

# OLSR-AWARE CROSS-LAYER CHANNEL ACCESS SCHEDULING IN WIRELESS MESH NETWORKS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Miray Kaş

June 30, 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Assist. Prof. Dr. İbrahim K rpeođlu (Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Assoc. Prof. Dr. Ezhan Karařan

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

---

Prof. Dr.  zg r Ulusoy

Approved for the Institute of Engineering and Science:

---

Prof. Dr. Mehmet B. Baray  
Director of the Institute

## ABSTRACT

# OLSR-AWARE CROSS-LAYER CHANNEL ACCESS SCHEDULING IN WIRELESS MESH NETWORKS

Miray Kaş

M.S.in Computer Engineering

Supervisor: Assist. Prof. Dr. İbrahim Körpeoğlu

June 30, 2009

A wireless mesh network (WMN) is a communications network in which the nodes are organized to form a mesh topology. WMNs are expected to resolve the limitations and significantly improve the performance of wireless ad-hoc, local area, personal area, and metropolitan area networks, which is the reason that they are experiencing fast-breaking progress and deployments.

WMNs typically employ spatial TDMA (STDMA) based channel access schemes which are suitable for the high traffic demands of WMNs. Current research trends focus on using loosening the strict layered network implementation in order to look for possible ways of performance improvements. In this thesis, we propose two STDMA-based cross-layer OLSR-Aware channel access scheduling schemes (one distributed, one centralized) that aim better utilizing the network capacity and increasing the overall application throughput by using OLSR-specific routing layer information in link layer scheduling. The proposed centralized algorithm provides a modification of the traditional vertex coloring algorithm while the distributed algorithm is a fully distributed pseudo-random algorithm in which each node makes decisions using local information. Proposed schemes are compared against one another and against their Non-OLSR-Aware versions via extensive ns-2 simulations. Our simulation results indicate that MAC layer can obtain OLSR-specific information with no extra control overhead and utilizing OLSR-specific information significantly improves the overall network performance both in distributed and centralized schemes. We further show that link layer algorithms that target the maximization of concurrent slot allocations do not necessarily increase the application throughput.

*Keywords:* Centralized Scheduling, Distributed Scheduling, Cross-Layer Design, OLSR, STDMA, MAC.

## ÖZET

# ÖRGÜSEL AĞLARDA OLSR-DUYARLI KATMANLAR ARASI KANAL ERİŞİM PLANLAMASI

Miray Kaş

Bilgisayar Mühendisliği, Yüksek Lisans

Tez Yöneticisi: Yrd. Doç. Dr. İbrahim Körpeoğlu

Haziran 30, 2009

Kablosuz örgüsel ağlar, radyoların örgüsel topoloji oluşturdukları bir çeşit haberleşme ağıdır. Kablosuz örgüsel ağların amaca özel tasarısız ağlar, kablosuz yerel ağlar, kablosuz kişisel ağlar ve kablosuz kentsel ağlar gibi bir çok ağ çeşidinde gözlenen kısıtlayıcı etkenlere çözümler sunması ve bu ağların performanslarını önemli ölçüde arttırması beklenmektedir. Bu sebeplerden ötürü, son dönemlerde kablosuz örgüsel ağlar alanında hızlı ilerlemeler kaydedilmiş ve bir çok yeni konuşlandırmalar yer almıştır.

Kablosuz örgüsel ağlar için tasarlanmış bir kanal erişimi zamanlama algoritması basit, ölçeklenebilir ve işletim yükü az olan bir algoritma olmalıdır. Son dönemlerdeki araştırmalarda katmanlı ağ gerçekleştirme yapısındaki katmanların harmanlanmasıyla elde edilebilecek olası performans arttırmaları da incelenmektedir. Kablosuz örgüsel ağlarda yüksek trafik taleplerini karşılamaya uygun STDMA-tabanlı kanal erişim yöntemleri sıkça kullanılmaktadır. Bu tezde, ağda üretilen toplam iş miktarını arttırmayı amaçlayan OLSR'a özgü bilgilerden faydalanarak işleyen, biri dağıtık, biri merkezi kontrollü olmak üzere iki farklı STDMA-tabanlı kanal erişimi zamanlama algoritması sunulmaktadır. Merkezi kontrollü algoritma klasik grafik renklendirme algoritmasının bir uyarlaması olup dağıtık algoritma her radyonun tahmin edilebilir bir rastgelelik ile karar aldığı bir algoritmadır. Önerilen algoritmalar birbirleriyle ve OLSR'a özgü bilgileri kullanmayan uyarlamaları ile ns-2 simülasyon ortamında kıyaslanmışlardır. Simülasyon sonuçlarımız OLSR bilgisinin MAC katmanınca ekstra bir yük olmaksızın elde edilebileceğini ve OLSR'a özgü bilgilerin kullanılması halinde elde edileceği öngörülen performans artışını doğrulamaktadır.

*Anahtar sözcükler:* OLSR, Merkezi Zamanlama, Dağıtık Zamanlama, Bölünmüş Zamanlı Ortam Erişimi, Ortam Erişimi Kontrolü, Katmanlar Arası Tasarım.

## Acknowledgement

I would like to thank Assist. Prof. Dr. İbrahim Körpeođlu and Assoc. Prof. Dr. Ezhan Karařan for their continuous support and invaluable guidance throughout my research.

I would also like to thank Prof. Dr. Özgür Ulusoy for evaluating my thesis.

I am also grateful to Ateř Akaydın for his valuable discussions throughout my research.

Finally, I would like to thank TÜBİTAK (The Scientific and Technological Research Counsel of Turkey) for providing me financial support for my M.S. studies.

# Contents

- 1 Introduction** . . . . . **1**
  - 1.1 Motivation and Scope . . . . . 1
  - 1.2 Contributions . . . . . 3
  - 1.3 Thesis Organization . . . . . 4
  
- 2 Background and Related Work** . . . . . **5**
  - 2.1 STDMA Scheduling: Definition, Classification and Examples . . . . . 5
    - 2.1.1 Definition and Classification of STDMA Scheduling . . . . . 6
    - 2.1.2 TDMA Based MAC Protocols . . . . . 9
  - 2.2 Cross-Layer Studies . . . . . 11
  - 2.3 OLSR . . . . . 13
    - 2.3.1 Neighborhood Discovery . . . . . 14
    - 2.3.2 Topology Dissemination . . . . . 15
    - 2.3.3 Multiple Interface Support . . . . . 16
    - 2.3.4 MPR Selection . . . . . 16

2.3.5	State-of-Art Research on OLSR . . . . .	17
<b>3</b>	<b>OLSR-Aware Channel Access Scheduling</b>	<b>20</b>
3.1	Network Model . . . . .	20
3.2	MPR Based Weighting Scheme . . . . .	22
3.2.1	Dissemination of Weight Information . . . . .	23
<b>4</b>	<b>Centralized Channel Access Scheduling</b>	<b>26</b>
4.1	Heuristic-based Algorithms for Vertex Coloring . . . . .	26
4.2	OLSR-Aware Centralized Scheduling Algorithm (OA-C) . . . . .	28
4.3	Non-OLSR-Aware Centralized Scheduling Algorithm (NOA-C) . . . . .	30
<b>5</b>	<b>Distributed Channel Access Scheduling</b>	<b>32</b>
5.1	Basic Features . . . . .	32
5.2	OLSR-Aware Distributed Scheduling Algorithm (OA-D) . . . . .	33
5.3	Non-OLSR-Aware Distributed Scheduling Algorithm (NOA-D) . . . . .	35
5.4	Additional Discussion . . . . .	36
<b>6</b>	<b>Simulation Implementation and Results</b>	<b>37</b>
6.1	Simulation Implementation . . . . .	37
6.2	Simulation Results and Analysis . . . . .	39
6.2.1	Concurrency Levels Achieved . . . . .	41
6.2.2	Simulation Results with Uniform Traffic Scenarios . . . . .	44

6.2.3	Simulation Results with Nonuniform Traffic Scenarios . . .	47
6.2.4	The Effects of Network Size . . . . .	49
6.2.5	The Effects of the Maximum Queue Size . . . . .	50
6.2.6	Case Study: Simulation Results Obtained on a Sample 15- Node Network . . . . .	53
<b>7</b>	<b>Conclusion</b>	<b>61</b>



# List of Figures

2.1	Classification of STDMA scheduling algorithms. . . . .	7
2.2	Snapshot of OLSR tables in node $n0$ . . . . .	16
3.1	RFC 3626 specification on OLSR HELLO Message structure. . . . .	23
3.2	Proposed OLSR HELLO Message structure. . . . .	24
6.1	Sample topology for 20-node network. . . . .	42
6.2	Results on achieved/utilized concurrency levels. . . . .	43
6.3	Averaged uniform traffic simulation results on 20-node networks. . . . .	45
6.4	Averaged nonuniform traffic simulation results on 20-node networks. . . . .	48
6.5	The effects of network size. . . . .	49
6.6	The effects of the maximum queue size on the performance of OA-C. . . . .	51
6.7	The effects of the maximum queue size on the performance of NOA-C. . . . .	51
6.8	The effects of the maximum queue size on the performance of OA-D. . . . .	52
6.9	The effects of the maximum queue size on the performance of NOA-D. . . . .	52
6.10	Sample 15-node network topology and MPR Selectors. . . . .	54

6.11	Histograms showing the number of allocated time slots/sent packets on the sample 15-node network topology, Topology (1). . . . .	55
6.12	Total number of allocated slots/ transmitted packets and successfully delivered packets. . . . .	57
6.13	Uniform traffic simulation results on the sample 15-node network (Topology-1). . . . .	59
6.14	Nonuniform traffic simulation results on the sample 15-node network (Topology-1). . . . .	59

# List of Tables

6.1	Ns-2 simulation parameters. . . . .	40
-----	-------------------------------------	----

# Chapter 1

## Introduction

### 1.1 Motivation and Scope

A wireless mesh network (WMN) is a multi-hop communication network in which the nodes are organized to form a mesh topology, providing communication over multiple wireless links. In the last decade, wireless mesh networking technology has emerged as a key enabling technology to provide better services in wireless networks. Despite the fact that there is a lot of research done/going on about mesh networks, the available MAC and routing protocols applied to WMNs do not have enough performance and scalability; requiring the redesign of the existing wireless networking protocols [4].

As in all types of wireless networks, bandwidth is a very scarce resource in multi-hop wireless mesh networks. Improving the performance of multi-hop wireless mesh networks, especially in terms of the overall network throughput, is a very active research area. Since CSMA based schemes are known to result in inferior performance in multi-hop networks with high traffic demands [53], in WMNs, usually mechanisms based on FDMA, CDMA or TDMA are employed for multiple access coordination. Considering that TDMA is the preferred MAC mechanism in emerging OFDM based standards (e.g. WiMAX 802.16d mesh mode [1]), we focus on MAC schemes based on Spatial TDMA (STDMA).

On the other hand, cross-layer networking is an increasingly important paradigm in the direction the wireless networks evolve and it is currently one of the most active research areas in wireless networking. In cross-layer networking, the strict layered network implementation is relaxed and the knowledge is shared among loosely-coupled layers through stricter cooperation in order to provide efficient allocation of network resources.

In this thesis, we unite the concept of cross-layer networking and the methods for improving the performance of multi-hop wireless mesh networks. We propose two different STDMA based weighted channel access scheduling schemes whose primary objective is to improve the overall application throughput by means of cross-layer interaction.

In [46], it is pointed out that with the evolving wireless communication technology, the paradigm shift towards cross-layer designs has already started to emerge. However, within the research community, there is still debate ongoing as to whether cross-layering is worthwhile or not. Therefore, as a second objective, through quantitative and qualitative measurements, we also aim contributing to the discussion of whether or in what cases the type of cross-layer information we exploit becomes useful.

Our cross-layer design is based on the interaction between MAC layer and the network layer. We use Optimized Link State Routing protocol (OLSR) as the routing protocol, which collects the topology information at every node. We propose utilizing the information collected by OLSR at every node in providing better channel access schedules and medium access coordination. In our approach, in order to increase the scalability of the MAC layer, we target low-overhead scheduling schemes which would exploit the information readily available at every node as much as possible while keeping the extra overhead at minimum level.

One of our proposed channel access scheduling schemes is a centralized scheme while the other one is distributed. Although they are quite different, their most important point is common; they both employ our proposed cross-layer weighting scheme, so called MPR Based Weighting Scheme, within their slot allocation process which improves the overall application throughput of the network as

presented in the simulation results. Our proposed scheduling algorithms are implemented as a part of MAC layer and they do not require the exchange of MAC level control messages to acquire the information they need such as the weight and the topology information.

Our work distinguishes itself from the other studies in the literature through its simplicity in using OLSR-specific MPR information within the slot allocation procedure and through showing that link layer scheduling algorithms that target maximization of concurrent time slot allocations do not necessarily increase the application throughput in multihop wireless mesh networks.

## 1.2 Contributions

In this thesis, we propose two different weighted channel access scheduling schemes (one distributed, one centralized) for scheduling the nodes in a wireless mesh network that with the aim of increasing the overall application level throughput.

The basic features of our proposals are as the following:

- We propose a cross-layer weighting scheme called MPR Based Weighting Scheme which makes use of OLSR-specific routing layer information. It is used as a part of the proposed channel access scheduling schemes.
- We propose a centralized channel access scheduling algorithm (OLSR-Aware Centralized Scheduling) which uses the above mentioned weighting scheme.
- We propose a distributed pseudo-random channel access scheduling algorithm (OLSR-Aware Distributed Scheduling) which uses the above mentioned weighting scheme.

### 1.3 Thesis Organization

The rest of this thesis is structured as follows. In the first half of Chapter 2, general definition of STDMA scheduling is given and the related work in the literature is briefly reviewed. The rest of Chapter 2 is devoted to explore the basic features of OLSR routing protocol. In Chapter 3, the major properties of the targeted network model are listed and MPR Based Weighting Scheme is presented in detail. Chapter 4 starts with a brief discussion on the graph-theoretical roots of OLSR-Aware Centralized Scheduling followed by the presentations of OLSR-Aware Centralized Scheduling Algorithm and its Non-OLSR-Aware version. Chapter 5 presents OLSR-Aware Distributed Scheduling Algorithm along with its Non-OLSR-Aware version. In Chapter 6, first, the details of the simulation implementation are presented, and then simulation results are reported and an in-depth analysis of the presented results is provided. The last section of Chapter 6 examines a sample case in detail in order to provide a better understanding of the discussed concepts and results. Chapter 7 finally concludes the thesis covering the key results obtained and discussing possible research directions.

# Chapter 2

## Background and Related Work

Along with the multiple access scheme, the routing protocol is one of the most important elements of a WMN required to keep the network functional. In the first part of this chapter, brief information on the general classification of Spatial TDMA (STDMA) based channel access scheduling is given and some examples in the literature are discussed. We, then, briefly discuss some of the cross-layer studies available in the literature. The final section of this chapter (Section 2.3) elaborates on the main features of Optimized Link State Routing (OLSR) protocol.

### **2.1 STDMA Scheduling: Definition, Classification and Examples**

In the current literature, most of the channel access protocols are either contention-based (CSMA, ALOHA) or schedule based (TDMA, FDMA, CDMA) [15]. Since the focus of this thesis is on access scheduling in STDMA based multi-hop networks, only a brief survey of STDMA related algorithms is presented.



### 2.1.1 Definition and Classification of STDMA Scheduling

In its most broad state, scheduling is the allocation of scarce resources to activities with the objective of optimizing one or more performance measures [25]. When the scheduling of nodes' accesses to the communication channel in multi-hop WMNs by the way of STDMA is of concern, the scarce resources are the time slots and the activities are either the nodes or the links in the network that are willing to transmit packets.

In TDMA scheduling problem, time is split into equal intervals called time slots [50]. Each time slot is designed to be long enough to allow a single packet of maximum size to be transmitted. In order to ensure collision-free transmissions, the duration of the time slots is usually calculated as in Eq (2.1) by adding a *GuardTime*, hence a limitation on TDMA-channel's potential bandwidth is introduced.

$$Duration (Slot) = (8 * (MaxPacketSize) / Bandwidth) + GuardTime \quad (2.1)$$

Each node should be aware of the time slots' starting and ending times. Therefore, TDMA based protocols require synchronization which might be quite complex to achieve in some cases. Since TDMA based protocols are able to provide collision-free transmissions while achieving higher throughput and fairness, TDMA scheduling problem attracts the attention of research community despite the difficulties of synchronizing the nodes and arranging the efficient use of time slots.

