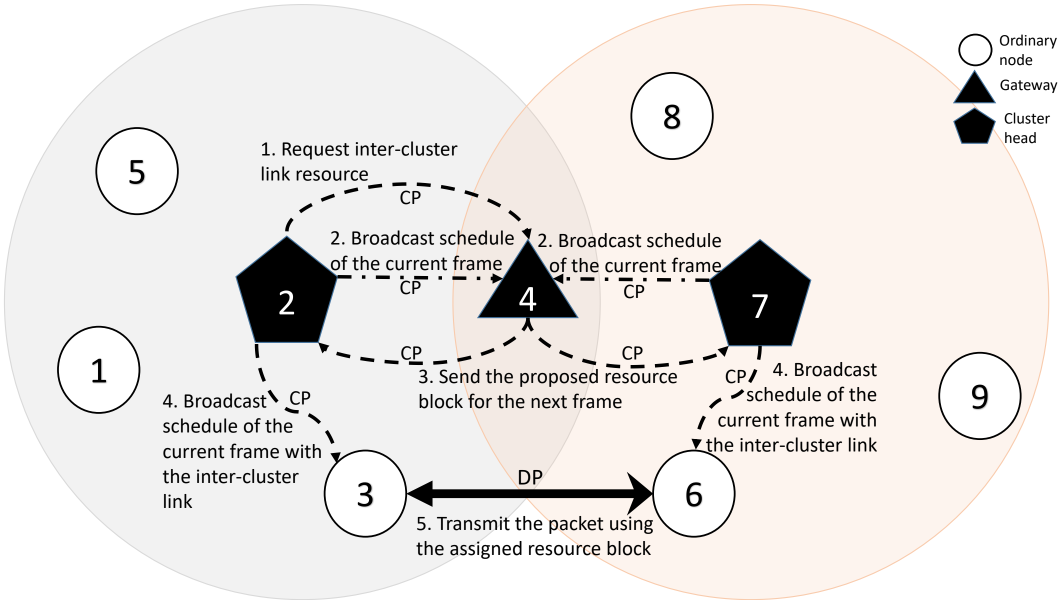


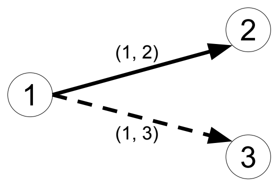
Figure 3.4: The inter-cluster link scheduling mechanism. Node 3 is the sender and node 6 is the receiver node in a neighboring cluster.

a link demand to at most one slot-channel block and vice versa as well as to prevent a node from either transmit or receive only one message at a time.

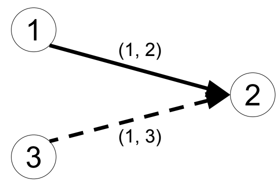
In Figure 3.5, a set of scheduling conflicts are illustrated. In all cases, node 1 attempts to send a message to node 2 in the resource block $(1, 2)$ and a conflicted transmission from node 1 to node 3 occurs at the same time in resource block $(1, 3)$. Four distinct conflicts are depicted separately in Figure 3.5 in separate subfigures. In Figure 3.5a, $1 \rightarrow 2$ transmission is scheduled in the same slot and causes a conflict in the sender node since a node cannot transmit two or more messages simultaneously. This conflict type is called **sender conflict**.

In Figure 3.5b, node 1 and node 3 plan to send a message to node 2 at the same slot. Since a node cannot listen to two distinct channels at a time, at least one of the messages cannot be received by node 2. This conflict type is known as **receiver conflict**.

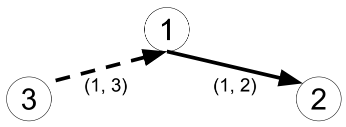
A node either sends or receives a message at a time due to the half-duplex characteristics of most radio hardware in devices. Therefore, if a sender node is scheduled to receive a message from a third node at the same slot, a receiver conflict occurs



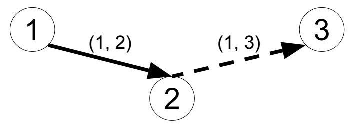
(a) Sender conflict.



(b) Receiver conflict.



(c) Receiver conflict in sender side.



(d) Sender conflict in receiver side.

Figure 3.5: Visualizations of sender-receiver conflict types.

on the sender side as shown in Figure 3.5c. Similarly, scheduling a node to send a message which is already scheduled to receive from another node (similar to node 2 in Figure 3.5d) ends up with conflict in that node.

The conflicts for inter-cluster communication and the constraints for inter-cluster link scheduling are not thoroughly addressed in the literature; we fulfill this gap in this study. An inter-cluster link should ensure the compatibility of two intra-cluster schedules of both sender and receiver clusters. That means, the transmission link should not interfere with an intra- or inter-cluster link and both sender and receiver nodes should not be assigned to another message transmission at that time. An inter-cluster link should conform to the following constraints:

- The reserved resource block should not be used by another intra-cluster link.
- Inter-cluster link should not cause sender and receiver conflicts in sender and receiver intra-cluster schedule, respectively.
- Receiver conflict in sender side and sender conflict in receiver side should also be avoided in the sender and receiver intra-cluster schedules, respectively.
- All inter-cluster links of a cluster should satisfy the requirements imposed for intra-cluster scheduling.

In Figure 3.3a, $4 \rightarrow \underline{\{6\}}$ is a valid inter-cluster link instance occupying one resource block in the first frame of Cl_1. Neither node 4 nor node 6 has an active role in intra-clustering schedules of their clusters at the slot that intra-cluster link resides. Besides, there are no extra inter-cluster links conflicting with $4 \rightarrow \underline{\{6\}}$ link.

It can be derived from the listed constraints that applying intra-cluster constraints to the combination of sender and receiver channel resources is sufficient for generating inter-cluster link scheduling. Therefore, our approach is to create the inter-cluster link in the gateway node with respect to constraints generated from both histories of intra-cluster schedules. After the inter-cluster link is generated, it is added to the histories of both intra-cluster schedules in the next frame. The inter-cluster link appears in the intra-cluster schedules of the next frames of sender and receiver clusters, thereby being served in the schedules of both the sender and receiver sides.

3.1.5 Intra- and Inter-Cluster Multi-Channel Link Scheduling (ICLS) Problem

The scheduling scheme is aimed to organize intra- and inter-cluster links with maximum utility and to resolve the constraints concurrently. Restrictions originated from any node affect all other cluster nodes due to densely distributed nodes within the cluster. Therefore, proportional to the amount of node population and packet traffic, the scheduler may be overloaded by a large list of rules, resulting in complex computations and long run time durations. The designed schedule is adaptive and responsive to the communication needs of nodes as well so that any size of instant or repetitive packet demand of cluster nodes can be satisfied by means of available resources.

The mathematical formulation of the problem extends to multicast and broadcast links, takes into consideration previous scheduling which contains permanently reserved blocks, and introduces extra restrictions for the current scheduling process.

Table 3.1 shows the parameters used in the problem definition. The matrices are given with their sizes in parentheses. $\mathbf{D}$ is the demand matrix, which denotes the message demand of each node given with the destination node(s). $\mathbf{H}$ matrix shows the preserved resource blocks from the previous frame, whereas $\mathbf{X}$ matrix simply shows the result of assignments, i.e., whether the transmission is assigned to a resource block

Table 3.1: Nomenclature.

Parameter/Matrix	Definition
T	Total number of time slots in a frame
C	Total number of channels
N	Total number of nodes
M	Maximum number of link request of a node in a frame
$\mathbf{D}$	Demand matrix $(N \times M \times N)$
$\mathbf{S}$	Intermediate resulting matrix $(N \times N \times M \times T \times C)$
$\mathbf{H}$	History matrix $(N \times N \times T \times C)$
$\mathbf{X}$	Resulting matrix $(N \times M \times T \times C)$

or not. $\mathbf{S}$ is an intermediate matrix used for resolving sender and receiver conflicts. It is basically a detailed version of $\mathbf{X}$ matrix in which destination nodes of message demand are expanded on an additional dimension. $\mathbf{D}$ and $\mathbf{H}$ are the input matrices when $\mathbf{X}$ is defined as the decision variable and $\mathbf{S}$ is categorized as an output matrix.

$$\mathbf{D}_{imj} = \begin{cases} 1, & \text{if node } i \text{ is allocated to a reasonable block} \\ & \text{for its } m^{th} \text{ demand to transmit node } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{S}_{ijmtc} = \begin{cases} 1, & \text{if node } i \text{ is allocated to } (t,c)^{th} \text{ reasonable block} \\ & \text{to transmit node } j \text{ for its } m^{th} \text{ demand,} \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{H}_{ijtc} = \begin{cases} 1, & \text{if node } i \text{ is allocated to } (t,c)^{th} \text{ reasonable block} \\ & \text{to transmit node } j, \\ 0, & \text{otherwise.} \end{cases}$$

$$\mathbf{X}_{imtc} = \begin{cases} 1, & \text{if node } i \text{ is allocated to } (t,c)^{th} \text{ reasonable block} \\ & \text{for its } m^{th} \text{ demand,} \\ 0, & \text{otherwise.} \end{cases}$$

Here, i and j are used as subscripts in matrices that range from 1 to N and represent the source and destination node indices, respectively. Similarly, $t = 1, 2 \ldots, T$, $c = 1, 2 \ldots, C$, and $m = 1, 2 \ldots, M$ refer to slot, channel, and link demand indices, respectively. Using the parameters and the matrices in Table 3.1, we define the following non-linear ICLS problem:

$$\text{maximize} \quad U_{NL} = \sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} \sum_{c=1}^{C} \mathbf{X}_{imtc} \tag{3.1}$$

subject to

$$\sum_{i=1}^{N} \sum_{m=1}^{M} \mathbf{X}_{imtc} \leq 1, \tag{3.2a}$$

$$\sum_{t=1}^{T} \sum_{c=1}^{C} \mathbf{X}_{imtc} \leq 1, \tag{3.2b}$$

$$\sum_{j=1}^{N} \mathbf{D}_{imj} - \sum_{t=1}^{T} \sum_{c=1}^{C} \mathbf{X}_{imtc} \geq 0, \tag{3.2c}$$

$$\sum_{t=1}^{T}\sum_{c=1}^{C}(\mathbf{X}_{imtc}\mathbf{D}_{imj} - \mathbf{S}_{imjtc}) = 0, \qquad (3.2d)$$

$$\left(\sum_{i=1}^{N}\sum_{m=1}^{M}\mathbf{X}_{imtc}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbf{H}_{ijtc}\right) = 0, \qquad (3.2e)$$

$$\left(\sum_{c=1}^{C}\mathbf{X}_{im_1tc}\right)\left(\sum_{j=1}^{N}\sum_{c=1}^{C}\left(\mathbf{S}_{im_1jtc} - \left(\mathbf{H}_{ijtc}\sum_{m_2=1}^{M}\mathbf{S}_{im_2jtc}\right)\right)\right) = 0, \qquad (3.2f)$$

$$\sum_{c=1}^{C}\left(\mathbf{X}_{i_1m_1tc} + \sum_{i_2=1}^{N}\left(\mathbf{H}_{i_2i_1tc} + \sum_{m_2=1}^{M}\mathbf{S}_{i_2ni_1pr}\right)\right) \leq 1, \qquad (3.2g)$$

$$\sum_{c=1}^{C}\left(\mathbf{S}_{im_1j_1tc} + \sum_{j_2=1}^{N}\left(\mathbf{H}_{j_1j_2tc} + \sum_{m_2=1}^{M}\mathbf{S}_{j_1m_2j_2tc}\right)\right) \leq 1, \qquad (3.2h)$$

$$\sum_{i=1}^{N}\sum_{c=1}^{C}\left(\mathbf{H}_{ijtc} + \sum_{m=1}^{M}\mathbf{S}_{imjtc}\right) \leq 1. \qquad (3.2i)$$

The objective of the formula is to maximize the utilization of available resource blocks U_{NL} as defined in (3.1). The constraint (3.2a) specifies that a resource block should be occupied by at most one message demand. Similarly, (3.2b) states that a message demand should be served in at least one resource block. (3.2c) ensures that m^{th} demand of node i should exist in the $\mathbf{D}$ matrix. (3.2d) states that if a message demand in $\mathbf{D}$ is served (the corresponding entry in $\mathbf{X}$ is set to 1) then it is also presented in $\mathbf{S}$ by setting the related indices to 1. (3.2e) denotes that a message demand should not be placed to a resource block which a history message belongs to. (3.2f) provides that a message demand does not have a common sender node with another current demand and history at the same slot if the message is served in a resource block. In other words, it tries to resolve the sender conflict depicted in Figure 3.5a. Similarly, (3.2i), which is also a definition of receiver conflict in Figure 3.5b, constrains that none of the target nodes of a served message demand receives a message from another node at the same slot. (3.2g) ensures that a history message or a current demand does not transmit to the node which sends a message to another node as a served message demand at the same slot (resolving receiver conflict in sender side shown in Figure 3.5c). Finally, (3.2h) ensures that a served message demand does not transmit to a node which is also transmitting as a history message or a current demand at the same slot (resolving sender conflict in receiver side shown in Figure 3.5d).