Apart from these, if scheduling in a TDMA-enabled multi-hop environment is compared against scheduling in a TDMA-enabled single-hop environment, performing scheduling in a multi-hop environment turns out to be more challenging than performing scheduling in a single-hop network since the spatial reuse of time slots is possible [17]. In multi-hop networks, multiple nodes can transmit during the same time slot as long as they are on the non-conflicting parts of the network. Hence, in our case, a schedule dictates the concurrent allocation of one or more

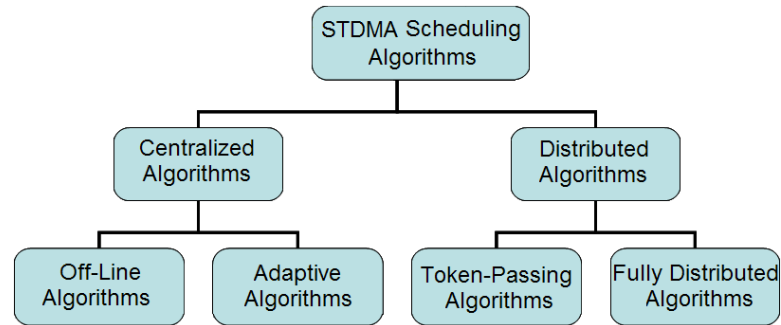


Figure 2.1: Classification of STDMA scheduling algorithms.

time slots to one or more nodes so that the spatial reusability is exploited and the number of packets that can be delivered in a collision-free manner is maximized.

To be able to achieve a collision-free schedule, networks consisting of nodes with single half-duplex radios should avoid the two main types of conflicts:

1. **Primary Conflict:** Observed if a node is scheduled to transmit and receive at the same time.
2. **Secondary Conflict:** Observed if a node is scheduled to receive from two different nodes simultaneously.

In STDMA-operated multi-hop networks, in order to ensure that both kinds of conflicts are avoided, no two nodes within the same 2-hop neighborhood should be scheduled to transmit at the same time slot [26, 50].

Figure 2.1 presents a broad classification of multi-hop STDMA scheduling algorithms, identifying centralized and distributed scheduling as the two main types. Centralized scheduling algorithms commonly require global knowledge about the network topology and are run at a central site. In off-line centralized scheduling algorithms, the scheduling problem is solved once and for all, whereas in adaptive centralized scheduling algorithms, the central scheduler site solves the scheduling problem dynamically to be able to adapt to the topology changes.

The proposed centralized algorithms usually construct a tree of nodes which has the central scheduler as its root or employ graph theoretic solutions such as link/vertex coloring. The tree-based centralized scheduling algorithms might prefer inferring the routing tree from the MAC level control messages [12] or constructing it through interference and/or some other metric based cost assignments [51] or using traditional minimum spanning tree algorithms [14].

On the other hand, in order to decrease vulnerability to topological changes and to improve flexibility, distributed algorithms are considered essential. Distributed algorithms are usually implemented as either token passing algorithms or fully distributed algorithms. In token passing algorithms, a token is passed around the nodes in the network and a node holds the token while computing its portion of the algorithm [26]. In fully distributed algorithms, each node can simultaneously run the algorithm, essentially resulting in parallel computation. In the literature there are many proposed distributed channel access schemes which can be further classified as cluster based, hybrid [42], randomized [7, 15, 43] or graph-theoretic [29, 54].

Apart from these, recently, cross-layer paradigm is acknowledged more frequently as a potential solution to overcome the performance problems in WMNs and it is a very important aspect of our work. In the literature, there are various studies each dealing with different aspects of cross-layering. For instance, in [20], the authors discuss different cross-layer methodologies aiming to improve the performance of OLSR using the information available at MAC layer. Similarly, [55] studies cross-layer design on wireless mesh networks, defining a cross-layer routing metric and discussing its implementation in routing protocols such as DSDV. Other works, such as [6, 10, 37], which use dynamically changing cross-layer information acquired from either the physical layer or the routing layer or both in order to improve the performance of MAC layer also exist.

However, to the best of our knowledge, there is no research done on STDMA based distributed or centralized channel access scheduling which uses OLSR-specific routing layer information in a cross-layer manner. In the following sections, we propose one adaptive centralized and one fully distributed scheduling

algorithm, each of which avoid both types of conflicts (primary and secondary) while achieving significant performance improvements. The most remarkable advantage of our proposed schemes is that they improve the overall performance of the network using the readily available information at *no* extra communication overhead and without incurring the additional cost of dynamical adjustments as done in most of the cross-layer studies in the literature. Either distributed or centralized, there are also many TDMA-based scheduling studies in the literature which only focus on maximizing the number of concurrent slot allocations to be able to maximize the overall throughput [49, 35]. In our work, through our simulation results, we show that maximizing the number of concurrent slot allocations does not necessarily maximize the overall application throughput. Before moving on to the proposed algorithms, we discuss some of the related works in the literature in somewhat more detail and provide some background information about OLSR.

### 2.1.2 TDMA Based MAC Protocols

The scheduling algorithms, the algorithms developed for arranging the efficient use of time slots, are usually implemented as a part of MAC layer. In the literature, there is considerable amount of research on TDMA-based channel access schemes. In this section, we discuss some of them in more detail.

The centralized slot allocation/broadcast scheduling algorithms available in the literature follow many different methods. For instance, in [31], a centralized scheduling algorithm which uses a modified genetic algorithm, called the genetic-fix algorithm is presented. In this work, for a network of  $n$  nodes, the number of slots in a frame is assumed to be fixed (say  $m$ ), and a  $m \times n$  transmission matrix is generated by the proposed genetic-fix algorithm which explores a fixed subset of solution space, as the name implies.

In [16], first, the problem of optimal scheduling of broadcasts in a TDMA based multi-hop radio network is shown to be NP complete. The authors form an augmented graph and show that the scheduling problem is equivalent to finding

a maximal independent set. They next describe a heuristic based algorithm which again fills a transmission matrix which is  $n \times n$  (i.e. the number slots in a frame and the number of nodes in the network are both  $n$ ).

Along the same lines, [17] proposes two heuristics in order to solve the scheduling problem in TDMA based networks in a centralized manner. One of these heuristics is based on the direct scheduling of the nodes through coloring while the other works by scheduling the levels in the routing tree before scheduling the nodes. The performances of these two heuristic based centralized algorithms are then compared against one another and against a token based distributed algorithm.

We next discuss some examples of TDMA-based distributed channel access schemes in more detail. As mentioned, some of these schemes are cluster based, some are hybrid like Z-MAC (ZEBRA-MAC [42]), some of them are randomized (DRAND [43], NAMA [7]) and there are some others like HBS [15], TRAMA[38] that perform node access scheduling taking design specific metrics into account.

Z-MAC [42] is designed as a hybrid of CSMA and TDMA. It starts as CSMA, and switches to TDMA when the load exceeds a certain threshold. Z-MAC requires DRAND at startup to establish the first schedule. Each node runs a distributed slot selection algorithm and the owner of the slot uses a smaller random back-off value.

DRAND [43], the algorithm which is also used in Z-MAC, is the distributed version of the heuristic based centralized RAND algorithm [39]. It is a complex algorithm requiring the implementation of a four-state finite state machine with states IDLE, GRANT, SEND and RECEIVE and special control messages.

In [7], NCR (Neighbor-Aware Contention Resolution) algorithm is presented and four channel access protocols based on NCR are given. In NCR, it is assumed that knowledge about 2-hop neighborhood (that is, the contenders) is achieved by some means and the nodes have mutual knowledge. A seed value for random value generation is formed as the combination (node-id, contended-slot-number) for each of the 2-hop neighbors and a winner, the node that draws the highest random

value for the contended slot, is elected as the winner of the slot. The priority value of each node is calculated by Eq (2.2) below where  $t$  is the contended-slot-number and  $k$  is the node-id:

$$p_k^t = \text{Rand}(k \oplus t) \oplus k \quad (2.2)$$

In NAMA [7], every node runs NCR and no two nodes within the same 2-hop neighborhood transmit simultaneously.

HBS (History Based Scheduling) [15] is another protocol using NCR algorithm, improving on NAMA by using the ratio of the number of slots in a frame used by a node to transmit to the number of slots in a frame given to that node as a success indicator. If this ratio drops below some pre-set threshold, the weight of the node is decremented. Similarly, the weight of a node is incremented if the ratio above is larger than another pre-set threshold.

TRAMA (Traffic Adaptive Medium Access) [38] is another TDMA protocol which assigns time slots to the nodes through the use of 1-hop traffic information and 2-hop neighborhood information. For each transmission time, each node selects one of the transmitting, receiving, stand-by modes. The nodes with no data to send are not involved in the election.

## 2.2 Cross-Layer Studies

Recently, cross-layer paradigm is acknowledged more frequently as a potential solution to overcome the performance problems in WMNs and it is an important dimension of our work. In the literature, there are various studies each dealing with different aspects of cross-layering.

To exemplify a few, in [24, 10], cross-layering network and MAC layers is considered, whereas in [37] an intelligent MAC layer which acts coherently with the routing layer as well as the physical layer is preferred.

The authors of [24] suggest using distributed scheduling for intranet and centralized scheduling for the Internet traffic. The main idea is that the split between the distributed and centralized scheduling in a time frame may not actually represent the ratio between the intranet and Internet traffic. The links that are in the centralized scheduling tree are called centralized links and the rest of the links between any two mesh nodes are called distributed links. The authors suggest checking the queue lengths of the associated centralized links and change the route for the Internet traffic to the distributed links if the queue length of a centralized link exceeds a certain threshold. Their aim is to reduce the end-to-end delay and number of packet drops in the Internet traffic. However, not to hinder the actual intranet traffic for a long time, they switch back to the normal routes (the routes through the centralized links) if the congestion (queue length) drops below a certain threshold.

In [10], cross-layer concept is introduced and a centralized scheduling algorithm based on multi-path routing is proposed. The implementation of the cross-layer module is separated into two interdependent sub-modules:

1. Multi-path Routing Module
2. Centralized Scheduling Module

The Multi-path Routing Module in the Network Layer is responsible for multi-path source routing (searching for different routes and selecting the optimized routes), interference avoidance, load balancing and QoS guarantee. The optimized route is selected via calculating a metric as a combination of least interference, load balance and QoS indicators. The routing module passes the routing tree and interference table down to the MAC layer.

MAC layer contains the Centralized Scheduling Module which uses the information obtained from the Multi-path Routing Module and is responsible for the resource allocation, spatial reuse and request collection. MAC layer is also responsible for informing the Multi-path Routing Module about nodes' resource requests. Associating these requests with the possible available routes, the centralized scheduler can then assign minislots to mesh nodes using the information

obtained from interference table and the routing tree.

In [37], a novel cross-layer structure is presented which achieves increase in the overall network throughput as well as decrease in the power consumption by overcoming mutual interferences. The interfering link pairs are specified as input to the proposed cross-layer architecture. The authors put emphasis on three distinct parts as the components of this architecture: Power Control Process, Tree-type Routing Construction, and Tree-level based Scheduling. The scheduling algorithm combines the rate adaptation and the power control algorithms. In the scheduling algorithm, the requests are relayed while being concatenated at each node with the children nodes' requests. Each node assigns bandwidth to its links in proportion to its links' queue loads.

## 2.3 OLSR

OLSR is one of the most widely used MANET routing protocols in wireless ad-hoc and mesh networks which collects various information in its tables. OLSR was developed at INRIA and standardized by IETF in RFC 3626 in 2003 [13]. It is mainly aimed for mobile wireless networks [22]. OLSR provides an optimization of the classical link state routing protocol and it uses the traditional shortest path algorithm. In a link state routing protocol, the overhead introduced by the transmission of broadcast packets is quite high and OLSR addresses this problem by using multi point relay (MPR) nodes.

A node is called an MPR node if it is chosen by one or more of its 1-hop neighbors to forward their messages and the collection of MPR nodes form a connected backbone. Since only a subset of neighbors receiving a broadcast message has to relay a message (i.e. only MPR nodes), the message overhead is reduced in comparison to pure flooding mechanism.

In OLSR, there are three kinds of control messages: HELLO messages, Topology Control (TC) messages and Multiple Interface Declaration (MID) messages.



These messages are used to perform main functionalities of OLSR such as neighborhood discovery, topology dissemination and multiple interface support which are explained in detail in the subsequent subsections.

### 2.3.1 Neighborhood Discovery

Every node detects its neighborhood and periodically broadcasts HELLO messages that contain the list of its known neighbors along with their associated link status. According to the specifications in RFC 3626, the link status can be one of the following:

1. SYM\_LINK: If communication is possible in both directions (Symmetric).
2. ASYM\_LINK: If communication is possible in only one direction (Asymmetric).
3. LOST\_LINK: If the timers for considering the link as symmetric or asymmetric have both expired, the link is considered as lost.
4. UNSPEC\_LINK: If no specific information about the link is given.

Similarly, RFC 3626 specifies the following as the possible neighbor types:

1. SYM\_NEIGH: Indicates that the node has at least one symmetrical link with this neighbor node. Communication is possible in both directions.
2. MPR\_NEIGH: Indicates that communication is bidirectional and the HELLO message's originating node has selected the associated neighbor node as its MPR.
3. NOT\_NEIGH: Indicates that the nodes are either no longer or have not yet become symmetric neighbors.

A HELLO message is sent once in every HELLO\_INTERVAL and the periodical use of HELLO messages enable link sensing, discovery of 2-hop neighborhood and MPR signaling. Information about links, 1-hop neighbors and 2-hop neighbors are kept by each node in its Link Set, Neighbor Set, and 2-hop Neighbor Set, respectively.

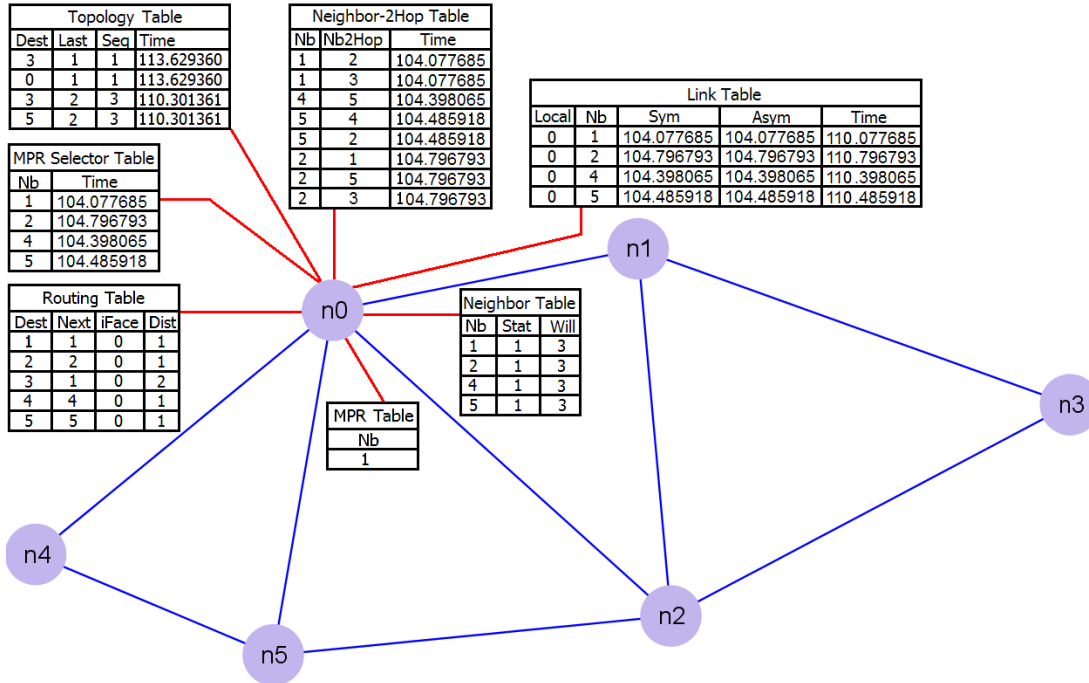
### 2.3.2 Topology Dissemination

The basic idea of link state routing is that the protocol is performed by every node and every node in the network holds a connectivity map of the network, as a graph showing which nodes are connected to which nodes. This is the reason why link state routing protocols are also called distributed-database protocols [30].