The resolution of the non-linear optimization problem in (3.1)-(3.2) contains a big number of constraints with matrix dimensions up to 5. Besides, the nonlinear structure of some of the constraints, e.g., (3.2e) and (3.2f), avoid to offer a feasible solution. Therefore, the optimization problem should be reformulated by eliminating nonlinearities. Redesigning the problem structure, the conflicts are combined into more compact constraints and nonlinearities are eliminated. The linear and simplified formulation, linear ICLS, is given as follows:

$$\text{maximize} \quad U_L = \sum_{i=1}^{N}\sum_{m=1}^{M}\sum_{t=1}^{T}\sum_{c=1}^{C}(N-i)(M-m)\mathbf{X}_{imtc} \tag{3.3}$$

subject to

$$\sum_{t=1}^{T}\sum_{c_1=1}^{C}\Bigg(\mathbf{X}_{i_1mtc_1} + 2\sum_{j_1=1}^{N}\Bigg(\mathbf{X}_{i_1mtc_1}\Bigg(1-\mathbf{D}_{i_1mj_1} + \mathbf{D}_{i_1mj_1}\sum_{c_2=1}^{C}\sum_{i_2=1}^{N}\sum_{j_2=1}^{N}(\mathbf{H}_{i_1j_1tc_2}+\mathbf{H}_{i_2j_1tc_2}+\mathbf{H}_{j_1j_2tc_2}+\mathbf{H}_{i_2i_1tc_2})\Bigg)\Bigg)\Bigg) \le 1, \tag{3.4a}$$

$$\sum_{m_1=1}^{M}\sum_{c_1=1}^{C}\mathbf{X}_{i_1m_1tc_1} + \sum_{i_2=1}^{N}\sum_{m_2=1}^{M}\sum_{c_2=1}^{C}\mathbf{X}_{i_2m_2tc_2}\mathbf{D}_{i_2m_2i_1} \le 1, \tag{3.4b}$$

$$\sum_{i_1=1}^{N}\sum_{m=1}^{M}N\mathbf{X}_{i_1mtc} + \sum_{i_2=1}^{N}\sum_{j=1}^{N}\mathbf{H}_{i_2jtc} \le N. \tag{3.4c}$$

The objective U_L in (3.3)-(3.4) is to maximize the total number of resource blocks used similar to the previous formulation except with a multiplier extension for preventing race conditions among the identical resource blocks. To overcome that, the resource blocks with less time and channel indices are favored in order to lead the scheduler to reserve these highlighted indices in prior. (3.4a) states that each demand can be assigned to at most one (t, c) pair and ensures that zero entries in $\mathbf{D}$ matrix are not reserved to a resource block. It also resolves the sender and receiver conflicts related to history assignments. (3.4b) is related to the schedule assignment of the recent resource demands and simply imposes that demand either sends or receives in a slot. In (3.4c), served demands are constrained to reside in exactly one resource block provided that it is not utilized by a history demand.

3.1.6 The Complexity of ICLS

In the given objective function (3.3), $(N \times M) + (N \times T) + (T \times C)$ number of constraints are introduced to optimize output matrix $\mathbf{X}$ attaining $N \times M \times T \times C$ number of 0-1 cells. Therefore, the objective function requires $(N \times M + N \times T + T \times C) \times 2^{N \times M \times T \times C}$ number of variable substitutions for the constraint check, denoting the run time complexity of the problem if the constraints are considered to be initially constructed.

Although the number of candidate solutions grows exponentially, eliminating some redundant variable substitutions is possible with the aid of some techniques such as branch and bound [104]. Besides, the values of the parameters are restricted by the requirements of a couple of neighboring clusters at most. The problem definition is feasible with the ability to serve multicast and broadcast links by taking into account link demands and previous link reservations.

3.2 Inter-Cluster Resource Sharing Algorithm

In a clustered network topology, each cluster head resolves management issues of its own cluster without an external contributor i.e. a central node or neighboring cluster head. Besides, sharing a TDMA source with a neighboring cluster is not applicable due to the interference. Therefore, each cluster should use a different type of source to maintain distributed link scheduling management and prevent inter-cluster interference.

Assigning each cluster to a different type of TDMA resource is not possible since the available TDMA resource blocks are limited. In the model proposed in this thesis, distributing communication resources so that no two neighboring clusters have the same set of resources is sufficient to prevent inter-cluster collisions. This objective can be addressed to a well-known graph coloring problem. The environment of the proposed resource sharing algorithm has similar aspects with that of distributed graph coloring algorithms investigated in 2.2.3.1 yet, even if labeled by distinct cluster heads ids, clusters in the network may not match consecutive numbers, unlike the ids that are assigned to each node of LOCAL mode graphs. Therefore, a resource sharing algorithm is proposed as a solution for the TDMA resource distribution among clusters. In contrast to the deterministic distributed graph coloring algorithms, this work has a basic heuristic approach in assigning resource blocks to clusters.

Initially, TDMA resource is divided into P number of disjoint resource parts. $P = \Delta + 1$ is the ideal case where Δ is the maximum number of neighboring clusters in the network topology. In the first round, the generated resource blocks are distributed to the clusters according to the Ids of each cluster head. Possible conflicts between neighboring clusters are resolved by simply giving the priority to the cluster with the smaller Id and choosing another available resource blocks for the other neighboring cluster. The execution of the algorithm is clarified in the following steps:

1. Initial resource part of a cluster is found in the following way:
   - $R(Cl_i) = Id(ch_i) \mod (P)$ where $R(Cl_i)$ denotes the resource part index of the cluster i, ch_i is a cluster head Id of cluster i. In this case, since the maximum number of neighboring clusters are estimated as $P - 1$, P is given as the parameter to $\mod$ function.
2. If the candidate resource part announced by this cluster head is not announced by the neighboring cluster heads, then the resource is taken and announced by the cluster head as its own.
3. If two neighboring cluster heads have the same resource part then the conflict is resolved in the following strategy:
   - The cluster head with smaller Id takes the resource and announce the resource part to the neighboring clusters as its own.
   - The other neighbor selects another type of resource that is not used by any of the neighboring cluster heads and announce it to the neighboring cluster head as a candidate resource part.
   - Go to Step 2.

As clusters collect announced resource parts of two-hop neighbors, conflicted cluster heads choose the neighboring resource part of the conflicted neighbors. The resource

Algorithm 1: Inter-cluster resource sharing algorithm.

```
AvailableRP = AllRPs;
CandidateRP = MapFunc(CHID);
AnnounceCandidate(CHID, CandidateRP);
foreach AllRP do
    round ++;
    ConflictedNeighbors = { };
    CollectedNeighbors = CollectNeighbors();
    foreach n ∈ CollectedNeighbors do
        if IsAnnounced(n) then
            AnnouncedNeighbors.add(n);
            AvailableRPs -= RP(n);
        if RP(n) == CandidateRP then
            ConflictedNeighbors.add(n);
    if ConflictedNeighbors == {} or CHID < min ( IDs(ConflictedNeighbors) )
    then
        AnnounceRP(CandidateRP);
        break;
    else
        if round == 1 then
            CandidateRP = Random(RP(AvailableRPs));
        else
            foreach n ∈ ConflictedNeighbors do
                if ID(n) < CHID then
                    foreach (m ∈ Neighbors(AnnouncedNeighbors(n)) or m ∈
                    Neighbors(ConflictedNeighbors(n))) and ID(m) != CHID
                    do
                        if RP(m) ∈ AvailableRPs then
                            AvailableRPs[RP(m)].point++;
            CandidateRP = MaxPoint(AvailableRPs);
        AnnounceCandidate(CHID, CandidateRP);
```

sharing algorithm is given in Algorithm 1. The complexity of the algorithm is $O(P)$, as each cluster sends candidate resource at most the number of neighboring cluster size.

3.2.1 Scenarios

The execution of the algorithm is illustrated as the following with three different scenarios in which 6 disjoint resources are determined, $P = 6$, and mapping function is $\mod (6)$ in common:

3.2.1.1 Scenario I: Single Round, No Spatial Reuse

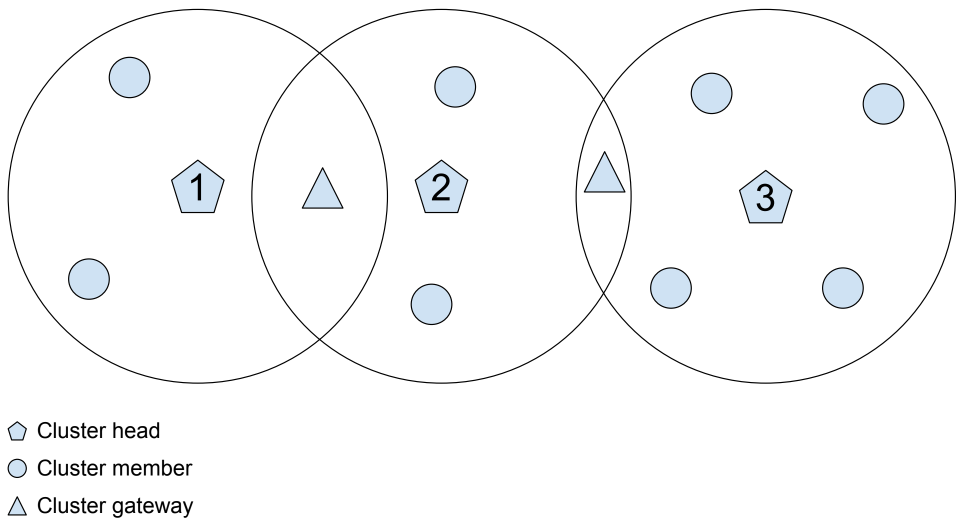
Figure 3.6a is the illustration of the network in Scenario I as Figure 3.6b shows the clusters as icons indicating cluster head ids inside the circles and mapping function results of their cluster head id at the top of the table and that of neighboring cluster head ids at other cells of the table, next to each circle.

In Figure 3.6, there are 3 clusters with cluster head ids 1, 2, and 3. In this scenario, clusters choose resources 1, 2, and 3 from left to right. This arrangement does not lead to any conflict among neighbors. Thus, each cluster uses the initial candidate resource part without passing to the second round.

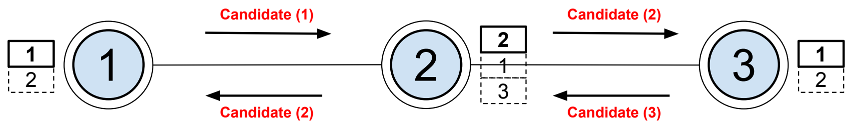
3.2.1.2 Scenario II: Single Round, Spatial Reuse

This scenario has 5 clusters with cluster head ids 1, 2, 3, 4, and 7. The topology of the network is shown in Figure 3.7. In the first round, clusters are mapped to resource part indices 1, 2, 3, 4, ve 1 from left to right. Since due to lack of conflict between resource part of any two neighbor clusters, the algorithm terminates in the first round similar to Scenario I.

Note that using resource part 1 in both the bottom and the top cluster is completely allowed as the resource distribution rule is not violated by assigning the same resource to one-hop neighbors.



(a)



(b)

Figure 3.6: Inter-cluster resource distribution scenario I, (a) node view, (b) cluster view.

3.2.1.3 Scenario III: Multiple Rounds, Spatial Reuse

Figure 3.8 shows a clustered ad hoc network including 7 clusters with cluster head ids 4, 5, 8, 10, 14, 16, and 22. In the first round, the clusters are mapped to resource part indices 4, 5, 2, 4, 2, 4, and 4 in order and send candidate messages to each neighbor cluster(s).

Candidate messages received from the neighboring clusters show that clusters with cluster head ids 5, 8, and 14 are mapped to candidate resource parts compatible with the surrounding clusters and thereby finishing the execution of the algorithm. Clusters with cluster head id 4 and 16 end up with the initial candidate resource parts as well,