In OLSR, the link state information is *only* generated by MPR nodes and flooded to each node in the network through TC messages. Every MPR node generates a TC message at least once in every TC\_INTERVAL seconds and a TC message originated by an MPR node  $X$  contains the list of  $X$ 's MPR Selectors stating which nodes are directly accessible through the MPR node  $X$ .

Each node keeps the information collected via TC messages in its Topology Table. Therefore, each node has an overview of the network, and this overall network information is used by each node to calculate the route to each known destination. The routes are computed using Dijkstra's shortest path algorithm having the hop-count as the metric and the computed routes are kept in the Routing Table.

In Figure 2.2, a snapshot of the information node  $n0$  keeps in its OLSR tables on 100<sup>th</sup> second of a ns-2 simulation is presented on a sample network configuration.

Figure 2.2: Snapshot of OLSR tables in node  $n0$ .

### 2.3.3 Multiple Interface Support

The third message type (MID) is used to declare the presence of multiple interfaces on a node. Each node with multiple interfaces periodically sends MID messages to announce its interface configuration to the other nodes. In other words, MID messages perform the task of declaring the relation between OLSR interface addresses and the main address of a node. In this thesis, we focus on networks consisting of nodes with single radio hence, in our work, we do not make use of MID messages.

### 2.3.4 MPR Selection

The performance of an OLSR enabled network depends highly on MPR nodes. Therefore, MPR Selection process should be well comprehended. In [48], the authors prove through polynomial time reduction to the Dominating Set Problem that the problem of finding an optimal MPR Set (i.e. finding an MPR set with

minimal size) is NP-Complete. An efficient heuristic proposed for performing MPR Selection using a greedy approach is specified in RFC 3626 [13]. The intuition behind this heuristic function is that a node selects its MPRs such that at the end of the selection process, it has no 2-hop neighbors that it cannot reach. However, it should also be noted that the resulting MPR set is not necessarily minimum.

The main steps of MPR Selection procedure are as follows:

1. If there are any 2-hop neighbors that are reachable through only one of the 1-hop neighbors, add that 1-hop neighbor to the MPR Set.
2. While there are any remaining 2-hop neighbors, not covered by the nodes that are already added to the MPR Set during the first step, select the neighbor node  $X$  which covers the largest number of uncovered 2-hop neighbors, and add  $X$  to the MPR Set.

Step-2 is repeated until MPR Set covers all 2-hop neighbors. Each node keeps information about the nodes it has selected as its MPRs and the nodes that have selected it as MPR in its MPR Set and MPR Selector Set, respectively.

Using the algorithm given above, the complexity of the MPR Selection Process depends not only on the size of 1-hop neighborhood but also on the size of 2-hop neighborhood. Assume that the degree of nodes in the network are bounded by  $n$ . In the worst case, all nodes in the 1-hop network will be of largest degree, resulting in a 2-hop neighborhood of size  $n^2$ . In such a case, it is intuitive to see that the MPR Selection Process usually is completed in less than  $O(n^2)$  steps, resulting in a complexity upper bounded by  $O(n^2)$ .

### 2.3.5 State-of-Art Research on OLSR

In the current literature OLSR is examined in many papers from many different aspects. In this section, some of these studies are discussed in more detail.

For instance, there are several studies that compare different routing protocols' effects on the performance of the wireless networks [11, 8]. The results presented in these papers indicate that OLSR provides better performance in terms of data packet delivery ratio, throughput, packet latency and routing overhead when compared against other MANET routing protocols such as AODV, DSDV, and DSR. Such results make OLSR a popular routing protocol which has been studied in many ways.

[32, 33] investigate the effects of interference on OLSR protocol. In [32], the authors show that interference can degrade the performance of the already established flows and the authors propose a bandwidth reservation model in order to ensure that the interference a new flow causes does not affect the performance of the already accepted flows.

In [23], a modified version of OLSR which uses the link cost values in the establishment of routes is used. Link cost value involves maximum signal strength (RSSI), link capacity and contention information. The information required for link cost calculation is provided by their Resource Aware Routing for Mesh (RARE) module which employs passive monitoring to collect radio link information.

[2] and [34] discuss another modification to OLSR protocol which aims to introduce scalability into OLSR through the use of fish-eye routing techniques. Fish-eye technique is actually a technique used in graphics in order to reduce the size of graphical data. The eye of a fish captures with high detail the pixels near the focal point [36]. This idea is adapted into routing concept such that the topology information is refreshed more frequently for nearby nodes than for distant nodes.

In [18], MPRs are used for estimating node positions in heterogeneous WMNs through anchoring. Greedy and Convex Hull anchor selection methods and their respective estimation accuracies are discussed.

In [41], the authors propose an adaptive multi-channel OLSR based on topology maintenance, which they refer as OLSR-TM. Since Dijkstra is used in OLSR,

the calculation of routes only depend on the collected information. In OLSR-TM, the problem of lacking routing entries is addressed and control messages are sent at adaptive frequencies.

In [3], joint routing and scheduling on heterogeneous ad hoc networks is discussed. OLSR is taken as ground for the routing functionality and the authors propose a task partitioning and scheduling mechanism which distributes  $N$  different tasks having different real-time constraints on  $M$  heterogeneous devices.

Other than all these kinds of theoretical works, there are also other OLSR related studies mostly focusing on the implementation side providing OLSR implementations for ns-2 [21, 45], for GNU/Linux, Windows, iPhone, MAC OS systems [47].

## Chapter 3

# OLSR-Aware Channel Access Scheduling

In this chapter, the features that are common to both of our proposed scheduling algorithms (centralized and distributed) are discussed. The discussion starts with a description of the network model that the proposed algorithms are intended for and continues with the cross-layer weighting scheme used in the proposed scheduling algorithms and the dissemination of the weight information.

### 3.1 Network Model

In our study, we consider a multi-hop wireless mesh network which can be modeled as an undirected graph  $G = (N, L)$ , where  $N$  is the set of nodes and  $L$  is the set of undirected links connecting the nodes in  $N$ . Each node represents a wireless mesh node with a wireless communication range of  $R$ . Link  $l(i, j)$  exists if and only if the distance between the nodes  $i$  and  $j$  is less than or equal to  $R$ , enabling bidirectional communication from  $i$  to  $j$  and from  $j$  to  $i$ . Besides, there exists a set of flows (connections)  $F$  that are active in the network. Each flow has fixed source and destination nodes, and a route determined by OLSR. From this point onwards, the terms flow and connection are used interchangeably.

The targeted system operates in discrete (or, slotted) time. In any time slot, each node may attempt for transmission according to the algorithms discussed in this thesis. A packet transmission attempt is considered successful unless it is interfered by another simultaneous transmission from a node within the node's interference range. The interference range of a node is usually much larger than its transmission range, and nodes more than 2-hops away may also be involved in its interference range [52]. However, we simplify the problem here by ignoring such cases and use 2-hop interference model (i.e. interference range is equal to the transmission range), which assumes that there is no interference between nodes that are separated by more than 2-hops in the physical topology [27].

The following describe the basic features of the network model we are working on:

- Each node is uniquely identifiable.
- Node and time synchronization are available. However, methods for achieving synchronization are out of the scope of this thesis.
- A maximum sized packet can fit into a time slot.
- Nodes are stationary and no further maintenance is done after deployment.
- Communication is bidirectional and communication is established via omnidirectional antennas over a single physical radio channel.
- Each node in the network has a single half-duplex radio and the nodes' radios are always on.
- Each node keeps a single packet queue, not differentiating the packets from different connections.
- Each node is eligible to generate traffic destined to any other node.
- Routing protocol is OLSR.



## 3.2 MPR Based Weighting Scheme

In this section, we present the proposed cross-layer weighting scheme. This weighting scheme utilizes MPR information available at nodes due to use of OLSR routing protocol. In [9], the authors show that in most cases 75% of all MPRs are elected in the first round. Since MPR Selection is mandatory for route calculations and the selection process converges quite fast, we propose that MPR related information might be used by MAC layer within the slot allocation decision phase.

Therefore, we model MPR Based Weighting Scheme with the idea that the nodes that are liable to carry the traffic generated by other nodes should be able to get more transmission opportunities (time slots). In order to achieve this, we assign  $W_X$ , the weight of a node  $X$  as in Eq (3.1):

$$W_X = \text{Size}(\text{MPR Selector Set}(X)) + 1 \quad (3.1)$$

In a multi-hop mesh network, the nodes that forward data on behalf of other nodes carry more traffic than the nodes which do not forward others' data, dealing with their own traffic only. The number of nodes in a MPR node's MPR Selector Set indicates the number of nodes which will possibly route their incoming packets through that particular MPR node. Assuming that all nodes are eligible to generate traffic, for each node that has selected node  $X$  as MPR, 1 unit of weight is added to the weight of node  $X$  along with 1 unit of weight for node  $X$  itself. The number of slots assigned to node  $X$ ,  $Slots_X$ , is proportional to its weight,  $W_X$ ; approximately proportional if a randomized algorithm is used, and exactly proportional if a deterministic algorithm is preferred.

$$\frac{Slots_X}{\sum_{k \in N} Slots_k} \propto \frac{W_X}{\sum_{k \in N} W_k} \quad (3.2)$$

As MPR information is compulsory for route calculation, it is readily available

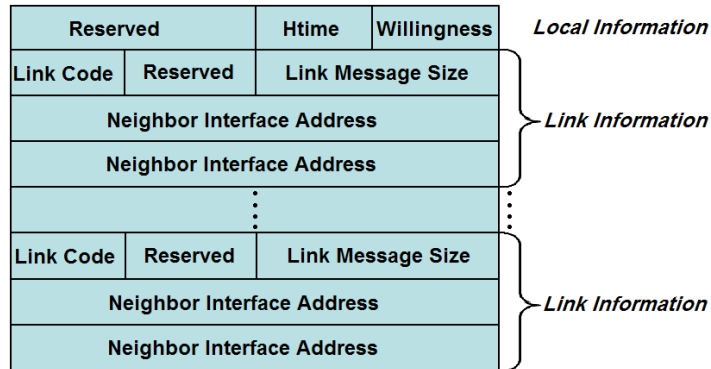


Figure 3.1: RFC 3626 specification on OLSR HELLO Message structure.

and it is sure to be disseminated by the routing layer. As discussed in the next section (Section 3.2.1), this information can be collected by the MAC layer at *no* extra messaging cost. Apart from its having no extra overhead, the main reason for designing such a weighting scheme is that it can be used to approximate the traffic passing through each node without duplex traffic monitoring where all the nodes in the network are eligible to send packets to any other node in the network. Another reason for selecting the size of MPR Selector Set as the weight indicator is that, if the network is not too mobile, once the network stabilizes, the weights calculated using Eq (3.1) will mostly be stable, hence consistent.

### 3.2.1 Dissemination of Weight Information

Once MPR Based Weighting Scheme is decided to be used, it becomes one of the most important parts of the implementation affecting the overall performance of the network. As explained in Section 2.3, OLSR exchanges periodic HELLO messages and collects 2-hop neighborhood and MPR information to be able to construct the routes. This mechanism can be easily extended to carry the weight information as well.

In RFC 3626, the structure of an HELLO message is as given in Figure 3.1. Htime field holds HELLO emission interval (HELLO\_INTERVAL), the time until

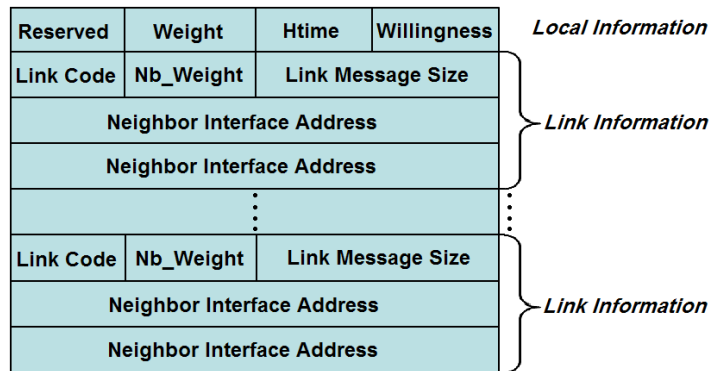


Figure 3.2: Proposed OLSR HELLO Message structure.

the next HELLO message transmission and Willingness field defines the willingness of a node to carry or forward traffic on behalf of other nodes. Link Code specifies information about a particular link. It is formed as the combination of Neighbor Type and Link Type, which were discussed in Section 2.3.1. Link Message Size specifies the message length between two consecutive Link Code fields. Finally, Neighbor Interface Address specifies the address of the neighbor node's associated interface.

In the HELLO message structure specified in RFC 3626, Reserved fields are unused and filled with zeros. Reserved field within the local information section is 2 bytes while Reserved field in the link information section is 1 byte long. This message structure can be easily extended to include weight information for the originating node itself and its listed 1-hop neighbors. There are many different ways of placing information into HELLO messages such as using Reserved fields or adding extra fields to the message.

The proposed modified message structure is shown in Figure 3.2. We propose using the second half of the Reserved field within the local information section for Weight field and substituting the Reserved field in the link information section with Nb\_Weight field. In a single HELLO message, there is only one Weight field, but there might be multiple Nb\_Weight fields depending on the number of the links advertised. Both Weight and Nb\_Weight fields are of 1 byte long. Weight field holds the weight information of the originating node while the Nb\_Weight

field holds the weight information for the neighbor node associated with the advertised link.

Using this new HELLO message structure, every node will be able to collect the weight information of all the nodes in its 2-hop neighborhood via the routing layer control messages without requiring the MAC layer to exchange any further messages. Additionally, there is *no* extra overhead introduced by the proposed MPR Based Weighting Scheme as the unused parts of HELLO messages are utilized for the dissemination of the weight information.

# Chapter 4

## Centralized Channel Access Scheduling

This chapter starts with a brief discussion on the well known heuristics for graph coloring followed by the presentations of the proposed cross-layer OLSR-Aware Centralized Scheduling algorithm and its Non-OLSR-Aware version.

### 4.1 Heuristic-based Algorithms for Vertex Coloring

In designing solutions for channel access scheduling, different forms of graph coloring algorithms are widely used. Given an undirected graph  $G = (N, L)$ , *vertex coloring* is the assignment  $\alpha : N \rightarrow C$  of different colors ( $C$ ) to vertices ( $N$ ) such that no two adjacent vertices get the same color and the number of colors used is minimized [28, 29]. Finding the minimum number of colors in this assignment process is shown to be NP-hard [44].

Since the slot assignment problem is NP-hard, there are several heuristics proposed to provide an approximate solution. Among these heuristics, First-Fit and Degree-Based Ordering are among the most well-known solutions. Algorithm

1 presents the most general form of the First-Fit Vertex Coloring Algorithm [5].

In First-Fit Vertex Coloring Algorithm, every time the algorithm tries to assign a color to a node  $i$ , it starts with checking the already assigned nodes list associated with each color and assigns the first color  $j$  it finds suitable, i.e. non-conflicting with the nodes that are already assigned the color  $j$ . An unused color is assigned to node  $i$  if all used colors are unsuitable.

---

**Algorithm 1:** First-Fit Vertex Coloring Algorithm

---

**Data:** Undirected graph  $G = (N, L)$  where  $N$  is the set of nodes and  $L$  is the set of links connecting the nodes in  $N$ .

**Result:** Nodes in  $N$  are assigned colors such that no two conflicting nodes in  $N$  are assigned the same color.

```

1 begin
2   for  $i \leftarrow 1$  to  $|N|$  do
3     foreach  $Color\ j$  do
4       if  $IsNonConflicting(assignedLst(j), i)$  then
5          $i.Color \leftarrow j$ ;
6          $assignedLst(j).Insert(i)$ ;
7         break;
8 end

```

---

In First-Fit Vertex Coloring Algorithm, no particular strategy is applied for the selection order of the nodes to be colored. However, the order in which the nodes are assigned colors can have a significant effect on the number of colors used. In Maximum Degree First (MDF) Vertex Coloring Algorithm, the vertex with the highest number of neighbors is selected first, providing an intuitively better coloring than First-Fit Vertex Coloring Algorithm [5].