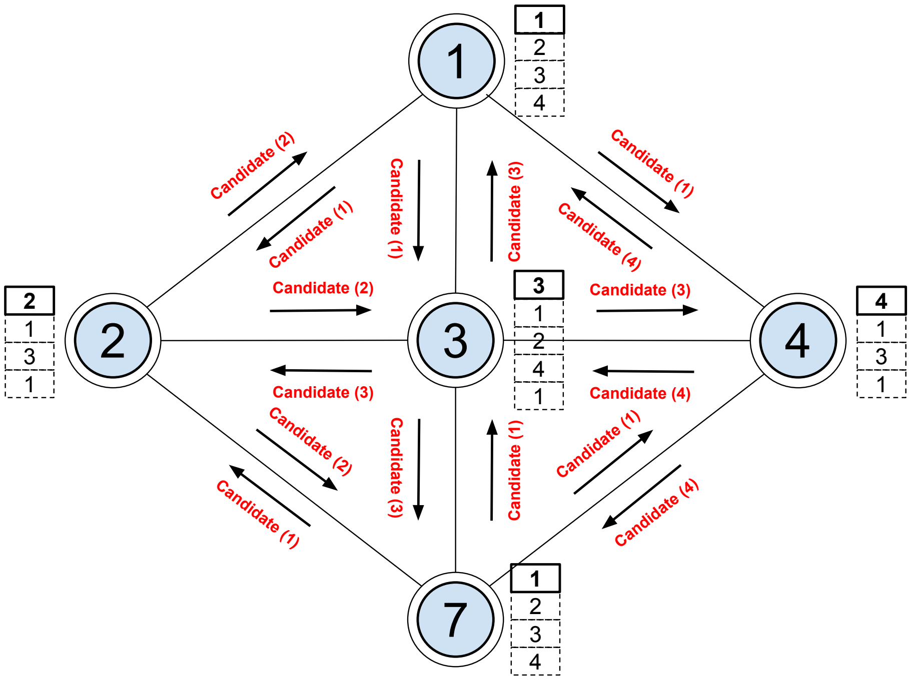
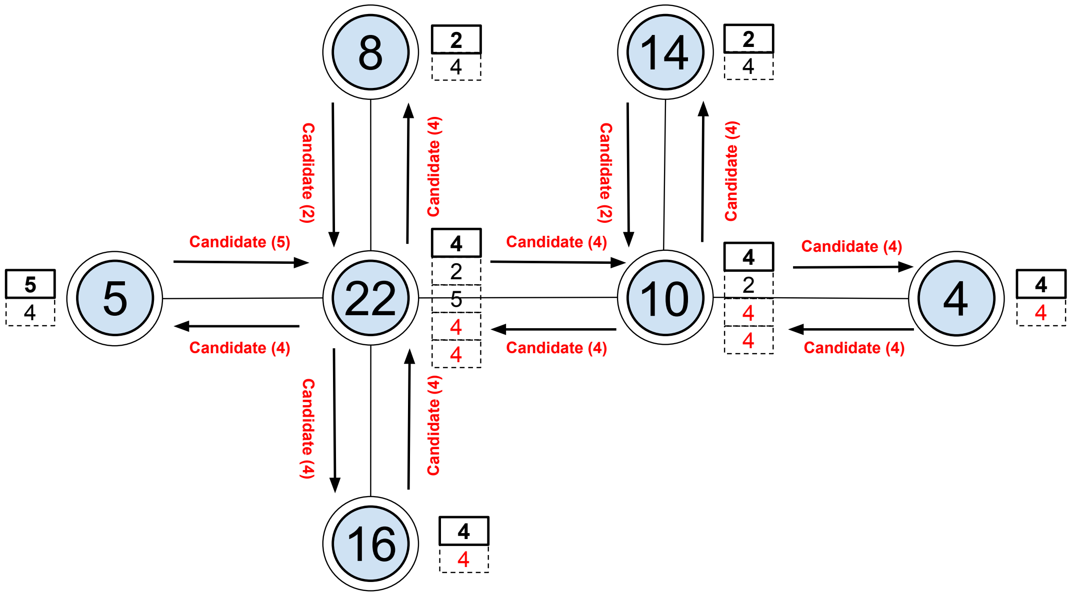


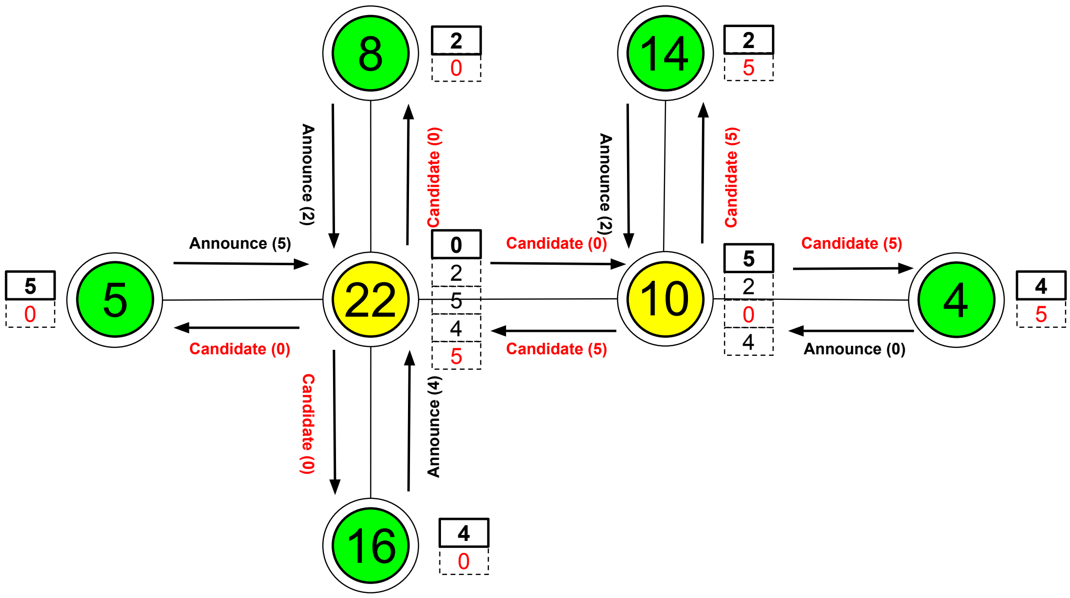
Figure 3.7: Inter-cluster resource distribution scenario II.

as the cluster head ids of these clusters are less than those of neighboring clusters mapped to the same resource parts, namely, clusters with cluster head ids 10 and 22, respectively. On the other hand, the execution of the algorithm continues in the remaining clusters.

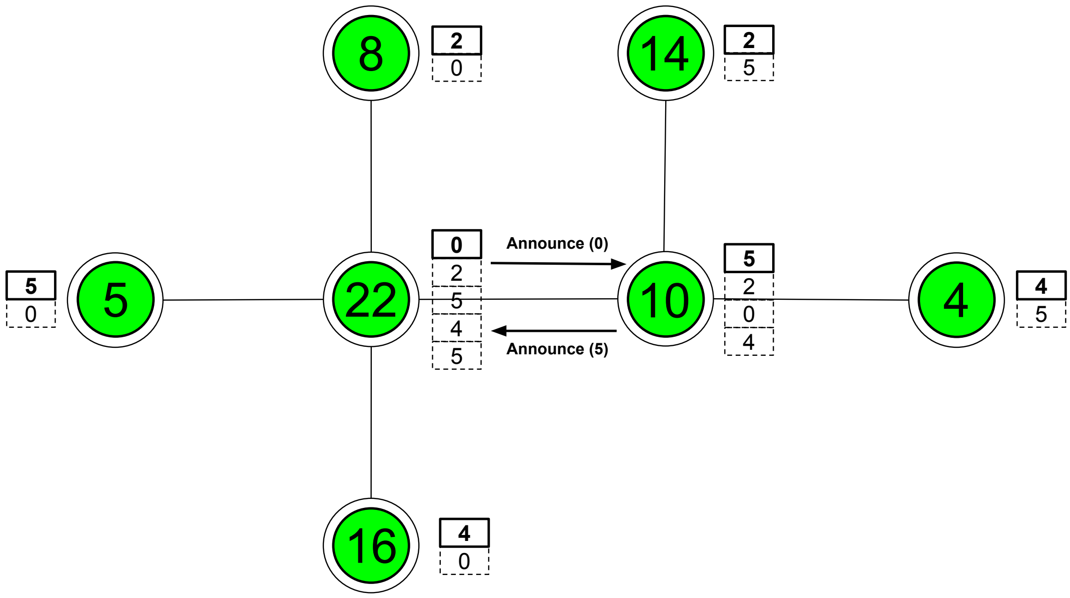
In the second round, the clusters with cluster head ids 10 and 22 randomly choose available resource parts 0 and 5, respectively and send candidates to the neighboring clusters as demonstrated in Figure 3.8b. Both resource part selection does not cause collusion among the neighboring clusters. Therefore, the resource assignment is completed in the whole network and the final state of the resource distribution process is shown in Figure 3.8c.


8
14
5
22
10
4
16
Candidate (2)
Candidate (4)
Candidate (5)
Candidate (4)
Candidate (4)
Candidate (4)
Candidate (2)
Candidate (4)
Candidate (4)
Candidate (4)
Candidate (4)
Candidate (4)

(a)


8
14
5
22
10
4
16
Announce (2)
Candidate (0)
Announce (5)
Candidate (0)
Candidate (0)
Announce (4)
Candidate (0)
Candidate (5)
Announce (2)
Candidate (5)
Candidate (5)
Announce (0)

(b)



(c)

Figure 3.8: States of inter-cluster resource distribution scenario III, (a) initial candidates, (b) at the end of round 1, and (c) at the end of round 2.

CHAPTER 4

EXPERIMENTAL RESULTS

This chapter reports outputs of the simulation environment of both proposed techniques and reviews simulation results in terms of several performance metrics and real-life scenarios. Furthermore, Section 4.1 tests two different input configurations for ICLS as well as compares the resulting performance metrics of both configurations.

4.1 ICLS Results

The scheduling problem introduced in Section 3.1.5 is interpreted as an Integer Linear Optimization problem and solved in MATLAB Optimization Toolbox [105] on a computer with Intel® Xeon® Silver 2.20 GHz with 40 CPU and 62 GB RAM[1]. The results of the MATLAB implementation are shown and discussed in this section.

Unicast, multicast, and broadcast messages are randomly created in a link demand matrix following a discrete uniform distribution. Various combinations of N, M, T, and C parameters are given as an input ranging from 4 to 10. Each value in the figures is obtained by taking the average of at most 100 simulation runs with a 95% confidence interval. The objective function parameters are modified so that resource blocks are prioritized inversely proportional to slot and channel numbers in order to improve *run time* metric of the optimizer as formulated in (3.3) and (3.4).

Two metrics are introduced to assess results: *Satisfaction* is the ratio of the served link demands to the total number of input link demands. *Efficiency* is a performance

[1] At most half of the computing resources are utilized and each simulation is run concurrently.

measure indicating the utility of resource blocks, which is the metric that is optimized by ICLS, and it is calculated by dividing the number of reserved link demands by the total number of resource blocks, $R = T \times C$.

Moreover, the normalized demand load, l,

$$l = \frac{\sum_{i=1}^{N} \sum_{m=1}^{M} \sum_{j=1}^{N} \mathbf{D}_{imj}}{N \times M \times (N-1)}, \tag{4.1}$$

represents packet density of a case where a source node can target at most $N - 1$ number of nodes within a link demand as a broadcast message and each node can have at most M number of link demand in a frame. The normalized load, l is expressed as the total amount of destined nodes introduced by all link demands divided by the maximum number of target nodes that can be generated for that frame, which is $N \times M \times (N-1)$. $l = 0$ if none of the nodes requests a link request and $l = 1$ if all nodes demand M number of broadcast links in a frame. Note that l is simply a link demand parameter that corresponds to link demand density in a frame duration and refer to neither ratio of occupied slots in a frame nor the data rate of nodes.

We propose two different approaches to solve the scheduling problem: *holistic approach* simply attempts to devise an instant solution to the given demand on entire resource blocks and *iterative approach*, which exploits our history based heuristics by gathering link demands into groups of equal size, G, and solves scheduling problem of each partitioned link demands on available resource blocks some of which have reserved for previous link demand groups, D_is.

We discuss the Iterative ICLS use about the possible improvement of the algorithm performance in particular use cases. In order to evaluate Iterative ICLS under practical use, Iterative ICLS with a single iteration is performed by introducing additional terms, i.e., d, a, and l_f. d is the number of new link requests collected in a frame duration and l_f is the normalized frame load,

$$l_f = \frac{a}{N \times T}, \tag{4.2}$$

representing the capacity of a frame, where a is the total number of active (sender or receiver) nodes in all slots. Note that a frame is fully utilized if N number of nodes is in either sender or receiver role in each slot due to the constraints mentioned in Section 2.4. Similar to the l parameter, l_f is basically the total number of active nodes in each slot divided by its maximum value, $N \times T$. $l_f = 0$ if the frame is empty and $l = 1$ if the frame is totally reserved and unable to assign a new link request.

4.1.1 The Holistic Approach

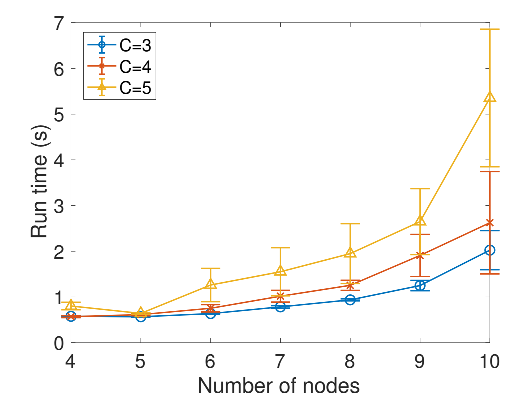
In this part, the algorithm is tested where all the link demands are given as input at once altogether. Initially, the history matrix is considered as an empty matrix. In other words, there is no resource block occupied in the previous frames and all link demands are to be placed into empty resource blocks in one run.

Figure 4.1 shows experimental results of holistic approach with a different number of nodes and packet densities. Two cases representing moderate and dense link demands are tested on fixed slot size $T = 16$ and 3 different channel sizes $C = 3, 4,$ and 5.[2] M is fixed to 6 for each holistic approach simulation and N has various values ranging from 4 to 10.

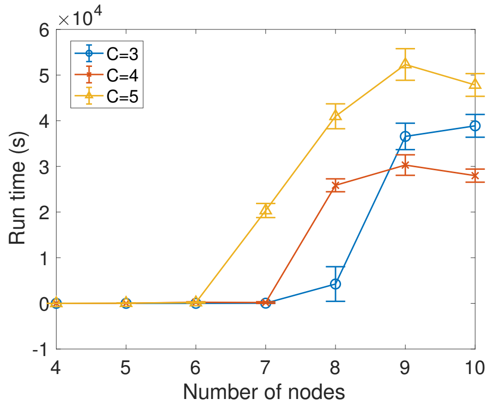
100 parallel simulation runs are performed for each normalized $l \times N$ case combination with different C values. At most 20 simulations are run simultaneously. The simulations which last more than twice of average run times are terminated and ignored in the calculation of performance results as outliers.