*Distance- $d$  coloring* is a special form of vertex coloring. In *distance- $d$  coloring*, the colors are assigned such that no two vertices of distance  $d$  or less share the same color. TDMA channel access scheduling using 2-hop interference model reduces to *distance-2 coloring* when the time slots are perceived as colors to be assigned and both the primary and the secondary types of conflicts which are explained in Section 2.1.1, are to be avoided.

## 4.2 OLSR-Aware Centralized Scheduling Algorithm (OA-C)

In this section, we present the proposed OLSR-Aware Centralized Scheduling Algorithm (OA-C) which uses a modification of the Distance-2 Maximum Degree First Vertex Coloring Algorithm as its slot allocation mechanism. In our solution, we propose that the size of MPR Selector Set is a good predictor for the amount of traffic that can pass through a node if all the nodes are active and eligible to generate traffic destined to any other node in the network.

---

### Algorithm 2: OLSR-Aware Centralized Scheduling

---

**Data:** Undirected graph  $G = (N, L)$  where  $N$  is the set of nodes and  $L$  is the set of links connecting the nodes in  $N$ .

**Data:**  $S$ : Neighborhood size vector.

**Data:**  $W$ : Weight vector.

**Result:** Each node in  $i$  in  $N$  is assigned  $W_i$  many slots such that no two nodes within the same 2-hop neighborhood are assigned the same slots.

```

1 begin
2    $N \leftarrow \text{Sort}(N, S, \text{Nonincreasing});$ 
3    $\text{cycle\_count} \leftarrow 0;$ 
4   for  $i \leftarrow 1$  to  $|N|$  do
5      $j \leftarrow 1;$ 
6     while  $j < W_i$  do
7        $\text{count} \leftarrow 0;$ 
8       while  $\text{IsFeasible}(i, \text{count}) = \text{FALSE}$  do
9          $\text{count} ++;$ 
10         $\text{slots}[\text{count}].\text{Add}(i);$ 
11        if  $\text{count} > \text{cycle\_count}$  then
12           $\text{cycle\_count} \leftarrow \text{count};$ 
13         $j ++;$ 
14 end

```

---

Therefore, we modify Distance-2 Maximum Degree First Vertex Coloring Algorithm and integrate MPR Based Weighting, which is explained in Section 3.2, into the algorithm. The resulting solution is presented in Algorithm 2. In Algorithm 2, using MPR Based Weighting, each node  $i \in N$  is associated with a

weight  $W_i$  (i.e.  $W_i = \text{Size}(\text{MPR Selector Set}(i)) + 1$ ) and assigned  $W_i$  time slots in a single scheduling cycle.

As we deal with 2-hop neighborhood of the nodes, in this chapter, we define the neighborhood of a node to consist of its 1-hop and 2-hop neighbors. In OA-C, nodes in  $N$  are sorted in a non-increasing order with respect to the size of their neighborhoods. In this way, the nodes whose assignments resolve more conflicts are assigned first and the nodes that are assigned later are less likely to require new slots, resulting in a smaller scheduling cycle length.

In Algorithm 2,  $(\text{cycle\_count} + 1)$  is the number of different slots that are used to schedule all nodes in the network, in other words, the length of the scheduling cycle, resulting in a variable frame size for the centralized scheduling scheme. Depending on the network conditions, OLSR-Aware Centralized Scheduling Algorithm (OA-C) can be configured to run at the end of every frame so that the scheduling mechanism responds to the topological changes in the network (e.g. a new node entering to the network) in a timely manner.

---

**Algorithm 3:** IsFeasible Function

---

**Data:**  $n\_id$ : Node Identifier.

**Data:**  $slot\_count$ : Slot Number.

**Result:** Returns a boolean value indicating whether the slot  $slot\_count$  is feasible for the node  $n\_id$ .

```

1 begin
2    $nbr\_index \leftarrow -1$ ;
3   if  $slots[slot\_count].\text{Empty}()$  then
4      $\lfloor$   $\text{return } TRUE$ ;
5    $nbrLst \leftarrow N_{1,n\_id} \cup N_{2,n\_id} \cup n\_id$ ;
6   foreach  $nbr \in nbrLst$  do
7      $\lfloor$  if  $slots[slot\_count].\text{Contains}(nbr)$  then
8        $\lfloor$   $\text{return } FALSE$ ;
9    $\text{return } TRUE$ ;
10 end

```

---

In Algorithm 3, IsFeasible function is presented. IsFeasible function ensures that the requirements of distance-2 coloring is fulfilled by checking if there are any nodes scheduled to transmit at the given time slot,  $slot\_number$ , within the



2-hop neighborhood of the given node  $n_{id}$ . In the function,  $N_{1,n_{id}}$  and  $N_{2,n_{id}}$  represent the 1-hop and 2-hop neighbors of node  $n_{id}$ , respectively.

Since each node  $i$  should be assigned  $W_i$  many time slots during a single scheduling cycle in OA-C, for each time slot  $T_s$ , we need to check whether node  $i$  has already been inserted to the list of nodes that is to transmit during the time slot  $T_s$ . Therefore, in order to ensure that exactly weight many number of slots are gained by each node in IsFeasible function the combination of  $N_{1,n_{id}}$ ,  $N_{2,n_{id}}$ , and  $n_{id}$  are checked against conflicts.

### 4.3 Non-OLSR-Aware Centralized Scheduling Algorithm (NOA-C)

---

#### Algorithm 4: Non-OLSR-Aware Centralized Scheduling

---

**Data:** Undirected graph  $G = (N, L)$  where  $N$  is the set of nodes and  $L$  is the set of links connecting the nodes in  $N$ .

**Data:**  $S$ : Neighborhood size vector.

**Result:** Nodes in  $N$  are assigned time slots such that no two nodes within the same 2-hop neighborhood are assigned the same slot.

```

1 begin
2    $N \leftarrow \text{Sort}(N, S, \text{Nonincreasing});$ 
3    $\text{cycle\_count} \leftarrow 0;$ 
4   for  $i \leftarrow 1$  to  $|N|$  do
5      $\text{count} \leftarrow 0;$ 
6     while  $\text{IsFeasible}(i, \text{count}) = \text{FALSE}$  do
7        $\text{count} ++;$ 
8        $\text{slots}[\text{count}].\text{Add}(i);$ 
9       if  $\text{count} > \text{cycle\_count}$  then
10         $\text{cycle\_count} \leftarrow \text{count};$ 
11 end

```

---

We next discuss a Non-OLSR-Aware version of our OA-C algorithm, so called Non-OLSR-Aware Centralized Scheduling Algorithm (NOA-C), which does not make use of MPR Based Weighting. It makes use of the above-mentioned Distance-2 Maximum Degree First heuristic which is widely used/extended in

many different studies [5, 40]. In Algorithm 4, we present a high level description of how we implement this algorithm in our framework so that a fair comparison of the discussed scheduling schemes (OA-C and NOA-C) becomes possible.

Although NOA-C is quite similar to OA-C algorithm, since MPR Based Weighting Scheme is not used, all the nodes are assigned a single slot in a single scheduling cycle instead of being assigned weight-many slots as in OA-C, leading all nodes to be assigned equal number of slots.

# Chapter 5

## Distributed Channel Access Scheduling

In this chapter, firstly, the basic features that are common to the proposed distributed cross-layer channel access scheduling scheme, OLSR-Aware Distributed Channel Access Scheduling, and its Non-OLSR-Aware version are listed. Then, OLSR-Aware Distributed Channel Access Scheduling (OA-D) and Non-OLSR-Aware Distributed Channel Access Scheduling (NOA-D) algorithms are presented, respectively.

### 5.1 Basic Features

In this section, the basic features that are common to both of the distributed channel access scheduling schemes discussed in this chapter are listed. These features also highlight the differences among the centralized and distributed algorithms presented in this thesis.

1. For the distributed algorithms presented in this chapter, the number of slots in a frame (*FRAME\_SIZE*) is fixed. On the other hand, the frame size is variable for the centralized schemes as discussed in Chapter 4.

2. Each node decides the time slots it will use for transmission based on local cross-layer information. In the centralized scheduling algorithms, a central scheduler node decides the schedule for all the nodes in the network.
3. Both OA-D and NOA-D are pseudo-randomized while the centralized algorithms (OA-C and NOA-C) are deterministic.

## 5.2 OLSR-Aware Distributed Scheduling Algorithm (OA-D)

In this section, the details of our OLSR-Aware Distributed Scheduling Algorithm (OA-D) are presented. In OA-D, each node determines the time slots it will use for transmission based on the local information it has about its 1-hop and 2-hop neighbors and their MPR based weights collected by OLSR. It is a pseudorandom weighted channel access scheme which requires no schedule negotiation messages and no negotiation delay. Since all the nodes have consistent data about their 2-hop neighborhood and their respective weights, nodes can run their algorithms without having to wait for their neighbors' approval signals.

For this access scheme, *FRAME\_SIZE* denotes the number of slots in a frame and it is a network parameter which remains fixed upon initialization. OA-D is independently run by each node  $i$  at the end of every frame in order to select the slots it is eligible to transmit during the next frame. OA-D algorithm is presented in Algorithm 5.

In OA-D, each node  $i$  generates as many agents as its weight,  $W_i$ , where all of its agents compete to win time slots on its behalf. Each agent of node  $i$  is defined by an *agentID*, which is formed as the concatenation of the node identifier and a number from 0 to  $W_i - 1$ . In the first two steps of the algorithm, *agentIDs* for the hosting node's agents and the *agentIDs* of its 1-hop and 2-hop neighbors' agents are generated and put into *localLst* and *nbrLst*, respectively. All the agents generated in these two steps are involved in all contentions held throughout the frame.

---

**Algorithm 5:** OLSR-Aware Distributed Scheduling

---

**Data:** Topology and weight information for 2-hop neighborhood of node  $i$ .**Result:** The set of time slots node  $i$  is eligible to transmit during the next frame.

```

1 begin
2    $localLst \leftarrow \text{FormLocalAgents}(i)$ ;
3    $nbrLst \leftarrow \text{FormNeighborAgents}(i)$ ;
4    $contenders \leftarrow nbrLst \cup localLst$ ;
5   for  $j \leftarrow 1$  to  $FRAME\_SIZE$  do
6      $slotID \leftarrow \text{FormSlotID}(FrameCount, j)$ ;
7      $res\_set \leftarrow \text{MeshElection}(slotID, contenders)$ ;
8      $winner \leftarrow \text{FindMax}(res\_set)$ ;
9     if  $localLst.Contains(winner)$  then
10    |    $slots[j].status \leftarrow WON$ ;
11 end

```

---

In the for loop, a separate contention is held for each time slot in a frame. MeshElection function returns a set of pairs where each pair involves the *agentID* and its corresponding *SmearValue*, which is described below. The agent with the largest *SmearValue* is then elected as the winner of the contended time slot. If the winner agent's *agentID* belongs to *localLst*, then the node marks the slot as one of the slots it is eligible to transmit (i.e. sets the slot's status to 'WON').

MeshElection function in OA-D algorithm is adapted from MeshElection algorithm specified in 802.16-2004 standard [1] as a part of the distributed EBTT mechanism which is responsible for the allocation of control slots such that the control messages are transmitted in a collision-free manner in 2-hop neighborhood without requiring explicit schedule negotiation. MeshElection function's first parameter, *slotID*, is formed by FormSlotID function as the concatenation of the contended frame count, *FrameCount*, and the contended slot number,  $j$ . The *SmearValue* is obtained as

$$SmearValue = smear(agentID \wedge slotID) \quad (5.1)$$

where smear function is the hashing function given in 802.16-2004 standard [1]

which converts a uniform value to an uncorrelated uniform hash value, through the use of mixing. Smear function uses only simple arithmetic operations, which can be computed very quickly in practice. This is the reason why we preferred using smear function over a random number generator.

Recall that the weight of each node is calculated via Eq(3.1). Each node might have at least 0 and at most  $N - 1$  nodes in its MPR Selector Set where  $|N|$  is the number of nodes in the network. This also implies that the weight of any node remains within  $[1, N]$  range leading the worst case complexity of MeshElection function to be  $O(N^2)$  and the worst case complexity of OA-D algorithm to be  $O(FRAME\_SIZE * N^2)$ .

### 5.3 Non-OLSR-Aware Distributed Scheduling Algorithm (NOA-D)

In this section, we discuss Non-OLSR-Aware Distributed Scheduling Algorithm (NOA-D), which is presented in Algorithm 6.

---

**Algorithm 6:** Non-OLSR-Aware Distributed Scheduling

---

**Data:** Topology information for 2-hop neighborhood of node  $i$ .  
**Result:** The set of time slots node  $i$  is eligible to transmit during the next frame.

```

1 begin
2    $nbrLst \leftarrow \text{FormNeighborAgents}(i)$ ;
3    $contenders \leftarrow nbrLst \cup agentID$ ;
4   for  $j \leftarrow 1$  to  $FRAME\_SIZE$  do
5      $slotID \leftarrow \text{FormSlotID}(FrameCount, j)$ ;
6      $res\_set \leftarrow \text{MeshElection}(slotID, contenders)$ ;
7      $winner \leftarrow \text{FindMax}(res\_set)$ ;
8     if  $agentID = winner$  then
9        $slots[j].status \leftarrow WON$ ;
10 end

```

---

The differences between OA-D and NOA-D are quite similar to those between OA-C and NOA-C. Similar to NOA-C, NOA-D does not use the MPR Based

Weighting Scheme introduced in Section 3.2. Therefore, in NOA-D, the weights of all nodes are equal to 1 as well. In addition, despite the use of local information and pseudorandomness involved in the decision mechanism, the distribution of the number of time slots gained by the nodes when NOA-D is used is close to uniform, especially when compared against the number of time slots nodes gain when OA-D algorithm is used.

## 5.4 Additional Discussion

In our current implementation, we keep the radios of all nodes always on. However, energy preserving can be introduced into OA-D and NOA-D very easily. In any OLSR enabled network, the nodes need to hear HELLO messages from all of their 1-hop neighbors. Therefore, OA-D and NOA-D generated schedules can be classified as broadcast schedules. In a broadcast schedule, it would be enough for a node to only keep its radio open when the winner is itself or one of its 1-hop neighbors. Besides, by the use of the pseudo-random distributed scheduling algorithms, each node knows the winner of a particular slot within its 2-hop neighborhood. Hence, it is possible to exploit this broadcast schedule property as an energy-preserving method in distributed scheduling schemes discussed in this chapter.

# Chapter 6

## Simulation Implementation and Results

This chapter starts with elaborating on the simulation implementations for the algorithms discussed in this thesis. The rest of this chapter reports the obtained simulation results and provides insightful analysis of these results. Different aspects of the obtained simulation results such as the achieved concurrency levels, achieved performance metrics under uniform/nonuniform traffic patterns, the effects of changing the network size and the maximum queue length are investigated in different subsections, respectively. In the final section of this chapter, we discuss the simulation results obtained on a sample 15-node network topology in detail.

### 6.1 Simulation Implementation

In this section, detailed information about the simulation implementations of the discussed scheduling schemes is given. All schemes discussed in this thesis are implemented in ns-2.31 environment as MAC classes. The implementation of each scheme is composed of two parts:



1. The implementation of the required changes in OLSR module.
2. The implementation of the proposed algorithm within a MAC class.

For OLSR implementation, we use UM-OLSR-0.8.8 for ns-2.31, as it is compliant with RFC 3626 and provides MAC layer feedback support which is useful in detecting lost links [45]. We replaced RFC 3626 specified HELLO message structure (Figure 3.1) with our proposed HELLO message structure (Figure 3.2) in order to disseminate the weight information of the originating node as well as its known neighbors' as described in Section 3.2. In addition, we have extended the OLSR table structures to hold the weight information.

For the implementation at MAC layer, we took a basic non-concurrent TDMA based MAC protocol as starting point which comes with ns-2 implementations from ns-2.23 onwards. This protocol does not support concurrent transmissions hence it does not exploit the slot reusability available in multi-hop environments [19]. In the implementation of this protocol, each TDMA frame contains data transmission slots and a preamble for supporting power-save mode. The number of data transmission slots in a frame is equal to the number of nodes in the network. During each frame, every node takes turn once even if it has no data to send. The preamble contains the IDs of the nodes which will receive data during each slot in the next frame. Being aware of the slots that they will send/receive, in order to reduce the power consumption, the nodes can sleep during the slots they do not send/receive anything.

However, this implementation has obvious drawbacks as it produces very low throughput, not taking the slot reusability and the traffic into account and not modeling the centralized scheduling to the full extent as there is no central controlling node that dictates the schedules of the remaining nodes.

Building on the code of this non-concurrent TDMA MAC implementation in ns-2, we implemented our distributed and centralized channel access schemes as MAC protocols, allowing multiple nodes to transmit concurrently. We also eliminated the use of preambles from the protocol implementation.

The only exception to the use of the proposed algorithms is seen in the very first frame of the simulation since it takes some time for the topology information to converge. All the nodes start with empty neighbor lists. Each node initially assumes that it is the only node in the contention context and tends to select every slot of the first frame as eligible to transmit in the distributed decision case. Similarly, in centralized decision case, before the 2-hop neighborhood information for all the nodes converge, the centralized scheduler perceives some of the conflicting nodes as nonconflicting and tends to assign the same time slots to conflicting nodes.

In order to avoid these initial collisions, only in the first frame, nodes act overly precautionary and every node selects its slots for transmission as if the network is a 1-hop network. In a scenario, where there are 20 nodes in the network and the *FRAME\_SIZE* is set to 100, then a node, say *node-7* selects 7<sup>th</sup>, 27<sup>th</sup>, 47<sup>th</sup>, 67<sup>th</sup>, 87<sup>th</sup> slots for transmission. After the first frame, the proposed scheduling algorithms are used accordingly. Once the 2-hop neighborhood information converges, the proposed schemes generate collision-free schedules.

## 6.2 Simulation Results and Analysis

In this section, we report our ns-2 based simulation results and provide comparisons for the following scheduling schemes:

1. OLSR-Aware Centralized Scheduling (OA-C)
2. Non-OLSR-Aware Centralized Scheduling (NOA-C)
3. OLSR-Aware Distributed Scheduling (OA-D)
4. Non-OLSR-Aware Distributed Scheduling (NOA-D)

We define the following performance metrics and present simulation results that illustrate how these metrics change under both uniform and nonuniform

traffic patterns while packet generation rate, network size and the number of active flows in the network change:

1. **Number of Packets Delivered:** The total number of packets received at the application layer by all receivers during the simulation.
2. **Packet Drops:** The total number of application layer data packets that could not be delivered within the simulation time.
3. **Packet Delivery Ratio:** The ratios of the number of packets delivered at application layer to the number of packets generated at application layer for the whole network, which is given by Eq (6.1).

$$\text{Packet Delivery Ratio} = \frac{\text{Packets Delivered}}{\text{Packets Generated}} \quad (6.1)$$

4. **Average End-to-End Delay:** End-to-end delay is the time taken for a packet to be transmitted across a network from source to destination. Average end-to-end delay is calculated for all packets that are successfully received at the application layer by the destination nodes.

Table 6.1 lists the parameters that are used in the ns-2 simulations.

Table 6.1: Ns-2 simulation parameters.

Parameter	Value
<b>OLSR Parameters (RFC)</b>	
<i>Hello Interval</i>	2 s
<i>TC Interval</i>	5 s
<b>MAC Parameters</b>	
<i>Bandwidth</i>	3 Mbps
<i>Max Packet Length</i>	1500 bytes
<i>Frame Size</i>	50 slots
<b>Traffic Parameters</b>	
<i>Packet Size</i>	200 bytes
<i>Traffic Generation Rate</i>	50-700 bps

### 6.2.1 Concurrency Levels Achieved

Before presenting our simulation results with uniform/nonuniform traffic distributions, we discuss the average concurrency levels achieved by different scheduling schemes, which are presented in Figure 6.2(a). Concurrency is defined as the average number of nodes that are able to transmit concurrently without conflicting, and it is calculated using Eq (6.2). In Eq (6.2),  $S$  denotes the set of time slots for the whole simulation time, while  $S[k].size()$  stands for the number of nodes that are allocated concurrently (i.e. in a conflict-free manner) to the same time slot  $k$ .

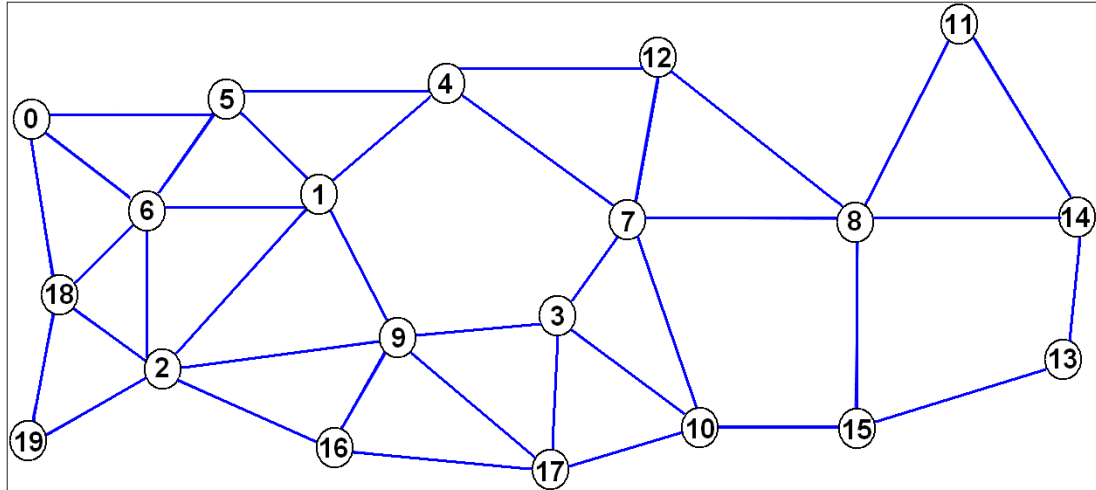
$$Concurrency = \frac{\sum_{k=1}^{|S|} S[k].size()}{|S|} \quad (6.2)$$

The following are the two main factors that affect the concurrency levels achieved by the discussed scheduling schemes:

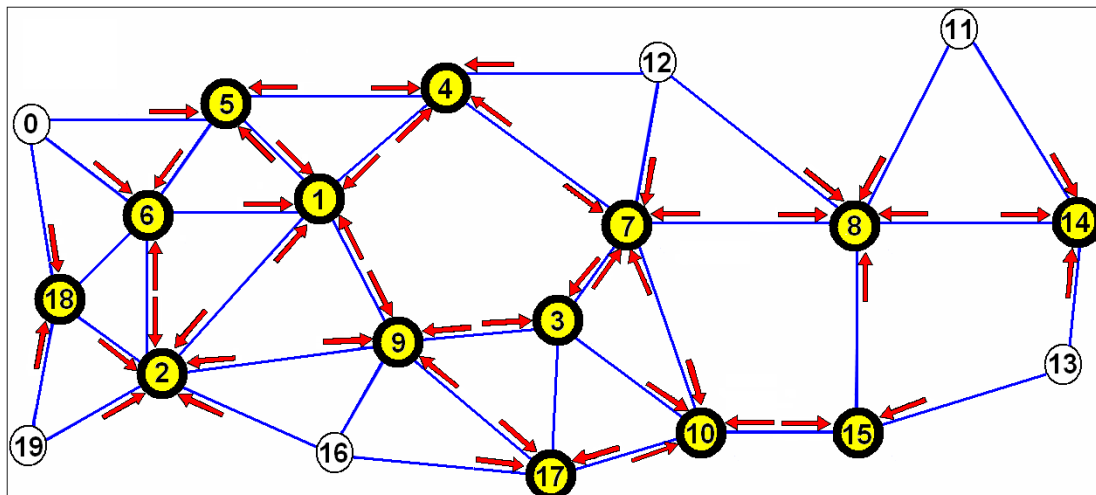
1. Computation Method (Distributed / Centralized)
2. Weighting Scheme (MPR Based / Uniform with no weights)

The computation method has a significant impact on the level of concurrency achieved by the discussed scheduling schemes mostly because they use different levels of information. In centralized schemes, the central scheduler exploits global information while in fully distributed schemes the nodes make decisions using only the local information available.

In both of the distributed schemes (OA-D and NOA-D), each node locally runs an election (e.g. MeshElection) independent from the other nodes. Taking the topology in Figure 6.1(a) as a reference, consider the scenario in which *node 9* loses to *node 7* because *node 7* has a larger *SmearValue* for the slot  $T_s$ . In this situation, *node 9* refrains from transmission during  $T_s$ . For the same time slot, *node 7* also runs an independent local election and assume that *node 12* has a larger *SmearValue* than that of *node 7*'s. In this case, *node 7* also refrains

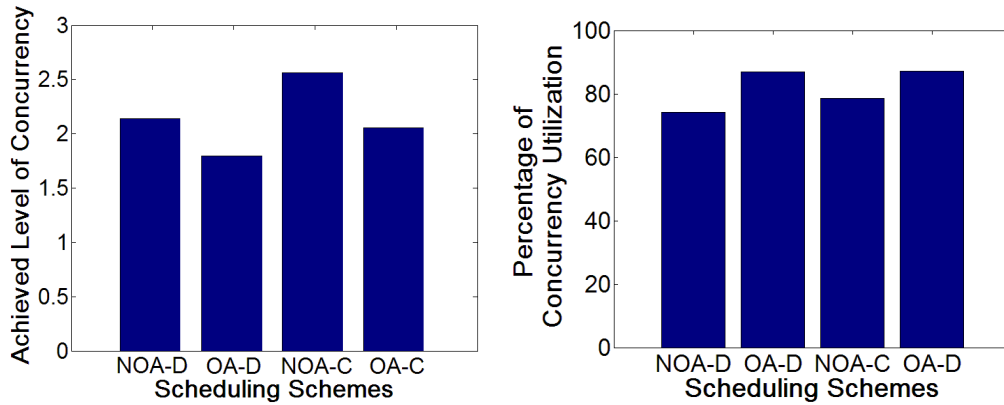


(a) Without MPRs highlighted.



(b) With MPRs highlighted.

Figure 6.1: Sample topology for 20-node network.



(a) Achieved concurrency levels averaged over 15 different 20-node networks. (b) Percentages of concurrency utilization averaged over 15 different 20-node networks.

Figure 6.2: Results on achieved/utilized concurrency levels.

from transmission during  $T_s$ . Indeed, *node 9* and *node 12* could have transmitted concurrently as the distance between these two nodes is more than two hops. However, since they are more than two hops away from each other, none of them can predict what the other node's *SmearValue* will be. In contrast, centralized scheduling schemes are able to resolve such problems and prevent unnecessary refrainment. Hence, centralized scheduling schemes provide higher levels of concurrency than their distributed counterparts which can be observed in Figure 6.2(a).

In Figure 6.2(a), the impact of the weighting schemes on the achieved concurrency levels is also clearly visible. The nodes that can suppress the transmissions from many other nodes are usually selected as MPR nodes since they usually have large 2-hop neighborhoods. In scheduling schemes using MPR Based Weighting, such nodes have more weight and win more slots, leading to a decrease in the average number of nodes that can transmit concurrently during a time slot when compared to the scheduling schemes not using any weighting scheme.

However, as discussed in the following subsections, the level of concurrency is not the only metric that determines the amount of traffic that can be delivered. In other words, there might be nodes provided with concurrent transmission opportunities more than they actually need. This situation causes the achieved

level of concurrency to appear higher, although the expected performance gain in terms of the number of packets delivered successfully is not obtained because of not utilizing the allocated slots.

In Figure 6.2(b), the percentages of slot utilization is presented. The presented values are calculated as the average of slot utilization under the same uniform scenario over 15 different 20-node networks. In the simulated scenarios, we used high traffic rates in order to ensure that most of the nodes will have packets in their queues most of the time, so that the impact of the scheduling scheme will become more accurately visible. Figure 6.2(b) shows that OLSR-Aware schemes improve the utilization of the allocated time slots over their Non-OLSR-Aware versions around 8 - 13%.

## 6.2.2 Simulation Results with Uniform Traffic Scenarios

The four scheduling schemes (OA-C, NOA-C, OA-D and NOA-D) are simulated and compared under the same uniform traffic scenarios in which every node generates a connection to every other node in the network using CBR traffic, resulting in  $O(n^2)$  connections. Each of these connections start and end at the same time. In CBR scenarios, the packet generation rates (in bps) and the packet sizes (in bytes) are kept the same for all connections in a single scenario and different packet generation rates are applied over different simulation scenarios.

In Figure 6.3, we report our uniform traffic simulation results. We present the average performance metrics achieved by the scheduling schemes, where the average is taken over 15 randomly generated connected topologies each with 20 nodes. In Figure 6.1(a), a sample topology is depicted. In Figure 6.1(b), MPR nodes are highlighted and arrows directed towards MPR nodes from each of their selectors are included.

Figures 6.3(a), 6.3(b) and 6.3(c) illustrate the number of delivered/dropped packets and the packet delivery ratio, respectively. In these figures, it is observed that OLSR-Aware schemes can deliver more packets than their Non-OLSR-Aware

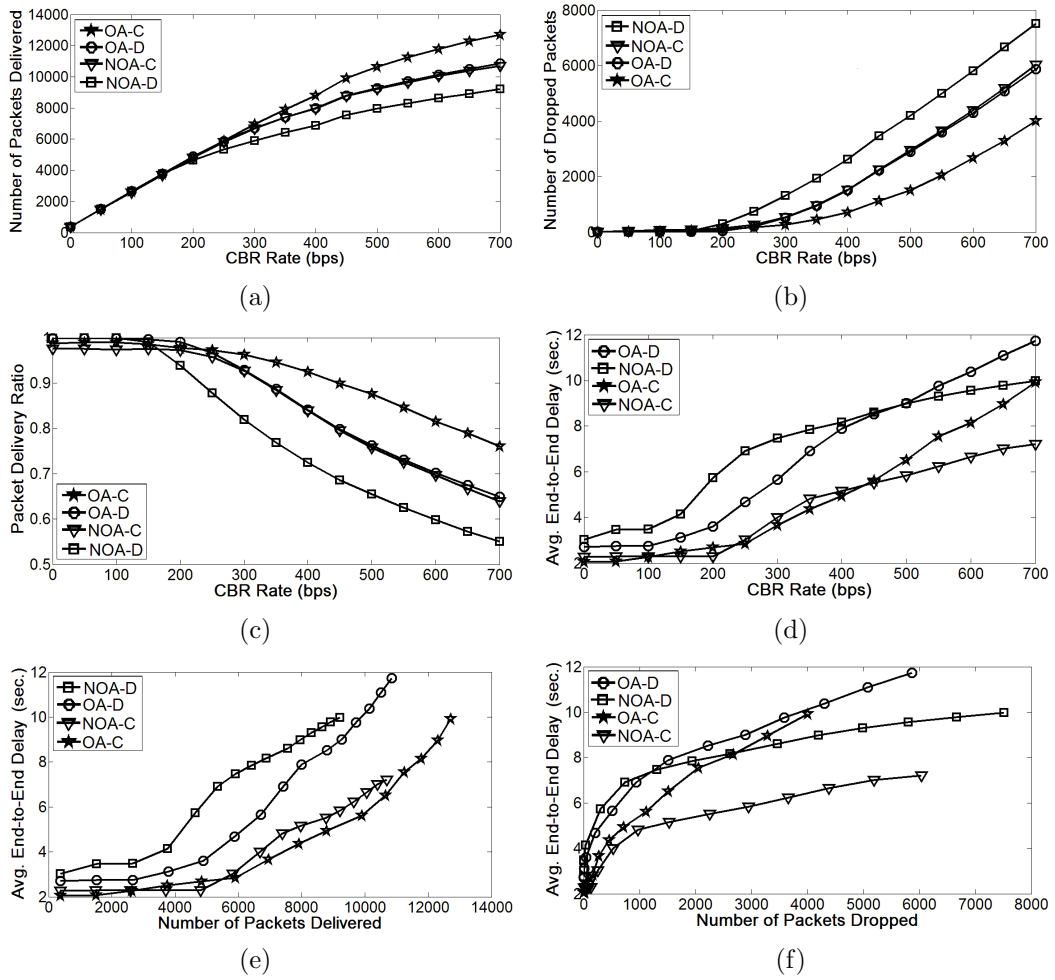


Figure 6.3: Averaged uniform traffic simulation results on 20-node networks.



counterparts at the cost of higher delay, especially when the network load increases. The higher delay is, however, observed due to the fact that OLSR-Aware scheduling schemes are able to deliver more packets even if they are delayed while Non-OLSR-Aware scheduling schemes have to drop them.