In Figure 4.1a it can be observed that average run time of the ICLS optimizer increases linearly up to around 5 seconds as number of nodes increases to 10 when $l = 0.10$ and $C = 5$. Figure 4.1b illustrates that in a dense link demand case where $l = 0.25$, as the number of nodes increases, average run time shows an exponential growth when the number of nodes is 9 and below. The run time of ICLS optimizer slightly increments or decrements when the number of nodes is 10 since after some point, available resource blocks cannot respond to increasing input link demand which eventually restricts the solution domain proportional to the number of nodes. However, optimizing link demands of 9 nodes when $l = 0.25$ and $C = 5$ takes about 25000 seconds, approximately 14 hours, on average which is extremely infeasible for scheduling of a single frame. In Figure 4.1a and Figure 4.1b, run time increases with C as the prob-

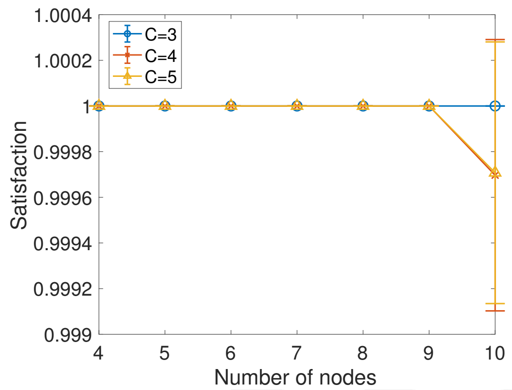
[2] Each slot can acquire $N/2$ number of links at most because a link occupied in a resource block eliminates at least two nodes from being active in the same slot. Therefore, T is chosen relatively greater than C in every simulation.



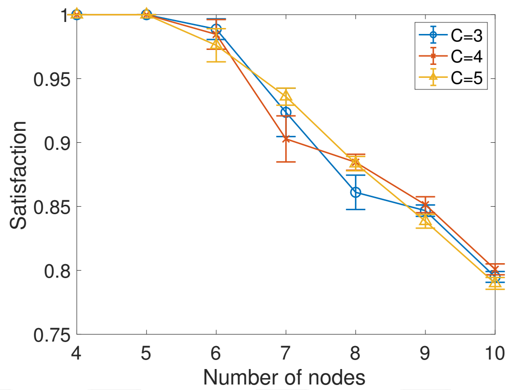
(a) Run time results where $l = 0.10$.



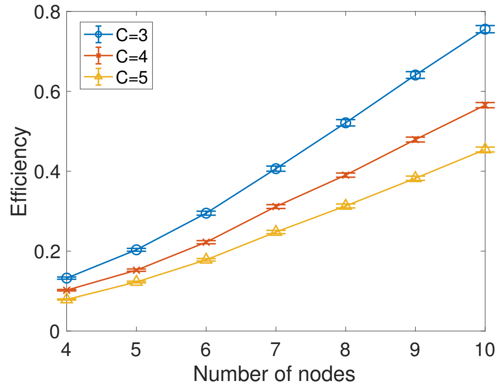
(b) Run time results where $l = 0.25$.



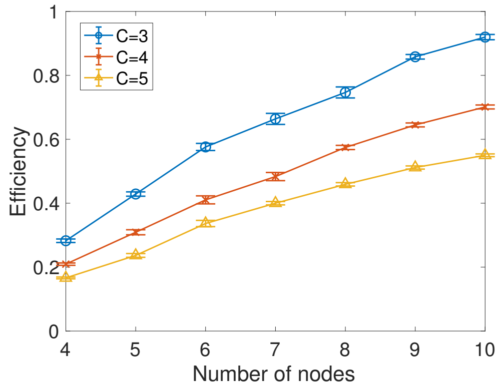
(c) Satisfaction results where $l = 0.10$.



(d) Satisfaction results where $l = 0.25$.



(e) Efficiency results where $l = 0.10$.



(f) Efficiency results where $l = 0.25$.

Figure 4.1: Holistic approach performances with $N = (4, 5, \ldots, 10)$, $M = 6$, $T = 16$, and $C = 3, 4,$ and 5.

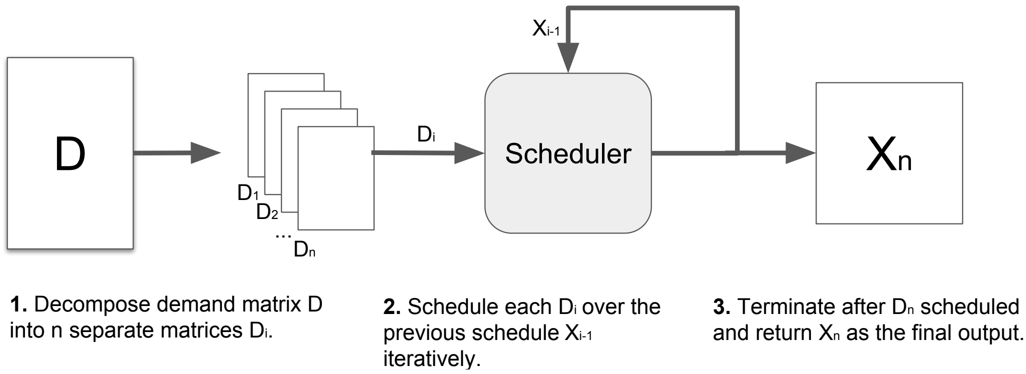


Figure 4.2: The steps of the Iterative approach.

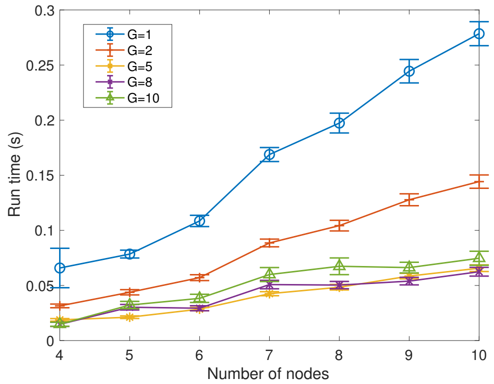
lem size grows with channel size except the demand traffic is dense and $N \geq 9$ case in which shortage of frame resources ($T \times C$ value) is prone to produce an excessive number of intermediate solutions in ICLS optimizer, thereby expands solution tree and prolongs the run time.

Figure 4.1c exhibits that almost all link demands are served by the ICLS scheduler in the moderate load case. In contrast, in Figure 4.1d satisfaction slightly decreases as the number of nodes increases to 10 for all C values.

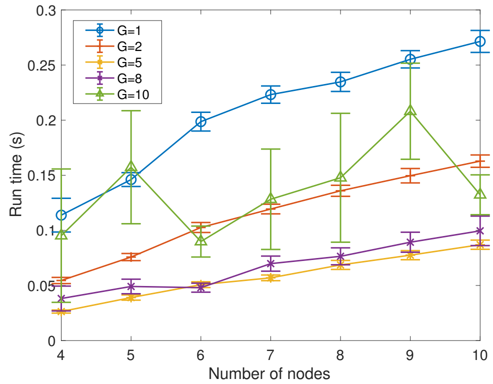
The efficiency exhibits a linear growth as the number of nodes increases for each case as illustrated in Figure 4.1e and Figure 4.1f. Average efficiency increases as the link demand gets denser and channel size shrinks, as well.

4.1.2 The Iterative Approach

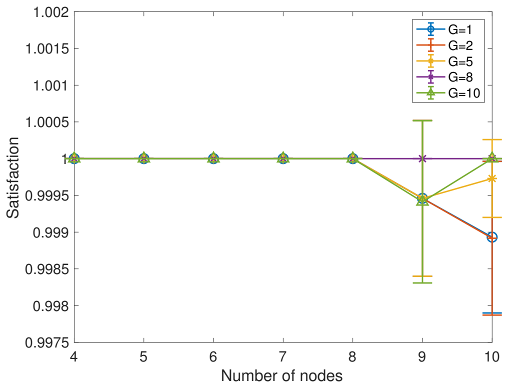
The ICLS optimizer with holistic approach is not able to support the scenarios in which the input demand matrix and available resources are relatively larger. Therefore, heuristics are required for optimizing the cases where the problem size is relatively large. Extending history approach to the optimization process is considered to resolve this issue. More specifically, dividing the link demands into separate groups and optimizing these sets of link demands by keeping the previous optimization results as history at each step aid to overcome the parameter size limitation. In Figure 4.2, the iterative approach steps are illustrated.



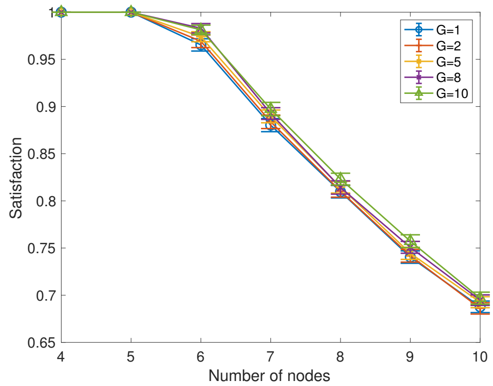
(a) Run time results where $l = 0.10$.



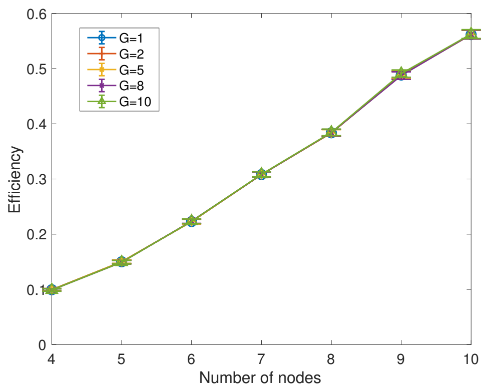
(b) Run time results where $l = 0.25$.



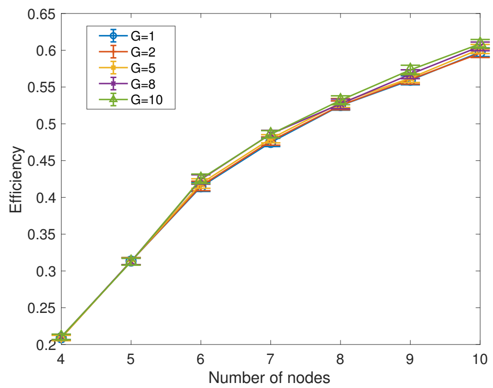
(c) Satisfaction results where $l = 0.10$.



(d) Satisfaction results where $l = 0.25$.



(e) Efficiency results where $l = 0.10$.



(f) Efficiency results where $l = 0.25$.

Figure 4.3: Iterative approach performances with $N = (4, 5, \ldots, 10)$, $M = 6$, $T = 16$, and $C = 4$.

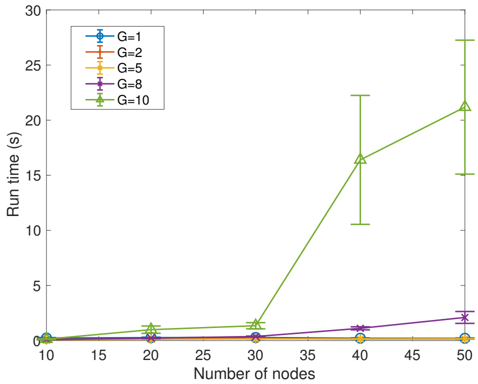
In Figure 4.3, a similar experimental setup in Figure 4.1 is constructed for iterative approach with the identical parameter set. Input link demands are partitioned into link demand groups with group sizes, $G = 1, 2, 5, 8,$ and 10. 100 separate link demands are generated for each $l \times N$ case combination and each link demand is ideally divided into the same-length link demand groups, D_is. Formally, $|D_i| = G$, where $i = 1, 2, \ldots, n-1$ and $|D_n| \leq G$. At each iteration, D_is are optimized and assigned to available resource blocks and previous allocations are given to optimizer as a history input.

In Figure 4.3a and Figure 4.3b, the run time performance of iterative approach is illustrated with 0.10 and 0.25 l values, respectively. In both Figure 4.3a and Figure 4.3b, link demand groups with $G = 1$ has the worst run time performances since assigning link demands one by one takes a relatively higher amount of time compared to handling less number of link demand groups with greater sizes. In contrast, Figure 4.3b, the link demand group with $G = 10$ has the second highest run time as greater link demand groups combined with high l values lead to an increase in the complexity of optimization problem. For every l case, run time increases with l value and cases in which $G = 5$ outperform the cases with the other link demand sizes.

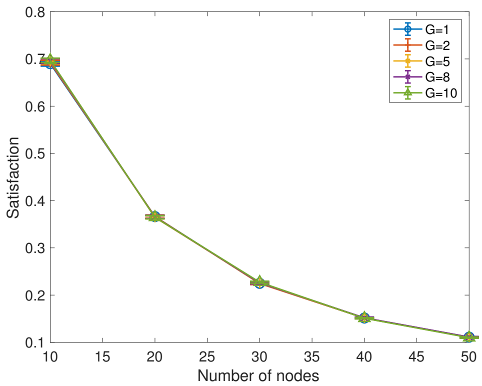
Satisfaction results depicted in Figure 4.3c show that almost all link demands are fully served in moderate demand load cases. In Figure 4.3d, satisfaction rates drop linearly as number of nodes increases. All satisfaction values are nearly the same and do not differ with respect to the G value.

In Figure 4.3e, efficiency values increase proportional to number of nodes. In contrast, the slope in Figure 4.3f slightly degrades for the cases in which number of nodes values are greater than 5 as the satisfaction decreases in Figure 4.3d. In general, the efficiency increases with l and N values. Similar to the satisfaction results, efficiency results are not affected by the group size, G.

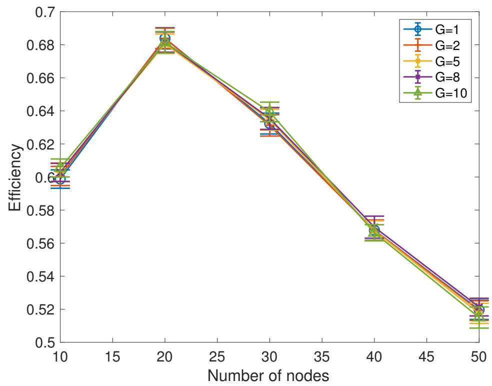
Figure 4.4 provides a further inspection of the performance of the iterative approach on scheduling over-populated clusters. Figure 4.4a shows that the iterative approach carries out scheduling process in a short period of time with particular G values, e.g., below 0.2 s where G is equal to 2 or 5. (Further analysis of G parameter is given in Section 4.1.4.) In Figure 4.4b, the decrease in satisfaction performance stems from



(a)



(b)



(c)

Figure 4.4: (a) runtime, (b) satisfaction, and (c) efficiency results of the iterative approach with $l = 0.25$, $N = (10, 20, \ldots, 50)$, $M = 6$, $T = 16$, and $C = 4$.

the concurrence of stable resource amount and the growth of the number of nodes. However, in Figure 4.4c the efficiency increases when the number of nodes is 20 then decreases since the fixed T number fails to meet resource requirements of an increasing number of destined nodes of each multicast links as N rises. Similar to the results in Figure 4.3, the number of resource blocks occupied does not rely on the G value.

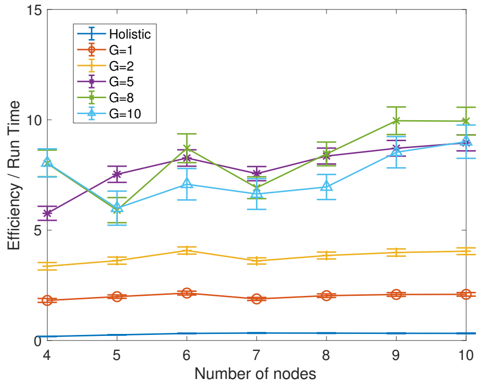
4.1.3 Comparison of Performance Assessments

In order to further investigate holistic and iterative approach results, a combined quality measure, which is efficiency divided by run time, has been applied to each simulation run in Figure 4.1 and Figure 4.3 and results are compared in Figure 4.5 with their corresponding efficiency values. Figure 4.5 shows that the holistic approach has the least and the iterative approach using link demand groups with $G = 5$ has the best quality measures for both moderate and dense link demand cases. It can be observed that the orders of quality assessment are similar among all link demand groups except the rank of link demand group with $G = 8$ and $G = 10$ in Figure 4.5a drops compared to dense link demand case illustrated in Figure 4.5c. Figure 4.5b and Figure 4.5d show that all Iterative ICLS results are as optimal as that of Holistic ICLS with a slight exception in Figure 4.5d when number of nodes is greater than 7. Compared to the holistic approach results in Figure 4.1 and Figure 4.5 the iterative approach shows nearly the same efficiency performance in substantially less amount of time with identical experimental setups.

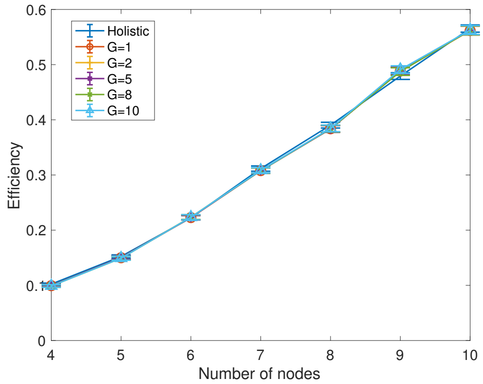
Figure 4.6 shows the performance assessments and efficiency values of another experiment that is devised by applying the same simulation run procedures with that of the experiments in Figure 4.5. In Figure 4.6a, increasing the normalized load, l, leads to a decrease in performance where $N > 5$. However, holistic approach and iterative approach using link demand groups with $G = 5$ yield to the worst and the best performance, respectively. In contrast, Figure 4.6b shows that Iterative ICLS has identical efficiency values for all G parameters. In addition, efficiency values for both Holistic and Iterative ICLS are nearly the same for $l \leq 0.2$ and close for $l > 0.2$ with a 0.07 (i.e., 7 percentage points in efficiency) difference at most. Therefore, it can be concluded that compared to Holistic ICLS, Iterative ICLS reaches near-optimal results with fewer time costs.

4.1.4 Group Size (G) Analysis

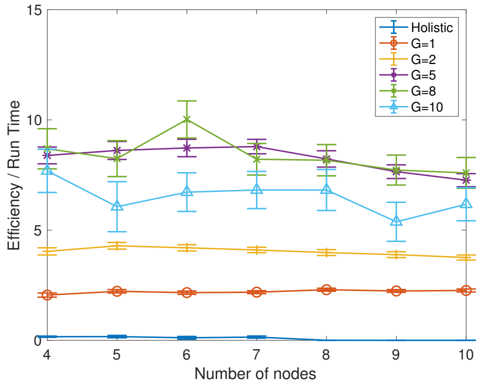
From previous simulations, it can simply be inferred that the performance results strictly depend on the parameter configuration. Apart from the input parameters such



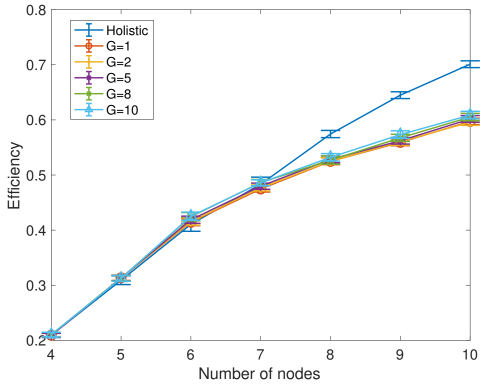
(a) Quality assessment where $l = 0.10$.



(b) Efficiency where $l = 0.10$.



(c) Quality assessment where $l = 0.25$.



(d) Efficiency where $l = 0.25$.

Figure 4.5: Quality assessment and efficiency of holistic and iterative approaches with $N = (4, 5, \ldots, 10)$, $M = 6$, $T = 16$, and $C = 4$.

as N, M, T, and C, which are static and adopted from the communication medium requirements before the beginning of the scheduling process, an implicit and adaptable parameter, the group size, G, has a direct influence on the performance of Iterative ICLS algorithm as addressed in Figure 4.3 prominently. Besides, an improper choice of G parameter may inflate the run time of Iterative ICLS, similar to the case in which run time is 20 s on average when $G = 10$ and $N = 50$ in Figure 4.4a. Therefore, it is essential to estimate the G parameter properly prior to the execution of the algorithm for the optimal scheduling performance.

In Figure 4.7, each parameter in the fixed parameter set $l = 0.25$, $N = 8$, $M = 6$, and $R = 64$ ($T = 16$ and $C = 4$) is set to 3 different values and tested with

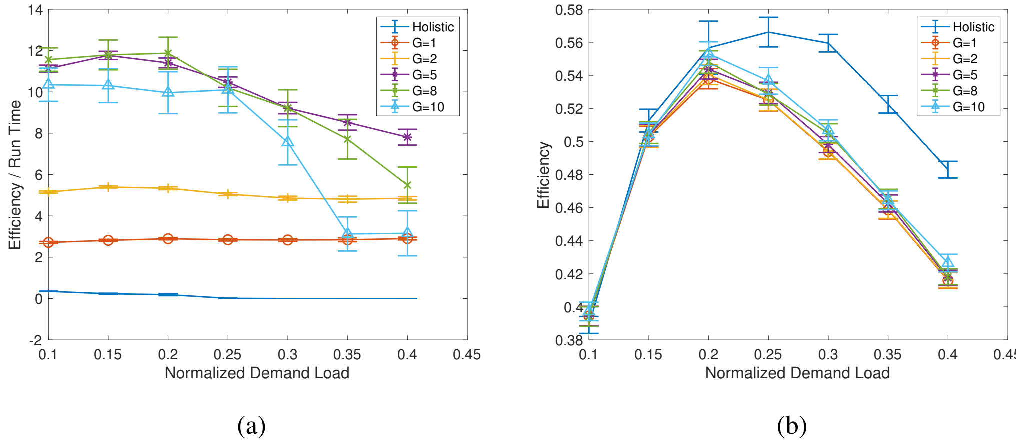
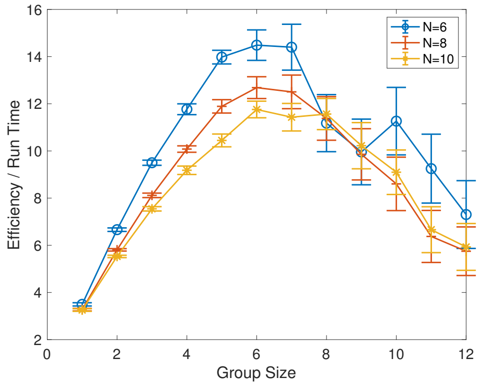


Figure 4.6: (a) quality assessment and (b) efficiency of holistic and iterative approaches with $l = (0.10, 0.25, \ldots, 0.40)$, $N = 8$, $M = 6$, $T = 16$, and $C = 4$.

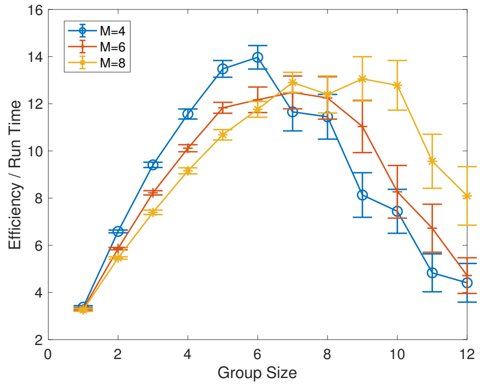
various G parameters in order to illustrate the impact of G on performance measures. Figure 4.7b shows that peak performances are correlated to the M value, as optimal G value is around $M = 6$ in Figure 4.7a, Figure 4.7c, and Figure 4.7d. It can be inferred that M is the most effective parameter on optimal G and choosing G close to M value leads iterative ICLS to work efficiently. The optimization process takes less amount of time when D_is have common nodes among the links, forcing the links to assign different slots and eliminating some of the cases that the algorithm tests. Selecting G close to M value leads to overlap D_is with D rows, causing links with the same source nodes reside in the same D_is, thereby increasing the run time performance.

4.1.5 Discussion on Iterative ICLS Use

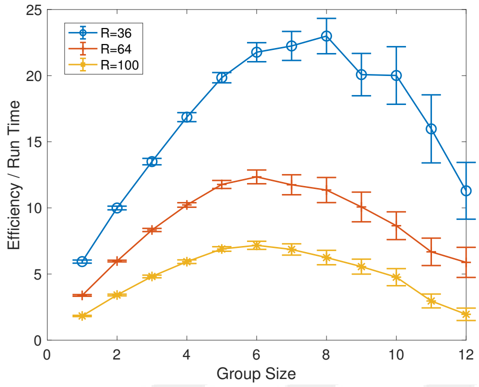
Iterative ICLS does not only significantly contribute to performance but also has some advantages in terms of flexibility as the algorithm can be adapted to various scenarios for boosting the scheduling process and solve some specific problems. For instance, when the frame size is large, D_is contain a relatively small amount of links, and the frame is empty, CSMA can be conducted or the target frame can be restricted for narrowing down problem space. Another enhancement can be achieved by adjusting the frequency of the scheduling process rather than running the scheduler for each



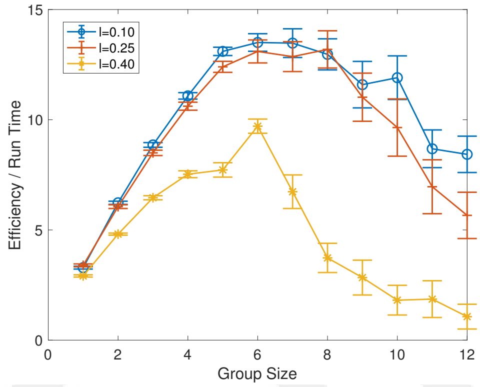
(a) Quality assessment with different N values.



(b) Quality assessment with different M values.



(c) Quality assessment with different R values.



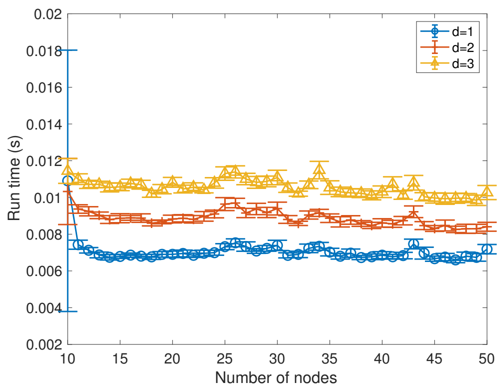
(d) Quality assessment with different l values.

Figure 4.7: The impact of group size on performance measures with different parameters.