On the other hand, Figures 6.3(d), 6.3(e) and 6.3(f) illustrate the delay behaviors of the four discussed scheduling schemes. In Figure 6.3(d), the average end-to-end delay turns out to be less in the centralized schemes than it is in the distributed schemes. In addition, when distributed and centralized cases are examined separately, in both cases we see that Non-OLSR-Aware algorithms exhibit a slower increase in the average end-to-end delay than OLSR-Aware algorithms. This is due to the fact that the average end-to-end delay is calculated only for the packets that could be delivered successfully.

The results presented in Figures 6.3(e) and 6.3(f) explicitly demonstrate the trade-off between the number of packets dropped/delivered and the average end-to-end packet delay. Consider the delay behaviors of NOA-C and OA-C in Figure 6.3(e). The number of packets delivered by OA-C cannot be reached by NOA-C. Up to the maximum number of packets NOA-C can deliver, the average end-to-end delay of OA-C is less than that of NOA-C. OA-C is able to deliver packets beyond this point, however, its average end-to-end delay increases; exceeding NOA-C's average end-to-end delay. Similar reasoning applies for the delay behaviors of NOA-D and OA-D.

There are two main conclusions that can be drawn from the simulation results presented in Figure 6.3. Firstly, the centralized schemes achieve better results than their distributed counterparts. Secondly, OLSR-Aware algorithms using MPR Based Weighting perform better than their Non-OLSR-Aware versions in terms of the number of delivered/dropped packets and the packet delivery ratio in both of the distributed and the centralized cases.

In this respect, the obtained results comply with our expectations regarding the benefit of MPR Based Weighting as well as the simulation results on the respective concurrency levels achieved by these four scheduling schemes.

The results presented in Figure 6.3 also justify our argument that the level of concurrency is not the only metric affecting the end-to-end throughput and delay since OLSR-Aware algorithms can deliver more packets despite the fact that Non-OLSR-Aware algorithms provide more concurrency. The level of concurrency achieved loses its importance when the transmission opportunities are not provided to the nodes that can utilize them effectively and when they are wasted.

### 6.2.3 Simulation Results with Nonuniform Traffic Scenarios

In this section, we present our simulation results under a number of different nonuniform traffic scenarios in which some nodes either create or receive more traffic than the other nodes. In this set of simulations, CBR traffic is used and CBR rate is held fixed at 500 bps for all connections. The simulations last for 200 seconds. All the connections start at sometime between 25<sup>th</sup> and 50<sup>th</sup> second and end at some time between 125<sup>th</sup> and 150<sup>th</sup>.

The connection pairs are chosen randomly such that in a single simulation scenario, there is only 1 connection (flow) from a particular source node to a particular destination node. That is, if  $(x, y)$  is selected as (source, destination) for a connection, it is still possible that  $(y, x)$  is selected for another connection in the same scenario. However, only 1 connection is allowed to have  $(x, y)$  as the (source, destination). There are 20x19 possible (source, destination) pairs in a 20-node network. Starting with a scenario of 20 connections and increasing the number of connections with step size of 30 until the number of connections reach 380, 13 different simulation scenarios are formed.

The related simulation results plotted in Figure 6.4 represent the average of the simulation results where the average is obtained using 15 different 20-node networks. Figures 6.4(a), 6.4(b) and 6.4(c) show the number of delivered/dropped packets and the packet delivery ratio while Figures 6.4(d), 6.4(e) and 6.4(f) depict the delay behaviors of the four discussed scheduling schemes.

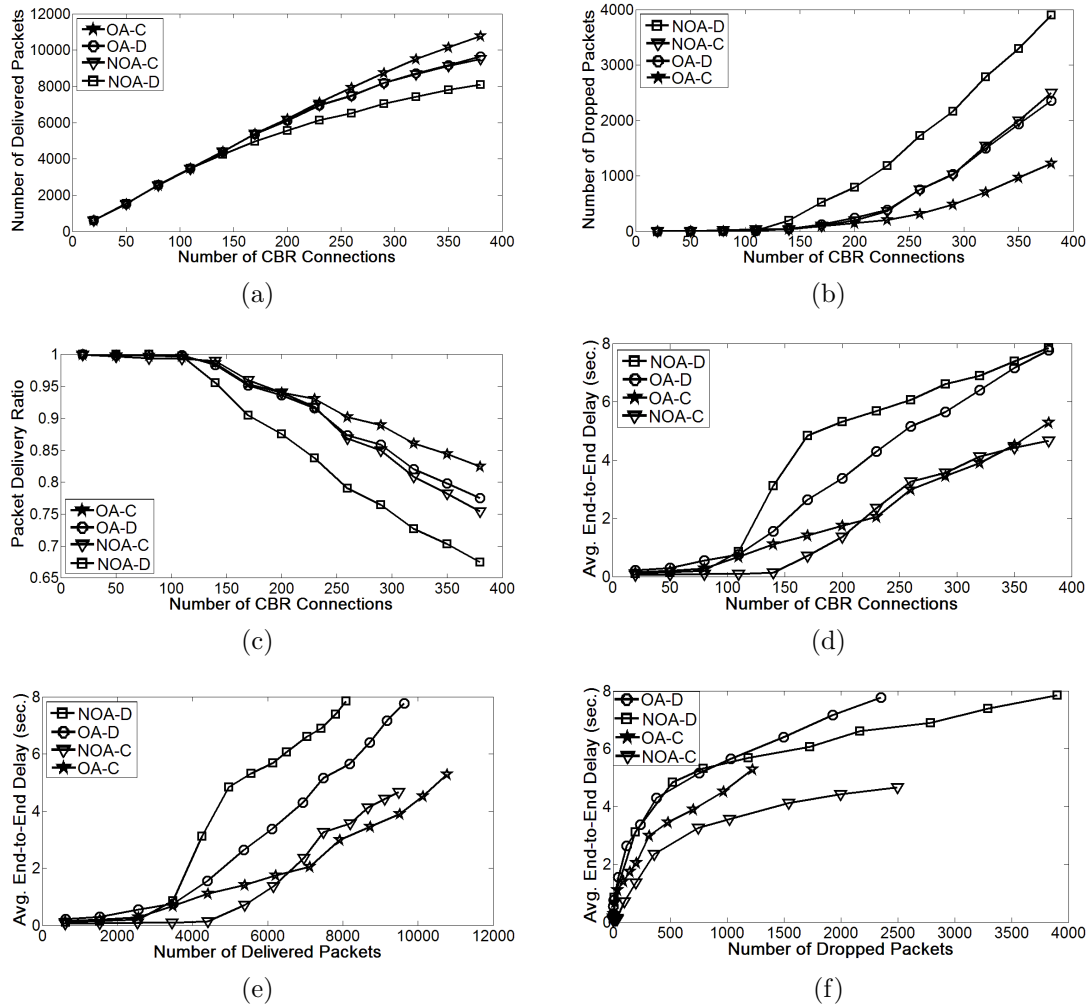


Figure 6.4: Averaged nonuniform traffic simulation results on 20-node networks.

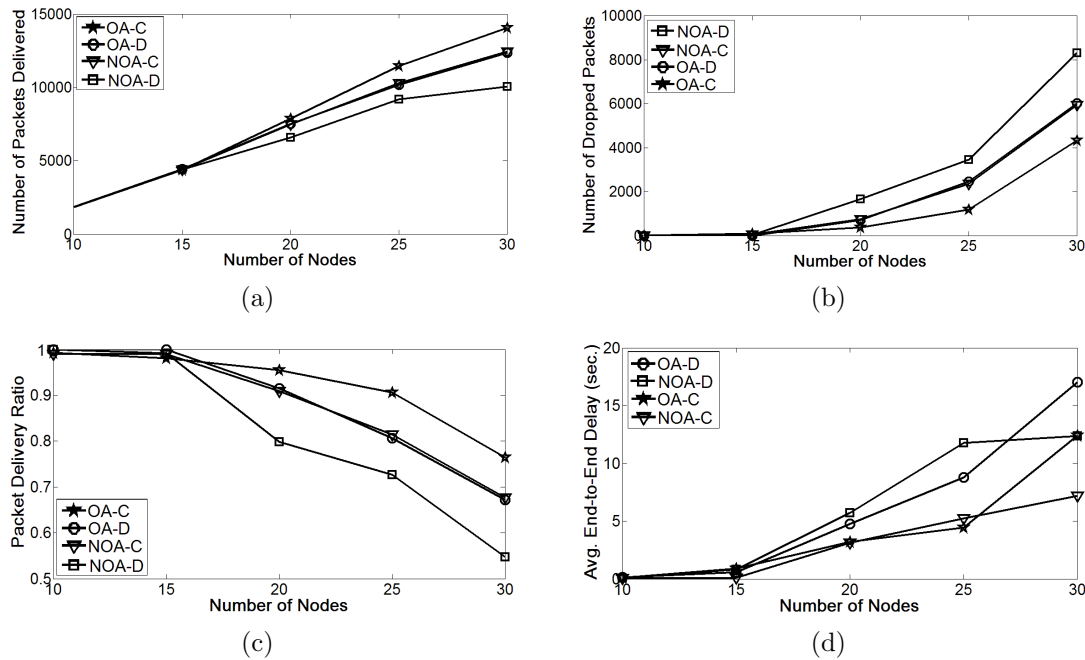


Figure 6.5: The effects of network size.

All the results presented in Figure 6.4 are investigated while the number of active connections in a network is changed. When the number of connections is 380 (19x20) for a 20-node network, the simulation scenario is very close to the uniform scenario, except the start and end times of the connections. As the number of active connections in a network is increased, since CBR rate is fixed for all experiments presented in this section, the load of the network increases.

The nonuniform traffic scenarios' results are quite similar to the results obtained with uniform traffic scenarios and can be interpreted with similar reasoning. Nonuniform traffic simulation results are important as they indicate that using MPR Based Weighting Scheme within the scheduling procedure is useful not only in uniform traffic patterns, but also in general traffic patterns.

#### 6.2.4 The Effects of Network Size

In order to study the effects of changing the network size, we extend our simulation results with general traffic patterns (i.e. nonuniform traffic scenarios).

In this set of experiments, CBR rate is fixed (set to 500 bps) while the network size is changed between 10 and 30 with a step size of 5. For each network size, we created 10 different random topologies and averaged their results. Figures 6.5(a), 6.5(b), 6.5(c) and 6.5(d) illustrate how the number of delivered/dropped packets, the packet delivery ratio and the average end-to-end delay change while the network size changes, respectively.

In the presented results, OA-C again performs better than the other schemes in terms of the number of delivered/dropped packets and the packet delivery ratio, while NOA-D is the scheme with the least performance. In the presented results, the average number of delivered/dropped packets and the packet delivery ratio values obtained for OA-D and NOA-C are quite similar. However, these two schemes behave very differently on different topologies. As a general observation, OA-D performs better than NOA-C when the network is well-spread, containing long paths, where the packets need to be forwarded over multiple hops. In contrast, NOA-C performs better than OA-D when the network is denser and the data forwarding paths are shorter.

In addition, the delay results are also consistent with the uniform/nonuniform traffic simulation results. The distributed schemes incur more delay than the centralized schemes and the delays for the OLSR-Aware algorithms tend to increase more steeply when compared against their Non-OLSR-Aware versions since they can deliver more packets within the given simulation time.

### 6.2.5 The Effects of the Maximum Queue Size

In our simulations, we have also observed that, for all the scheduling schemes discussed in this thesis, the maximum queue length has a significant effect on the overall performance of the network. The maximum queue size directly affects the number of dropped/delivered packets, the average end-to-end delay, the packet delivery ratio.

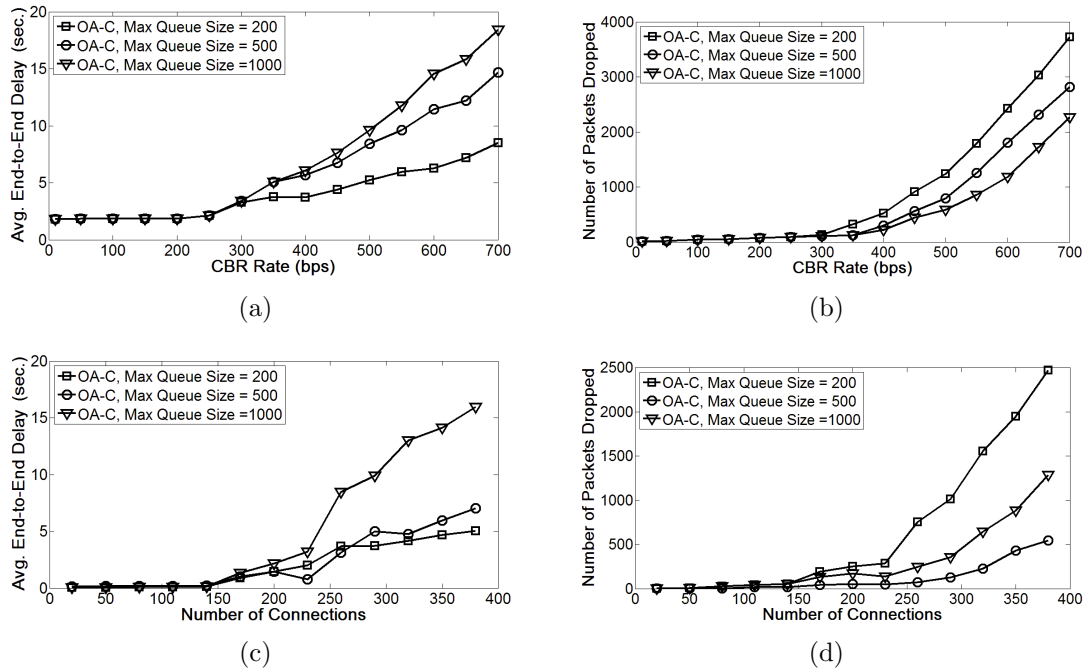


Figure 6.6: The effects of the maximum queue size on the performance of OA-C.

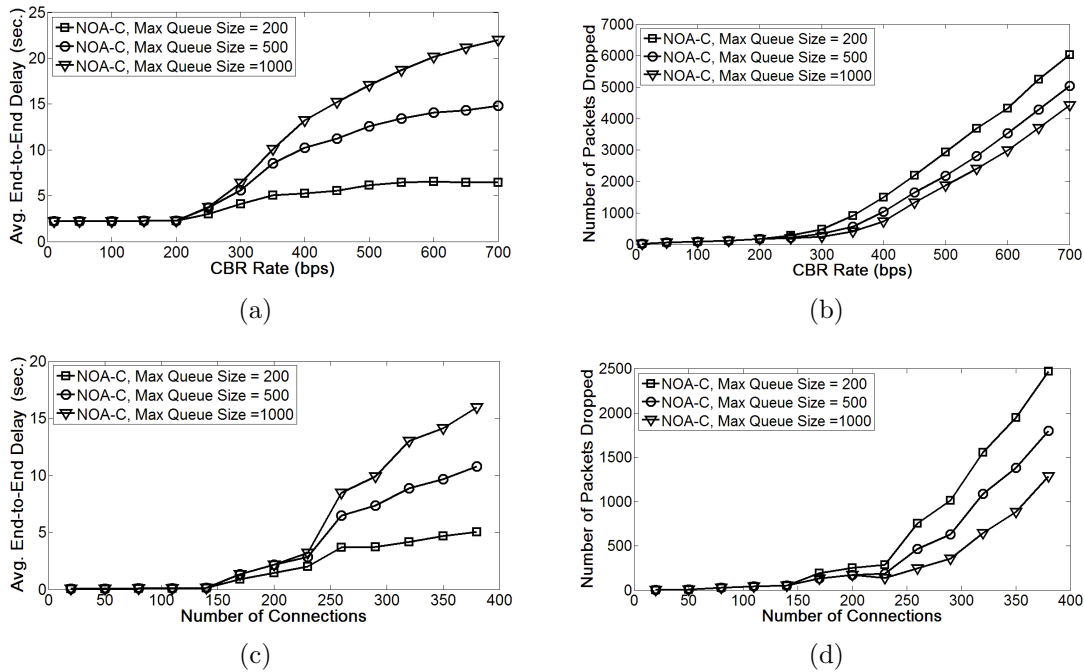


Figure 6.7: The effects of the maximum queue size on the performance of NOA-C.

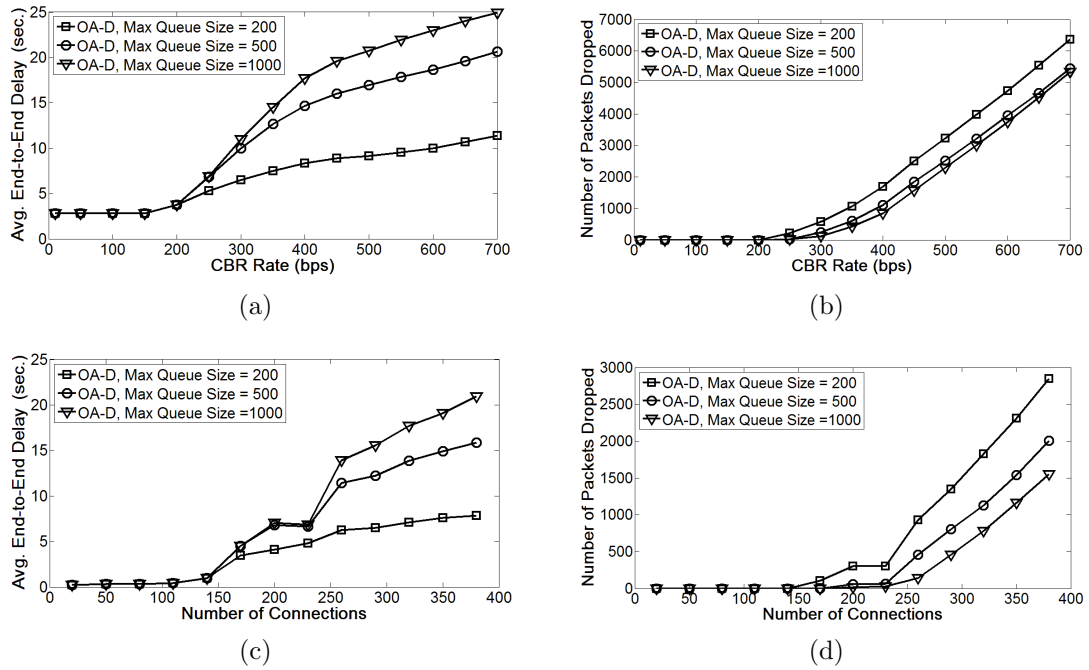


Figure 6.8: The effects of the maximum queue size on the performance of OA-D.

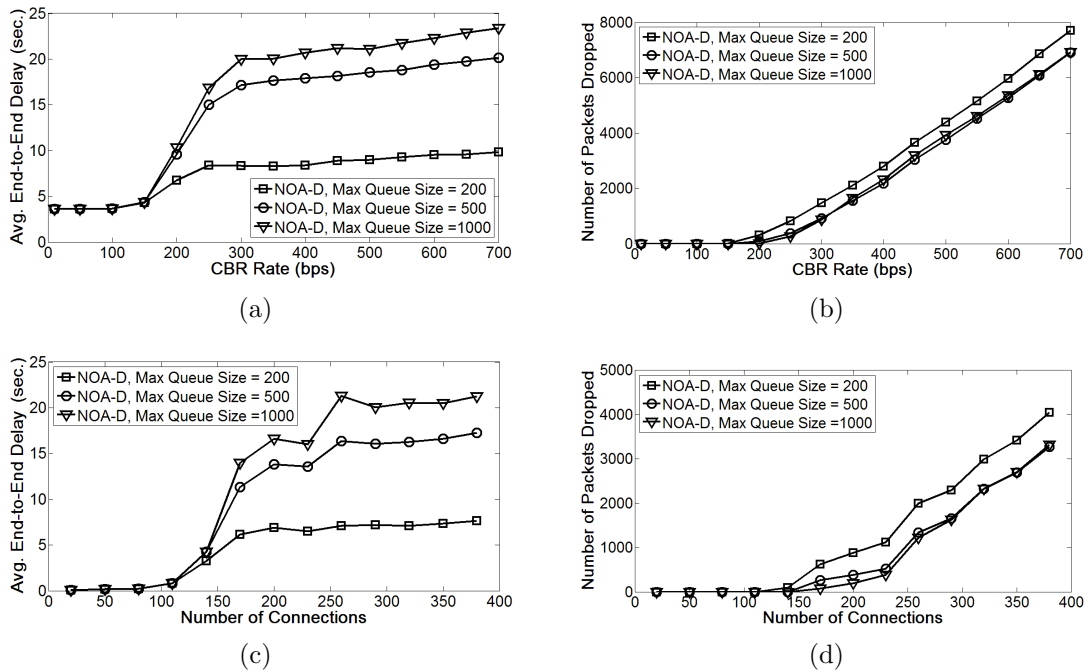


Figure 6.9: The effects of the maximum queue size on the performance of NOA-D.

Figures 6.6 - 6.9 present averaged simulation results taken on 4 different 20-node networks under both uniform and non-uniform traffic patterns for OA-C, NOA-C, OA-D, and NOA-D, respectively. In Figures 6.6 - 6.9, parts (a) and (b) depict the simulation results under uniform traffic patterns while parts (c) and (d) present results from non-uniform traffic patterns. For each scheduling scheme, each of the simulation scenarios discussed in Sections 6.2.2 and 6.2.3 is run 3 times on each topology, once for each different queue limit.

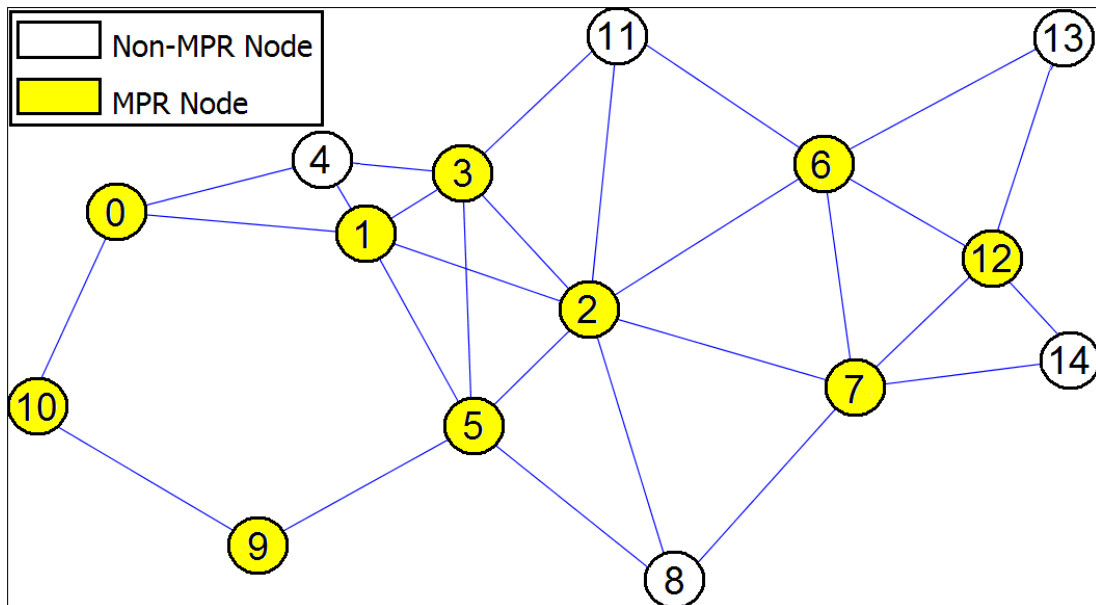
In general, the simulation results presented in Figures 6.6 - 6.9 indicate that, in all four discussed scheduling schemes, when the maximum queue size is increased, the number of dropped packets decreases at the cost of increasing the average end-to-end delay. With larger queue capacities, more packets seize the opportunity of being queued rather than being dropped when compared to the case where the nodes' queue capacities are smaller. Therefore, with larger queue capacities, greater number of delayed packets are transmitted, leading to an increased average end-to-end delay and decreased number of dropped packets.

### 6.2.6 Case Study: Simulation Results Obtained on a Sample 15-Node Network

In this section, detailed simulation results illustrating the slot allocation behaviors of the four discussed scheduling schemes (OA-C, NOA-C, OA-D and NOA-D) on a randomly generated 15-node network are presented. Figure 6.10(a) presents the sample network with 15 nodes and in Figure 6.10(b), the nodes and their corresponding MPR Selector Sets are listed. Nodes' MPR based weights can be calculated using the MPR Selector Set information given in Figure 6.10(b).

Figure 6.11(a) presents an histogram of the number of slots allocated to each node in Figure 6.10(a) during the simulations run on ns-2. Figure 6.11(b) shows the number of packets sent by each node at the link layer under a uniform traffic scenario. In the simulated traffic scenario the packet generation rate of each node is the same, equal to 700 bps, and every node generates traffic destined to every other node in the network.



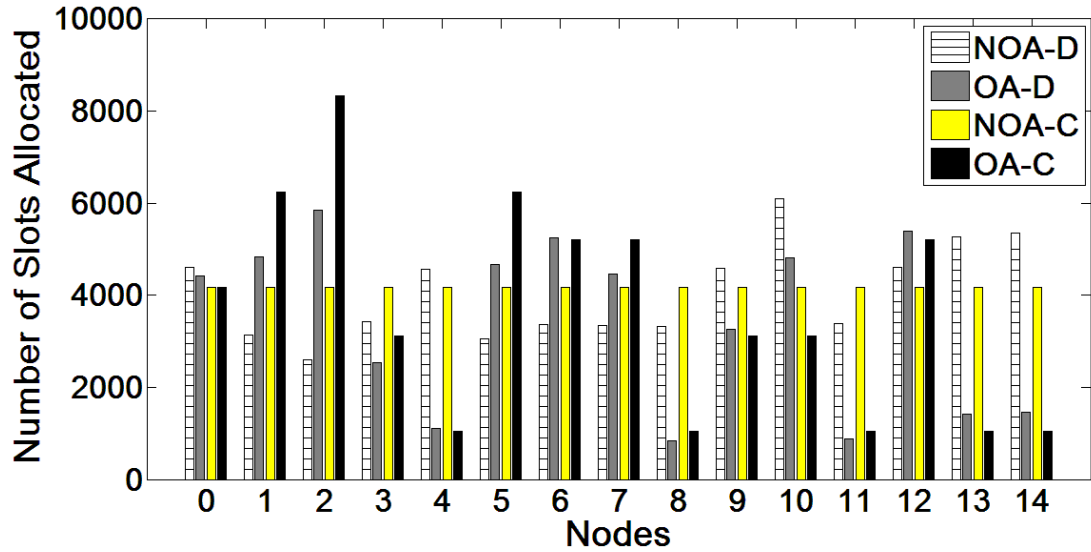


(a) Network topology - Topology (1).

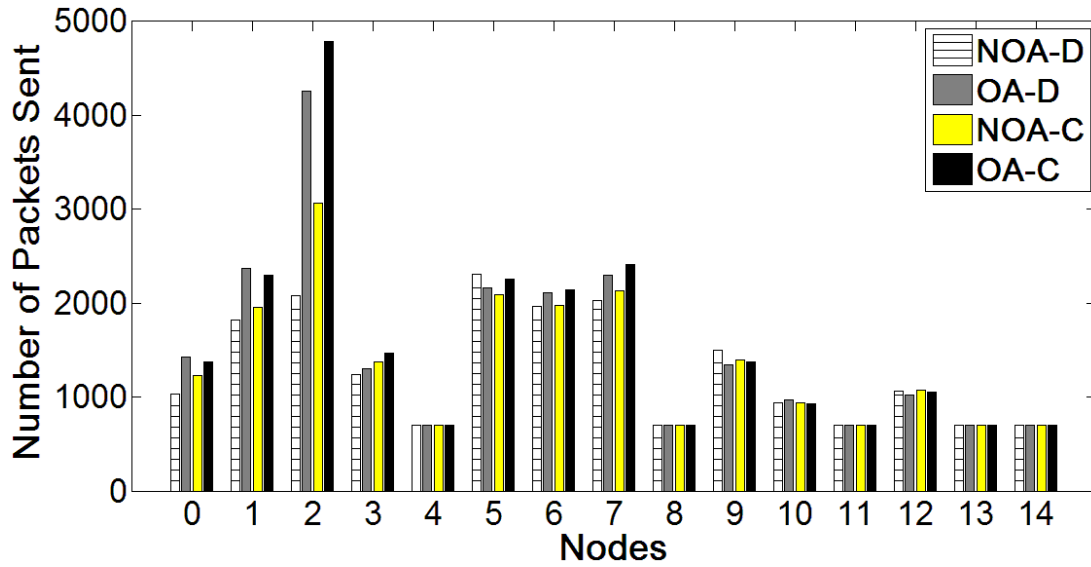
<i>Node</i>	<i>n0</i>	<i>n1</i>	<i>n2</i>	<i>n3</i>	<i>n4</i>	<i>n5</i>	<i>n6</i>	<i>n7</i>	<i>n8</i>	<i>n9</i>	<i>n10</i>	<i>n11</i>	<i>n12</i>	<i>n13</i>	<i>n14</i>
<i>MPR Selector</i>	1	0	1	4		1	2	2		10	0		6		
	4	2	3	11		2	11	8		5	9		7		
	10	3	5			3	12	12					13		
		4	6			8	13	14					14		
		5	7			9									
			8												
			11												

(b) MPR selector sets.

Figure 6.10: Sample 15-node network topology and MPR Selectors.



(a) Histogram showing the number of time slots allocated to each node.



(b) Histogram showing the number of packets sent by each node at the link layer.

Figure 6.11: Histograms showing the number of allocated time slots/sent packets on the sample 15-node network topology, Topology (1).

When NOA-C is used, since NOA-C is deterministic and non-weighted, every node gets exactly the same number of time slots. In NOA-C, every node is allocated a single time slot during a single scheduling cycle and by the end of the simulation, the number of time slots each node gets is equal to the number of scheduling cycles as observed in Figure 6.11(a).

On the other hand, the number of slots won by each node when OA-C is used is exactly proportional to its weight. Since OA-C is also a deterministic algorithm, in OA-C, each node gets exactly weight many time slots in a single scheduling cycle.

If we examine the number of slots each node wins when OA-D is used, it is observed that the number of slots won by each node is quite proportional to its weight, taking into account that the winner is selected pseudorandomly and using local information only, without having the complete view of the network. Pseudorandomness and the use of local information cause the behavior of OA-D algorithm to slightly deviate from the intended exact behavior.

In this particular example (Figure 6.10), the number of slots nodes win in OA-D are usually very close to those in OA-C. Nodes *node-1*, *node-2* and *node-5* are the nodes with the highest weights in the network and the number slots they win in OA-D can be less than number of nodes they win in OA-C. This is mainly due to the unnecessary refrainments caused by the locality of decisions.

On the other hand, the number of slots won by *node-10* requires a more detailed explanation for both of the distributed algorithms, OA-D and NOA-D. For OA-D, the number of slots won by *node-10* is quite high, considering that the weight of *node-10* is not very large. This is due to the fact that *node-10* is the only node that is not within the two-hop neighborhood of *node-2*, which is the node with the highest weight in the network. In addition, the size of its 2-hop neighborhood is not very large, either. These help *node-10* gain more slots than it is supposed to.