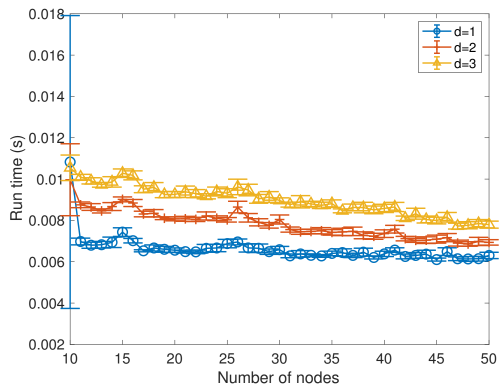
frame. Alternatively, some links or nodes can be reserved for the same resource blocks in consecutive frames as a hybrid solution. The variations of Iterative ICLS executions can be applied according to the needs of target clustered ad hoc networks and out of the scope of the study.

The inter-cluster scheduling conducted on gateway nodes, which is described as the schedule sent to neighboring cluster heads in the third step in Figure 3.4, can be simply conducted by Iterative ICLS by the following strategy: The inter-cluster link is scheduled in one Iterative ICLS iteration over the combination of both neighboring intra-cluster frames. A single iteration of a single Iterative ICLS leads to an immediate inter-cluster schedule by alleviating the effect of a large domain size of combined resource space on run time performance.

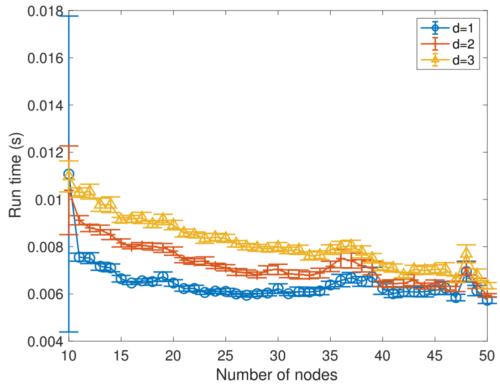
An iteration of Iterative ICLS is not useful only in inter-cluster links, but it is also capable of maintaining intra-cluster link scheduling during the network lifetime. D input with large l values that can occupy nearly all slots of an empty frame is not common, and is possible in a few particular cases only such as starting of the network. The total number of link demands newly arrived in a cluster head in a frame duration is expected to be less than that of new link demands within a frame in previous simulations, in which worst case l values are tested.



(a)



(b)



(c)

Figure 4.8: Run time performances of maintenance phase with (a) $l_f = 0.10$, (b) $l_f = 0.25$, and (c) $l_f = 0.40$ where $N = (10, 11, \ldots, 50)$, $T = 50$, and $C = 5$.

In Figure 4.8, run time performances of Iterative ICLS with a single iteration is tested with a few link request, demonstrating the expected input behavior in maintenance phase of Iterative ICLS scheduler. Given previously scheduled frames with 3 different l_f values 0.10, 0.25, and 0.40, recent d link demands are assigned to available resource blocks in the frame. Figure 4.8 shows that the Iterative ICLS responses in a reasonable period of time, around 10 ms, which is practical for handling a new request for a fully operational network. In Figure 4.8, for instance, given slot size of 4096 bits and slot time of 2 ms, providing up to 2 Mbps data rate and generating maximum of 30 link demands per second, the scheduling lasts less than 1 frame duration, $2 \times T = 100$ ms, for all d and N values. This slot time configuration also allows the scheduling of the 187.5 link requests per second in 2 frame times for the test case with $N = 10$ and $G = 8$ in Figure 4.3a, where recent link requests and all previously scheduled active links are scheduled simultaneously on an empty frame, unlike the maintenance phase.

The ICLS is tested up to 50 nodes with $T = 50$ slots. Results are ranging from 6 ms to 12 ms which is responsive to the scheduler needs and stable as N increases. The scheduler performance can further be increased if more lightweight optimizer alternatives are used.

4.2 Inter-Cluster Resource Sharing Results

The algorithm is implemented and Monte Carlo simulations are conducted in MATLAB. 500 nodes, each of which represents a cluster, are placed into 1×1 area with uniform random distribution for the simulation. The result values are the mean of 100 run results of the algorithm. The neighborhood among the nodes is defined with a coverage range. More specifically, if a node is in the given range of another node then these two nodes are accepted as neighbors. A node is labeled as *colored* if that node is assigned to a resource part.

The average number of maximum degree (maximum number of neighbors) are given in Table 4.1. It shows that the average number of maximum degree is proportional to the square of the range parameter.

Table 4.1: Average maximum degree with the corresponding node range parameters.

Range	**Average Maximum Degree**
0.05	10.34
0.10	26.56
0.15	49.45

Figure 4.9 illustrates the effect of estimated maximum degree (number of resource blocks) value given as input to the number of communication rounds needed to terminate the algorithm. Round cost refers to the average number of iteration of Algoritm 1 with the corresponding parameters. If the maximum degree is estimated as a small number, the algorithm terminates in a relatively small number of rounds as the algorithm conducts trials maximum degree times and fails to assign a resource part to nodes and terminates without finding an appropriate resource part. As the given maximum degree increases, the rounds increase up to some limit and gradually decrease as the algorithm succeeds to assign resource blocks with less number of rounds. However, the number of rounds increases when the given range is high.

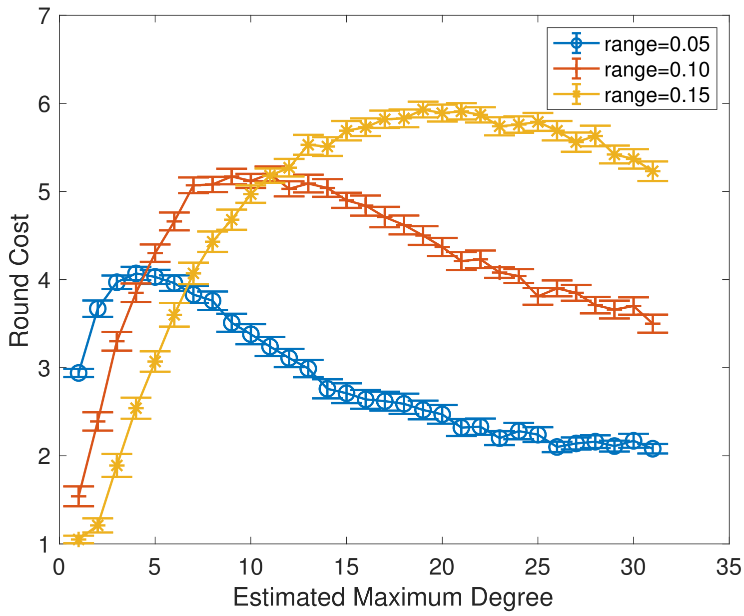


Figure 4.9: Maximum degree value given as input vs round cost.

Figure 4.10 shows the change in the number of non-colored nodes with respect to the given maximum degree. The results show that the number of failed nodes monotonically decreases as the given maximum degree increases. The number of neighboring clusters also affects the number of colored nodes, adversely.

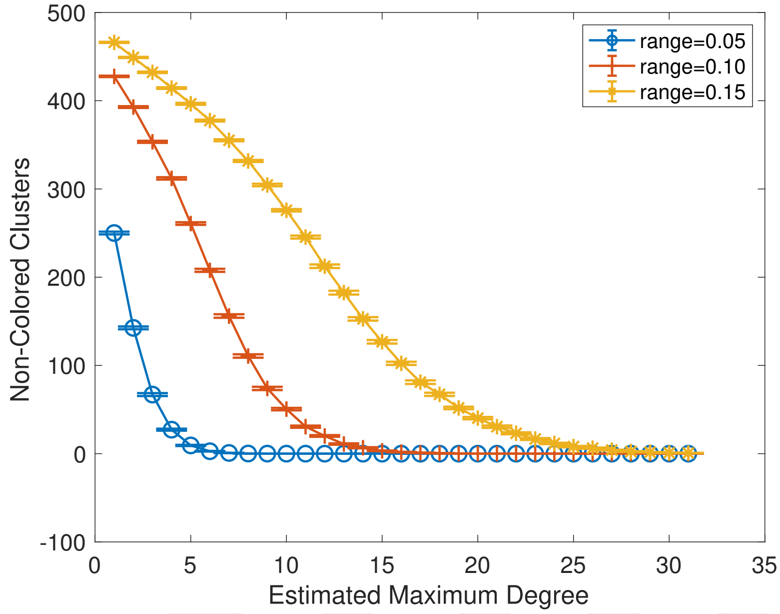


Figure 4.10: Maximum degree value given as input vs resulting non colored nodes.

In Figure 4.11, the probability of non-colored nodes less than 1% is shown as cumulative distribution function (CDF) of the number of non-colored nodes. It can be inferred that if 99% or higher success rate is desired, the critical points for the estimated maximum degree are around 5, 15, and 30 where input ranges are 0.05, 0.10, and 0.15, respectively.

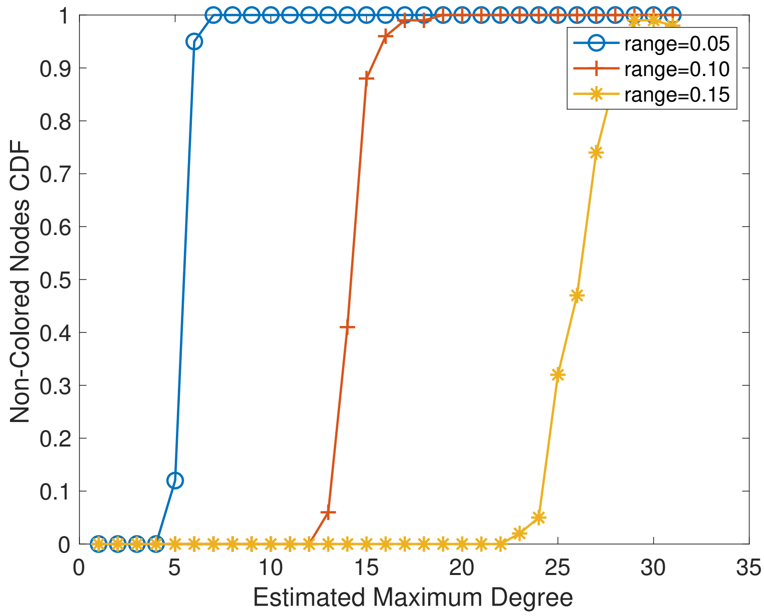


Figure 4.11: CDF of non-colored nodes less than 1%.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

This chapter summarizes the introduced tasks and accomplishments of this thesis and evaluates the results of the proposed methods in Section 5.1. In Section 5.2 further improvements and new research ideas for the thesis is discussed as future work.

5.1 Conclusions

Resource management in ad hoc networks, imposing some stringent requirements considering integrity and scalability, is maintained with an effective link scheduling algorithm compatible with the clustered topology. The cluster head nodes are determined as the controller nodes charged for resource management, offloading link scheduling tasks across the network while leading control data to be accessible to other nodes. The distribution of workload through cluster heads characterizes the control plane as a star topology within a cluster yet cluster heads can communicate with each other through gateway nodes. However, structuring the network with Control and User Plane Separation (CUPS) oriented architecture, the data plane may contain any link between any two nodes. That means, data transmission can flow in any route, preferably the shortest path, allowing a decrease in transmission time and data traffic over cluster heads by separating data and control planes.

Time Division Multiple Access (TDMA) which is a resource access solution fitting to reliable connection requirements is used with multi-channel support, enhancing the capacity of the communication resources as well as problem complexity significantly. Intra- and Inter-Cluster Link Scheduling Algorithm (ICLS) is first proposed for intra-cluster links and responsive to communication needs of the nodes including the links destined to multiple nodes: multicast and broadcast links thereby distinguished from the other counterpart algorithms. A practical method compatible with CUPS architecture is then introduced for scheduling communication between members of two neighboring clusters which are controlled by two cluster heads separately. Mutual agreement on inter-cluster link establishment is maintained with the assistance of a common gateway node that selects an inter-cluster link resource block regarding the current intra-cluster schedules of both clusters. Contribution of gateway node into scheduling decision process shortens control messaging path by accepting intra-cluster link schedules of both sides and responding with the common intra-cluster link decision, concurrently.

The problem formulation of ICLS on CUPS architecture is introduced and conflicted cases are presented for different scheduling scenarios. Link scheduling is formulated as an integer nonlinear optimization problem that maximizes the utilization of resource blocks in the resulting frame and reduced to its linear form for simplifying the formulation and discarding redundant computation.

In the proposed CUPS-based ad hoc network model, each cluster head is responsible for the enclosing cluster and intra-cluster link scheduling is decided without a common agreement or a centralized contributor. The autonomy of the intra-cluster scheduling requires the assignment of two distinct resource spaces that can be utilized simultaneously to any two neighboring clusters when avoidance of inter-cluster interference is targeted. In order to distribute communication resources into clusters, the available channel spectrum is divided into disjoint channel parts. As the channel spectrum is not sufficient to assign a distinct resource part to each cluster, a resource sharing algorithm supporting spatial reuse is presented with execution steps and illustrative scenarios.

In order to validate ICLS, a set of Monte Carlo simulations are conducted and the performance analysis of different parameter combinations is discussed. The results show that the time complexity of the optimization problem exponentially grows by the node size. In addition, the optimization requires a vast amount of calculation even in small parameter sizes. However, dividing link demands into separate link groups with the size of the maximum number of input link request for a node then optimizing iteratively by using the previous optimization results as history (Iterative ICLS) is a basic yet effective strategy and yields to rapid and efficient results as in this case link demand of a node is mostly collected in an input demand group leading a source node to become common in each request thereby decreasing the search space of the optimizer in the algorithm. Iterative ICLS is tested on maintenance phase conditions and it is stated that Iterative ICLS is responsive to the scheduler needs and stable as the input size increases.

Finally, inter-cluster resource distribution is carried out with a collective exchange mechanism managed by cluster heads. The results show that the algorithm achieves a proper inter-cluster resource assignment if the maximum degree is predicted adequately.

5.2 Future Work

ICLS is open to new research opportunities in several aspects. One of the main topics for future enhancements is resource optimization. A study can be conducted for eliminating or diminishing one or more dimensions of the problem size. Further constraints can be introduced to the optimization process, e.g., introducing a coefficient indicating channel quality.

Some refinements will be applied to the algorithm for reaching a more scalable solution. For instance, apart from the impact of a single parameter, a comprehensive strategy for composing D_i inputs optimizing run time performance of Iterative ICLS will be presented. Moreover, ICLS will be tested with other optimizers that are faster and compatible with the scheduling task. In order to obtain performance results more relevant to real-life conditions, the algorithm will run in a network simulator on a complete clustered network setup with a clustering method, routing algorithm, and an optimizer adapted to the simulator environment in which the proposed inter-cluster link scheduling mechanism and resource sharing algorithm can also be integrated to the ICLS.

A prediction mechanism for the maximum degree input of the inter-cluster resource distribution algorithm will be proposed. The resource distribution algorithm will also be compared to a state-of-the-art opponent algorithm.

REFERENCES

[1] M. Y. Arafat and S. Moh, "Routing protocols for unmanned aerial vehicle networks: A survey," *IEEE Access*, vol. 7, pp. 99694–99720, 2019.

[2] J. Jagannath, S. Furman, A. Jagannath, L. Ling, A. Burger, and A. Drozd, "Helper: Heterogeneous efficient low power radio for enabling ad hoc emergency public safety networks," *Ad Hoc Networks*, vol. 89, pp. 218 – 235, 2019.

[3] S. Ullah, K.-I. Kim, K. H. Kim, M. Imran, P. Khan, E. Tovar, and F. Ali, "Uav-enabled healthcare architecture: Issues and challenges," *Future Generation Computer Systems*, vol. 97, pp. 425 – 432, 2019.

[4] C. Cooper, D. Franklin, M. Ros, F. Safaei, and M. Abolhasan, "A comparative survey of VANET clustering techniques," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 657–681, 2017.

[5] G. V. Rossi, Z. Fan, W. H. Chin, and K. K. Leung, "Stable Clustering for Ad-Hoc Vehicle Networking," in *Proc. of the WCNC*, pp. 1–6, 2017.

[6] M. Adler, R. K. Sitaraman, A. L. Rosenberg, and W. Unger, "Scheduling time-constrained communication in linear networks," in *Proceedings of the tenth annual ACM symposium on Parallel algorithms and architectures*, pp. 269–278, ACM, 1998.

[7] P. Ruiz and P. Bouvry, "Survey on broadcast algorithms for mobile ad hoc networks," *ACM Computing Surveys (CSUR)*, vol. 48, no. 1, pp. 1–35, 2015.

[8] "How to: determine party size for mountaineer trips." `https://www.mountaineers.org/blog/how-to-determine-party-size-for-mountaineers-trips`. Accessed: 2020-07-20.

[9] "Modern military force structures." `https://www.cfr.org/`

backgrounder/modern-military-force-structures. Accessed: 2020-07-20.

[10] D. Ergenç, L. Eksert, and E. Onur, "Dependability-based clustering in mobile ad-hoc networks," *Ad Hoc Networks*, vol. 93, p. 101926, 2019.

[11] D. Ergenç and E. Onur, "Cupsman: Control user plane separation based routing in ad-hoc networks," *ArXiv*, vol. abs/1807.10747, 2018.

[12] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot, "Optimized link state routing protocol for ad hoc networks," in *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century.*, pp. 62–68, 2001.

[13] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *Proceedings WMCSA'99. Second IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, 1999.

[14] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: challenges and directions," *IEEE Communications Magazine*, vol. 40, pp. 20–22, 2002.

[15] J. Loo, J. L. Mauri, and J. H. Ortiz, *Mobile ad hoc networks: current status and future trends*. CRC Press, 2016.

[16] X. Zhang, J. Liang, and L. Zhang, "Delay-constrained streaming in hybrid cellular and cooperative ad hoc networks," *Computer Communications*, vol. 118, pp. 205 – 216, 2018.

[17] J. Shiral, "A comparative study on cellular, sensor and adhoc networks," *International Journal of Advanced Smart Sensor Network Systems*, vol. 2, 07 2012.

[18] M. Hollick, I. Martinovic, T. Krop, and I. Rimac, "A survey on dependable routing in sensor networks, ad hoc networks, and cellular networks," in *Proceedings. 30th Euromicro Conference, 2004.*, pp. 495–502, 2004.

[19] S. A. K. Omari and P. Sumari, "An overview of mobile ad hoc networks for the existing protocols and applications," *International Journal on Applications of Graph Theory In wireless Ad Hoc Networks And sensor Networks*, vol. 2, p. 87–110, Mar 2010.

[20] H. Wymeersch and A. Eryilmaz, "Chapter 12 - multiple access control in wireless networks," in *Academic Press Library in Mobile and Wireless Communications* (S. K. Wilson, S. Wilson, and E. Biglieri, eds.), pp. 435 – 465, Oxford: Academic Press, 2016.

[21] A. Grami, "Chapter 11 - communication networks," in *Introduction to Digital Communications* (A. Grami, ed.), pp. 457 – 491, Boston: Academic Press, 2016.

[22] V. K. Garg and Y.-C. Wang, "7 - wireless network access technologies," in *The Electrical Engineering Handbook* (W.-K. CHEN, ed.), pp. 1005 – 1009, Burlington: Academic Press, 2005.

[23] S. Vodopivec, J. Bešter, and A. Kos, "A survey on clustering algorithms for vehicular ad-hoc networks," in *Proc. of the 35th International Conference Telecommunications and Signal Processing (TSP)*, pp. 52–56, IEEE, 2012.

[24] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer Communications*, vol. 30, no. 14-15, pp. 2826–2841, 2007.

[25] O. Boyinbode, H. Le, and M. Takizawa, "A survey on clustering algorithms for wireless sensor networks," *International Journal of Space-Based and Situated Computing*, vol. 1, no. 2-3, pp. 130–136, 2011.

[26] A. Ephremides, J. E. Wieselthier, and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987.

[27] M. Gerla and J. T.-C. Tsai, "Multicluster, mobile, multimedia radio network," *Wireless networks*, vol. 1, no. 3, pp. 255–265, 1995.

[28] S. Basagni, "Distributed clustering for ad hoc networks," in *Proc. of the Fourth International Symposium of Parallel Architectures, Algorithms, and Networks (I-SPAN'99)*, pp. 310–315, IEEE, 1999.

[29] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of*

the 33rd Annual Hawaii International Conference on System Sciences, pp. 10–pp, IEEE, 2000.

[30] O. Younis and S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 3, pp. 366–379, Oct 2004.

[31] D. Ergenç, L. Eksert, and E. Onur, “Density-Aware Probabilistic Clustering in Ad Hoc Networks,” in *Proc. of IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, pp. 1–5, June 2018.

[32] A. K. Parekh, “Selecting routers in ad-hoc wireless networks,” in *Proceedings of the SBT/IEEE International Telecommunications Symposium*, vol. 204, Rio de Janeiro (Brazil), 1994.

[33] Y. Kim, K.-Y. Jung, T.-H. Kim, and J. Kim, “A distributed energy-efficient clustering scheme for deploying ids in MANETs,” *Telecommunication Systems*, vol. 52, no. 1, pp. 85–96, 2013.

[34] A. Karimi, A. Afsharfarnia, F. Zarafshan, and S. Al-Haddad, “A novel clustering algorithm for mobile ad hoc networks based on determination of virtual links’ weight to increase network stability,” *The Scientific World Journal*, vol. 2014, 2014.

[35] J. J. Blum, A. Eskandarian, and L. J. Hoffman, “Challenges of intervehicle ad hoc networks,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no. 4, pp. 347–351, 2004.

[36] P. Basu, N. Khan, and T. D. Little, “A mobility based metric for clustering in mobile ad hoc networks,” in *Proc. of the International Conference on Distributed Computing Systems Workshop*, pp. 413–418, IEEE, 2001.

[37] E. Dror, C. Avin, and Z. Lotker, “Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks,” in *Proc. of the 10th IFIP Annual Mediterranean Hawaii international conference Ad Hoc Networking Workshop (Med-Hoc-Net)*, pp. 1–8, IEEE, 2011.

[38] M. M. C. Morales, C. S. Hong, and Y.-C. Bang, “An adaptable mobility-aware clustering algorithm in vehicular networks,” in *Proc. of the 13th Asia-Pacific*

Network Operations and Management Symposium (APNOMS), pp. 1–6, IEEE, 2011.

[39] Y. Chen, M. Fang, S. Shi, W. Guo, and X. Zheng, "Distributed multi-hop clustering algorithm for VANETs based on neighborhood follow," *Eurasip Journal on Wireless Communications and Networking*, vol. 2015, no. 1, p. 98, 2015.

[40] S. Ucar, S. C. Ergen, and O. Ozkasap, "Multihop-cluster-based IEEE 802.11 p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 4, pp. 2621–2636, 2016.

[41] Z. Yang, W. Wu, Y. Chen, X. Lin, and X. Chen, "Navigation route based stable clustering for vehicular ad hoc networks," *China Communications*, vol. 15, no. 3, pp. 42–56, 2018.

[42] M. Chatterjee, S. K. Das, and D. Turgut, "WCA: A weighted clustering algorithm for mobile ad hoc networks," *Cluster computing*, vol. 5, no. 2, pp. 193–204, 2002.

[43] W.-d. Yang and G.-z. Zhang, "A weight-based clustering algorithm for mobile ad hoc network," in *Proc. of the Third International Conference on Wireless and Mobile Communications (ICWMC'07(*, pp. 3–3, IEEE, 2007.

[44] F. Belabed and R. Bouallegue, "An optimized weight-based clustering algorithm in wireless sensor networks," in *Proc. of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 757–762, IEEE, 2016.

[45] S. K. Dhurandher and G. Singh, "Stable clustering with efficient routing in wireless ad hoc networks," in *2007 2nd International Conference on Communication Systems Software and Middleware*, pp. 1–12, IEEE, 2007.

[46] W. Shahzad, F. A. Khan, and A. B. Siddiqui, "Clustering in mobile ad hoc networks using comprehensive learning particle swarm optimization (CLPSO)," in *Communication and Networking*, pp. 342–349, Springer, 2009.

[47] H. Ali, W. Shahzad, and F. A. Khan, "Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization," *Applied Soft Computing*, vol. 12, no. 7, pp. 1913–1928, 2012.

[48] F. Aadil, K. B. Bajwa, S. Khan, N. M. Chaudary, and A. Akram, "CACONET: ant colony optimization (ACO) based clustering algorithm for VANET," *PloS one*, vol. 11, no. 5, p. e0154080, 2016.

[49] M. Fahad, F. Aadil, S. Khan, P. A. Shah, K. Muhammad, J. Lloret, H. Wang, J. W. Lee, I. Mehmood, *et al.*, "Grey wolf optimization based clustering algorithm for vehicular ad-hoc networks," *Computers & Electrical Engineering*, 2018.

[50] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane, "A multi-objective genetic algorithm-based adaptive weighted clustering protocol in VANET," in *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pp. 994–1002, IEEE, 2015.

[51] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[52] A. Sgora, D. J. Vergados, and D. D. Vergados, "A survey of TDMA scheduling schemes in wireless multihop networks," *ACM Computing Surveys (CSUR)*, vol. 47, no. 3, p. 53, 2015.

[53] P. Huang, L. Xiao, S. Soltani, M. W. Mutka, and N. Xi, "The evolution of MAC protocols in wireless sensor networks: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 101–120, 2013.

[54] M. Sun, L. Zhao, W. Cao, Y. Xu, X. Dai, and X. Wang, "Novel hysteretic noisy chaotic neural network for broadcast scheduling problems in packet radio networks," *IEEE Transactions on Neural Networks*, vol. 21, no. 9, pp. 1422–1433, 2010.

[55] R. Gunasekaran, S. Siddharth, P. Krishnaraj, M. Kalaiarasan, and V. R. Uthariaraj, "Efficient algorithms to solve broadcast scheduling problem in WiMAX mesh networks," *Computer Communications*, vol. 33, no. 11, pp. 1325–1333, 2010.

[56] Y. Liu, V. O. Li, K.-C. Leung, and L. Zhang, "Topology-transparent distributed

multicast and broadcast scheduling in mobile ad hoc networks," in *Vehicular Technology Conference (VTC Spring), 2012 IEEE 75th*, pp. 1–5, IEEE, 2012.

[57] X. Zhang, J. Hong, L. Zhang, X. Shan, and V. O. Li, "CC-TDMA: Coloring- and coding-based multi-channel TDMA scheduling for wireless ad hoc networks," in *Wireless Communications and Networking Conference, 2007. WCNC 2007. IEEE*, pp. 133–137, IEEE, 2007.

[58] X. Zhang, J. Hong, L. Zhang, X. Shan, and V. O. Li, "CP-TDMA: Coloring- and probability-based TDMA scheduling for wireless ad hoc networks," *IEICE Transactions on Communications*, vol. 91, no. 1, pp. 322–326, 2008.

[59] Z. Wang, F. Yu, J. Tian, and Z. Zhang, "A fairness adaptive TDMA scheduling algorithm for wireless sensor networks with unreliable links," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 1535–1552, 2014.

[60] D. J. Vergados, A. Sgora, D. D. Vergados, D. Vouyioukas, and I. Anagnostopoulos, "Fair TDMA scheduling in wireless multihop networks," *Telecommunication Systems*, vol. 50, no. 3, pp. 181–198, 2012.

[61] L. Sitanayah, C. J. Sreenan, and K. N. Brown, "ER-MAC: A hybrid MAC protocol for emergency response wireless sensor networks," in *Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pp. 244–249, IEEE, 2010.

[62] W. L. Lee, A. Datta, and R. Cardell-Oliver, "FlexiTP: a flexible-schedule-based TDMA protocol for fault-tolerant and energy-efficient wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, no. 6, pp. 851–864, 2008.

[63] S. Ray, I. Demirkol, and W. Heinzelman, "Supporting bursty traffic in wireless sensor networks through a distributed advertisement-based tdma protocol (ATMA)," *Ad Hoc Networks*, vol. 11, no. 3, pp. 959–974, 2013.

[64] H. K. Le, D. Henriksson, and T. Abdelzaher, "A practical multi-channel media access control protocol for wireless sensor networks," in *Proceedings of the 7th international conference on Information processing in sensor networks*, pp. 70–81, IEEE Computer Society, 2008.

[65] Q. Yu, J. Chen, Y. Fan, X. Shen, and Y. Sun, "Multi-channel assignment in wireless sensor networks: A game theoretic approach," in *INFOCOM, 2010 Proceedings IEEE*, pp. 1–9, IEEE, 2010.

[66] O. D. Incel, L. Van Hoesel, P. Jansen, and P. Havinga, "MC-LMAC: A multi-channel MAC protocol for wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 1, pp. 73–94, 2011.

[67] C. Xia, X. Jin, L. Kong, C. Xu, and P. Zeng, "Heterogeneous slot scheduling for real-time industrial wireless sensor networks," *Computer Networks*, vol. 157, pp. 68–77, 2019.

[68] P. Björklund, P. Värbrand, and D. Yuan, "A column generation method for spatial TDMA scheduling in ad hoc networks," *Ad Hoc Networks*, vol. 2, no. 4, pp. 405–418, 2004.

[69] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Transactions on Wireless Communications*, vol. 5, no. 2, pp. 435–445, 2006.

[70] M. Bakshi, B. Jaumard, M. Kaddour, and L. Narayanan, "On TDMA scheduling in wireless sensor networks," in *Electrical and Computer Engineering (CCECE), 2016 IEEE Canadian Conference on*, pp. 1–6, IEEE, 2016.

[71] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 8, no. 10, pp. 1384–1396, 2009.

[72] S. C. Ergen and P. Varaiya, "TDMA scheduling algorithms for wireless sensor networks," *Wireless Networks*, vol. 16, no. 4, pp. 985–997, 2010.

[73] L. Shi and A. O. Fapojuwo, "TDMA scheduling with optimized energy efficiency and minimum delay in clustered wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 7, pp. 927–940, 2010.

[74] S.-C. Kim, "An energy efficient scheduling algorithm for cluster-based mobile wireless networks," in *Ubiquitous and Future Networks (ICUFN), 2012 Fourth International Conference on*, pp. 357–362, IEEE, 2012.

[75] H. Lee, C.-C. Lim, and J. Choi, "Cluster-based multi-channel scheduling algorithms for ad hoc networks," in *Wireless and Optical Communications Networks, 2007. WOCN'07. IFIP International Conference on*, pp. 1–5, IEEE, 2007.

[76] T. Liu, X. Lu, and Z. Dou, "An optimal design of time division multiple access protocol for data link network," in *2016 IEEE International Conference on Electronic Information and Communication Technology (ICEICT)*, pp. 158–161, 2016.

[77] W. Wenzheng, Z. Jinglun, and L. Pengcheng, "Review of tactical data link technology," *Electronics Optics & Control*, vol. 15, no. 11, pp. 41–46, 2008.

[78] K. Bok, J. Lim, S. Hong, and J. Yoo, "A multiple RSU collaborative scheduling scheme for data services in vehicular ad hoc networks," *Cluster Computing*, vol. 20, no. 2, pp. 1167–1178, 2017.

[79] P. Adasme, F. Seguel, I. Soto, and E. S. Juan, "Spatial time division multiple access for visible light communication networks," in *2017 First South American Colloquium on Visible Light Communications (SACVLC)*, pp. 1–6, 2017.

[80] Z. Naghsh and S. Valaee, "Semi-distributed conflict-free multichannel TDMA link scheduling for 5G," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 1407–1411, 2017.

[81] N. Linial, "Locality in distributed graph algorithms," *SIAM Journal on computing*, vol. 21, no. 1, pp. 193–201, 1992.

[82] R. Cole and U. Vishkin, "Deterministic coin tossing with applications to optimal parallel list ranking," *Information and Control*, vol. 70, no. 1, pp. 32–53, 1986.

[83] A. Goldberg, S. Plotkin, and G. Shannon, "Parallel symmetry-breaking in sparse graphs," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pp. 315–324, ACM, 1987.

[84] F. Kuhn and R. Wattenhofer, "On the complexity of distributed graph coloring," in *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pp. 7–15, ACM, 2006.

[85] N. Linial, "Distributive graph algorithms global solutions from local data," in *Foundations of Computer Science, 1987., 28th Annual Symposium on*, pp. 331–335, IEEE, 1987.

[86] M. Hadded, P. Muhlethaler, A. Laouiti, R. Zagrouba, and L. A. Saidane, "TDMA-based MAC protocols for vehicular ad hoc networks: A survey, qualitative analysis, and open research issues," *IEEE Communications Surveys Tutorials*, vol. 17, no. 4, pp. 2461–2492, 2015.

[87] A. Shahraki, A. Taherkordi, Øystein Haugen, and F. Eliassen, "Clustering objectives in wireless sensor networks: A survey and research direction analysis," *Computer Networks*, vol. 180, p. 107376, 2020.

[88] A. B. Tambawal, R. M. Noor, R. Salleh, C. Chembe, M. H. Anisi, O. Michael, and J. Lloret, "Time division multiple access scheduling strategies for emerging vehicular ad hoc network medium access control protocols: a survey," *Telecommunication Systems*, vol. 70, no. 4, pp. 595–616, 2019.

[89] R. Ding and Qing-An Zeng, "A clustering-based multi-channel vehicle-to-vehicle (v2v) communication system," in *2009 First International Conference on Ubiquitous and Future Networks*, pp. 83–88, 2009.

[90] T.-L. Sheu and Y.-H. Lin, "A cluster-based TDMA system for inter-vehicle communications," *Journal of Information Science and Engineering*, vol. 30, pp. 213–231, 01 2014.

[91] F. Borgonovo, A. Capone, M. Cesana, and L. Fratta, "Adhoc mac: New mac architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services," *Wireless Networks*, vol. 10, no. 4, pp. 359–366, 2004.

[92] L. Miao, F. Ren, C. Lin, and A. Luo, "A-ADHOC: An adaptive real-time distributed mac protocol for vehicular ad hoc networks," in *2009 Fourth International Conference on Communications and Networking in China*, pp. 1–6, 2009.

[93] M. S. Almalag, S. Olariu, and M. C. Weigle, "TDMA cluster-based MAC for VANETs (TC-MAC)," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pp. 1–6, 2012.

[94] D. J. Vergados, K. Kralevska, Y. Jiang, and A. Michalas, "Local voting: A new distributed bandwidth reservation algorithm for 6TiSCH networks," *Computer Networks*, vol. 180, p. 107384, 2020.

[95] "Ieee standard for local and metropolitan area networks–part 15.4: Low-rate wireless personal area networks (lr-wpans) amendment 1: Mac sublayer," *IEEE Std 802.15.4e-2012 (Amendment to IEEE Std 802.15.4-2011)*, pp. 1–225, 2012.

[96] H. A. Omar, W. Zhuang, and L. Li, "VeMAC: A TDMA-based MAC protocol for reliable broadcast in VANETs," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1724–1736, 2013.

[97] Wang Ke, Yang Weidong, Li Pan, and Zhu Hongsong, "A decentralized adaptive tdma scheduling strategy for vanet," in *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 216–221, 2013.

[98] M. Hadded, R. Zagrouba, A. Laouiti, P. Muhlethaler, and L. A. Saidane, "An adaptive TDMA slot assignment strategy in vehicular ad hoc networks," *Journal of Machine to Machine Communications*, vol. 1, no. 2, pp. 175–194, 2014.

[99] Z. Tianjiao and Z. Qi, "Game-based TDMA MAC protocol for vehicular network," *Journal of Communications and Networks*, vol. 19, no. 3, pp. 209–217, 2017.

[100] V. V. Mazalov and J. V. Chirkova, "Chapter 1 - nash equilibrium," in *Networking Games* (V. V. Mazalov and J. V. Chirkova, eds.), pp. 1 – 8, Academic Press, 2019.

[101] S. Jian and T. Hou, "Enabling efficient device to device multicast communication by clustering and resource sharing," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, 2019.

[102] S. V. Bana and P. Varaiya, "Space division multiple access (SDMA) for robust ad hoc vehicle communication networks," in *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, pp. 962–967, IEEE, Aug 2001.

[103] Z. Doukha and S. Moussaoui, "An SDMA-based mechanism for accurate and efficient neighborhood-discovery link-layer service," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 603–613, 2016.

[104] M. Conforti, G. Cornuéjols, and G. Zambelli, *Integer programming*, vol. 271. Springer, 2014.

[105] "MATLAB optimization toolbox." `https://www.mathworks.com/products/optimization.html`. Accessed: 2019-07-20.

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: Eksert, Mustafa Levent
Nationality: Turkish
Date and Place of Birth: 06.03.1987, Trabzon
E-mail: mleventeksert@gmail.com

EDUCATION

Degree	Institution	Year
P.H.D	Middle East Technical University - Computer Engineering	2020
M.Sc.	Middle East Technical University - Computer Engineering	2013
B.S.	Middle East Technical University - Computer Engineering	2010

RESEARCH INTERESTS

Computer Networks, Wireless Networks, Ad Hoc Networks, Clustering, Medium Access Control, TDMA, Link Scheduling, Optimization.

PROFESSIONAL EXPERIENCE

2020 - Present:

Haberleşme ve Bilgi Teknolojileri, ASELSAN, Ankara, Turkey
Position: Senior Expert Engineer

2010 - 2019:

Department of Computer Engineering, METU, Ankara, Turkey
Position: Teaching / Research Assistant
Assisted Courses:

- CENG111 - Introduction to Computer Engineering Concepts
- CENG140 - C Programming
- CENG213 - Data Structures
- CENG232 - Logic Design
- CENG352 - Database Management Systems
- CENG435 - Data Communications and Computer Networking
- CENG477 - Introduction to Computer Graphics
- CENG490 - Computer Engineering Design
- CENG530 - Computer Networks and Communications

2016:

Department of Computer Engineering, METU, Ankara, Turkey
Position: Internet-based Distance Education - Asynchronous (IDEA) Teaching Assistant
Assisted Courses:

- JVYA - Data Stuctures and Algorithms with JAVA Programming

Membership

(Researcher) - Wireless Systems, Networks and CyberSecurity Laboratory (WINS Lab), Middle East Technical University (METU), Computer Engineering, B-Z19.

https://wins.ceng.metu.edu.tr/

Research Projects

(Researcher) - ASELSAN HBT-TE-2017-012, Geniş Bant Dalga Şekli İçin Katmanlar Arası Eniyileme Temelli Öbekleme Projesi, 2017-2018.

PUBLICATIONS

International SCI Journals

- M. Levent Eksert, Hamdullah Yücel, Ertan Onur, "Intra- and Inter-Cluster Link Scheduling in CUPS-Based Ad Hoc Networks", Computer Networks, Elsevier, Oct 2020, doi:10.1016/j.comnet.2020.107659.

- Doğanalp Ergenç, Levent Eksert, Ertan Onur, "Dependability-Based Clustering in Mobile Ad-hoc Networks", Ad Hoc Networks, Vol. 93, Elsevier, Oct 2019, doi:10.1016/j.adhoc.2019.101926.

- Ahmet Oğuz Akyüz, M. Levent Eksert, M. Selin Aydin, An Evaluation of Image Reproduction Algorithms for High Contrast Scenes on Large and Small Screen Display Devices, Computers & Graphics, Volume 37, Issue 7, 2013, Pages 885-895, ISSN 0097-8493, doi:10.1016/j.cag.2013.07.004.

International Conferences

- Doğanalp Ergenç, Levent Eksert and Ertan Onur, "Density-Aware Probabilistic Clustering in Ad Hoc Networks," 2018 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom), Batumi, 2018, pp. 1-5, doi: 10.1109/BlackSeaCom.2018.8433605.

National Conferences

- Doğanalp Ergenç, Levent Eksert and Ertan Onur. ”Ağırlık Tabanlı Öbekleme Algoritmasının Başarım Eniyilemesi ve Analizi”, Türkiye Bilişim Derneği 35. Ulusal Bilişim Kurultayı, Bilişim 2018, 21 Kasım, Ankara.
- Levent Eksert and Ertan Onur, "Tasarsız Ağlarda Geçmiş-Tabanlı Bağlantı Çizelgeleme", ASELSAN 4. Haberleşme Teknolojileri Çalıştayı 2019, 18 Şubat, Ankara.