Likewise, for NOA-D, *node-10* is the node winning the highest number of slots in the network, while the number of slots nodes *node-13* and *node-14* are quite

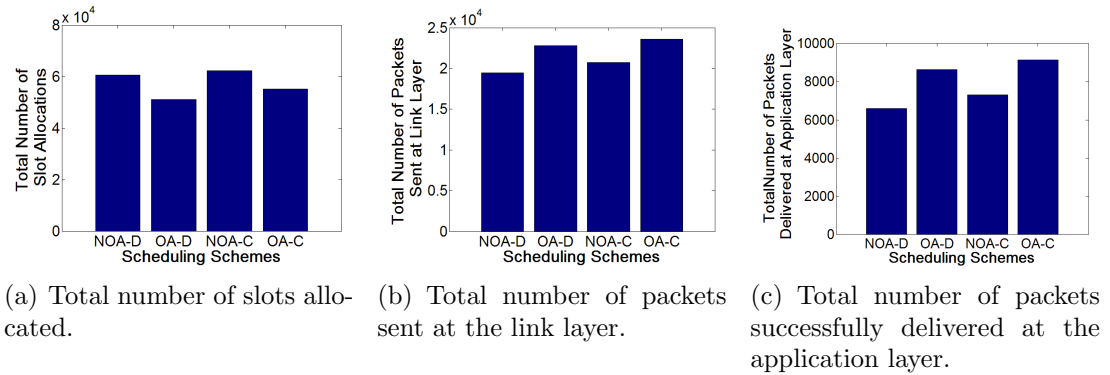


Figure 6.12: Total number of allocated slots/ transmitted packets and successfully delivered packets.

high as well. The sizes of *node-10*, *node-13* and *node-14*'s 2-hop neighborhoods are quite small, when compared against the other nodes' 2-hop neighborhood sizes. The opposite is also observed in the number of slots *node-2* wins, as *node-2* wins the lowest number of time slots among all nodes when NOA-D is used.

Figure 6.11(b) displays the number of packets each node in Topology (1) transmits at the link layer. In Topology (1), nodes *node-4*, *node-8*, *node-11*, *node-13* and *node-14* are Non-MPR nodes. Therefore, they only send the packets they have generated, which is equal under a uniform traffic scenario. The other nodes, which are MPR nodes, are liable to forward the traffic generated by the other nodes, which creates the difference in the number of packets each node sends.

Regarding the results presented in Figure 6.11(b), another point that is worth to mention is the following. In Figure 6.11(b), *node-2* is the node that sends the highest number of packets in all schemes, except NOA-D. Especially in OLSR-Aware schemes, OA-C and OA-D, the difference between the number of packets sent by *node-2* and the other nodes becomes even more apparent. In OLSR-Aware schemes, the number of slots allocated to *node-2* is proportional to its weight, that is, it is provided with more opportunities to transmit. Similarly, in NOA-D, examining Figure 6.11(a) and Figure 6.11(b) together, we observe that *node-2* fully utilizes the number of slots allocated to it, because the number of slots allocated to *node-2* is already very low.

We next discuss the results considering the total number of slots allocated (Figure 6.12(a)), the total number of packets transmitted at the link layer (Figure 6.12(b)) and the total number packets successfully delivered at the application layer (Figure 6.12(c)).

According to the results presented in Figure 6.12(a), the total number of allocated time slots is higher in Non-OLSR-Aware schemes than in OLSR-Aware schemes. In other words, NOA-C provides more concurrency than OA-C while NOA-D provides more concurrency than OA-D because the nodes are non-weighted in NOA-D and NOA-C. Therefore, they all have equal chances of winning. In other words, in Non-OLSR-Aware schemes, there are more occurrences of slot allocations in which nodes at different ends of the network topology are scheduled to transmit concurrently, without being suppressed by the nodes having large weights.

However, results in Figures 6.12(b) and 6.12(c) indicate that the total number of link layer transmissions and the total number of packets that are delivered successfully at the application layer are higher in OLSR-Aware schemes than in Non-OLSR-Aware schemes. This situation constitutes a sample case justifying our claim that increasing the total number of concurrent slot allocations at the link layer does not necessarily increase the end-to-end throughput at the application layer.

In Figure 6.13, we present the simulation results on the given 15-node network topology (Topology-1) under uniform simulation scenarios. Figure 6.13(a) shows the number of successfully packets while CBR rate is changed. The results presented here comply with our results shown in Figure 6.12(c) and Figure 6.3(a) and with our above-mentioned argument that increasing the number of concurrent allocations does not necessarily increase the throughput.

In Figures 6.13(b) and 6.13(c) the number of dropped packets and the packet delivery ratio are presented, respectively. The results presented in these two figures are complementary to those presented in Figure 6.13(a). Figure 6.13(d) depicts how the average end-to-end delay changes while CBR rate is changed. The results in Figure 6.13(d) comply in general with the results presented in Figure

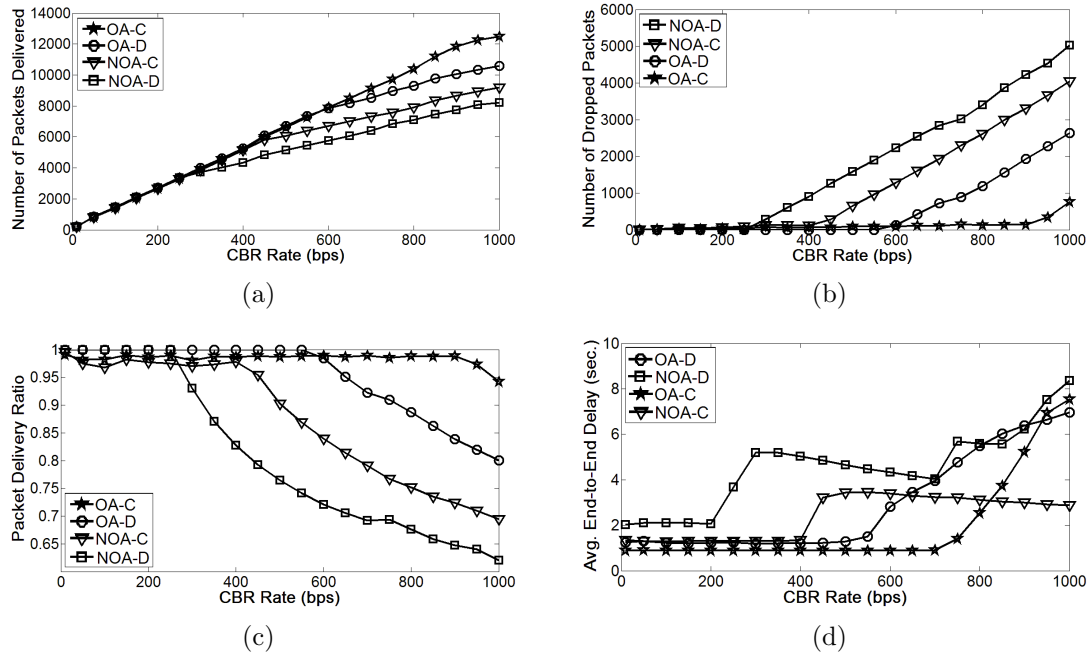


Figure 6.13: Uniform traffic simulation results on the sample 15-node network (Topology-1).

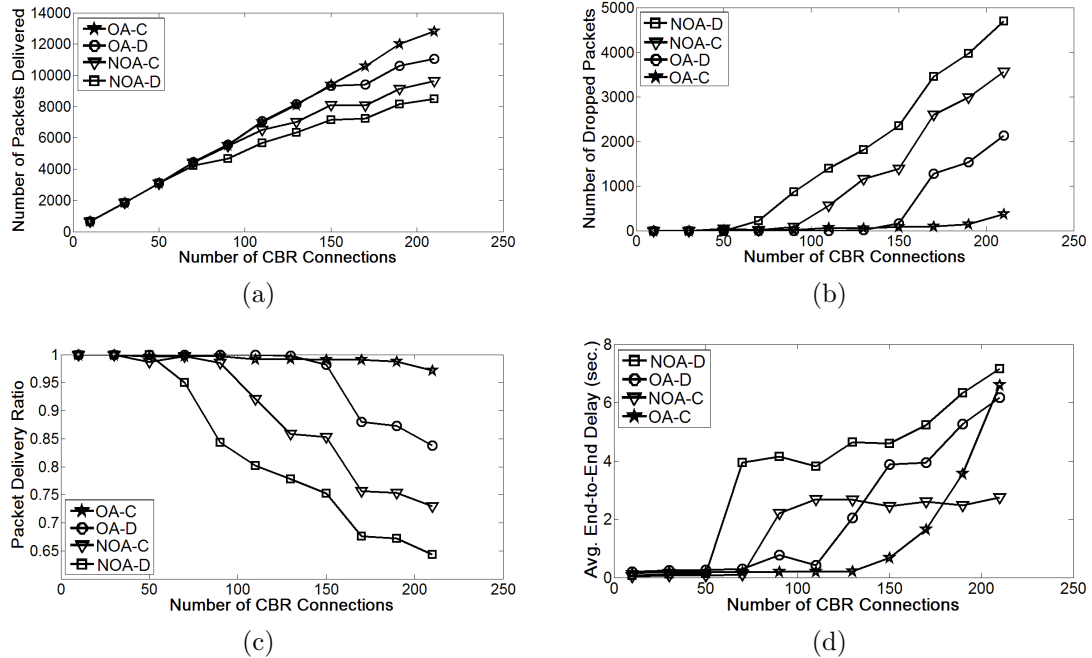


Figure 6.14: Nonuniform traffic simulation results on the sample 15-node network (Topology-1).

6.3(d) in the sense that Non-OLSR-Aware schemes exhibit a more conservative increase in the average end-to-end delay while the delay in OLSR-Aware schemes start to increase more steeply, especially when CBR rate is increased.

Finally, Figure 6.14 presents the simulation results on the given 15-node network topology (Topology-1) under nonuniform simulation scenarios. Therefore, neither the results in Figure 6.13 nor Figure 6.14 are averaged over a number of different topologies. In the simulated nonuniform traffic scenarios, CBR rate is fixed (1000 bps) and the number of CBR connections is changed between 10 to 210, increasing with a step size of 20 in each simulation scenario. Figure 6.14(a) shows the number of packets that are delivered successfully while CBR rate is changing. In Figures 6.14(b) and 6.14(c) the number of dropped packets and the packet delivery ratio are presented, respectively. Figure 6.14(d) depicts how the average end-to-end delay changes while CBR rate is changed.

# Chapter 7

## Conclusion

In this thesis, we propose a cross-layer weighting scheme, two routing layer (OLSR) aware STDMA based channel access schemes (one distributed, one centralized) and provide detailed performance comparisons for the scheduling schemes discussed throughout the paper.

Considering our simulation results, using MPR Based Weighting Scheme in the slot allocation procedure improves the total number of packets that can be delivered successfully at the application layer, that is, the total application throughput. To be able to improve the overall application throughput in a multi-hop wireless mesh network, the nodes that forward data should be acknowledged and given more transmission opportunities to alleviate their burden. In OLSR, this can be done by giving more weight (priority) to MPR nodes. In other words, MPR information provides a good approximation for the expected traffic and granting more transmission opportunities to MPR nodes improves the overall performance of the network as presented in our simulation results in Chapter 6.

Many studies in the literature focus on maximizing the number of concurrent allocations at the link layer. However, maximizing application throughput is harder, usually requiring dynamic calculations to get involved. Therefore, simpler methods such as handling the problem of maximizing the application layer throughput via the maximization of concurrent allocations at the link layer are



preferred very commonly. Nonetheless, the achieved level of concurrency is not the only metric that affects the overall application throughput. In our simulation results, we show that maximizing the number of concurrent slot allocations at the link layer does not necessarily increase the overall application layer throughput.

In addition, our simulation results also confirm that it is possible to achieve more concurrency with a deterministic-centralized algorithm than it is possible by using a pseudorandom-distributed algorithm. In our simulations, the average concurrency levels achieved by the centralized schemes improve over the distributed schemes' concurrency levels approximately by 15% in OLSR-Aware schemes and 20% in Non-OLSR-Aware schemes.

Another very important aspect of our work is its being a cross-layer study which is able to improve the application throughput by only using a static network layer parameter (i.e. MPR information in OLSR) in a very simple way. Since we perform cross-layer scheduling only by making use of a static parameter which is disseminated at no extra overhead, in terms of the implementation complexity, our work becomes advantageous over other cross-layer works in the literature that use dynamic calculations.

As one last point, the results reported in this thesis can be considered as open for further research. For instance, the integration of MPR Based Weighting Scheme into the link level and/or packet level scheduling or comparing the performances of OLSR-Aware node and link level channel access scheduling schemes under different traffic distributions and loads are among the possible directions that can be further studied. Since MPR information remains stable in a static network once it converges, the scheduling schemes proposed in this thesis (OA-C and OA-D) can be considered as static scheduling schemes. Therefore, another interesting research direction might be looking into possible ways of integrating MPR or some other OLSR-specific information into a dynamic scheduling scheme and providing detailed performance comparisons.

# Bibliography

- [1] IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems. *IEEE Std 802.16-2004*, pages 1–895, 2004.
- [2] C. Adjih, E. Baccelli, T. Clausen, P. Jacquet, and G. Rodolakis. Fish Eye OLSR Scaling Properties. *IEEE Journal of Communication and Networks (JCN), Special Issue on Mobile Ad Hoc Wireless Networks*, 6(4):343–351, 2004.
- [3] A. Akram, S. Pervez, and S. Khan. Multimedia Task Scheduling on OLSR Enabled Ad-Hoc Networks. *Proceedings of the 5th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications*, pages 40–44, 2006.
- [4] I. Akyildiz, X. Wang, and W. Wang. Wireless Mesh Networks: A Survey. *Computer Networks*, 47(4):445–487, 2005.
- [5] H. Al-Omari and K. E. Sabri. New Graph Coloring Algorithms. *American Journal of Mathematics and Statistics*, 2(4):739–741, 2006.
- [6] G. Athanasiou, T. Korakis, O. Ercetin, and L. Tassiulas. A Cross-Layer Framework for Association Control in Wireless Mesh Networks. *IEEE Transactions on Mobile Computing*, 8(1):65–80, 2009.
- [7] L. Bao and J. Garcia-Luna-Aceves. A New Approach to Channel Access Scheduling for Ad Hoc Networks. *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 210–221, 2001.

- [8] A. BR, L. Reddy, and P. Hiremath. Performance Comparison of Wireless Mobile Ad-Hoc Network Routing Protocols. *IJCSNS*, 8(6):337, 2008.
- [9] A. Busson, N. Mitton, and E. Fleury. An Analysis of the Multi-Point Relays Selection in OLSR (RR-5468). Technical report, Institut National De Recherche En Informatique Et En Automatique, 2005.
- [10] Y. Cao, Z. Liu, and Y. Yang. A Centralized Scheduling Algorithm Based on Multi-Path Routing in WiMAX Mesh Network. *International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM)*, pages 1–4, 2006.
- [11] J. Chen, Y. Lee, D. Maniezzo, and M. Gerla. Performance Comparison of AODV and OLSR in Wireless Mesh Networks. *Proceedings of the IFIP Med-Hoc-Net Conference*, pages 14–17, 2006.
- [12] S. Cheng, P. Lin, D. Huang, and S. Yang. A Study on Distributed/Centralized Scheduling for Wireless Mesh Network. *International Conference On Communications And Mobile Computing*, pages 599–604, 2006.
- [13] T. Clausen and P. Jacquet. Optimized Link State Routing Protocol (OLSR): RFC 3626. *IETF Internet Draft*, <http://www.ietf.org/rfc/rfc3626.txt>, 2003.
- [14] P. Djukic and S. Valaee. Centralized Scheduling Algorithms for 802.16 Mesh Networks. *WiMax/MobileFi: Advanced Research and Technology*, 2007.
- [15] Y. Du, Y. Bao, and J. J. Garcia-Luna-Aceves. A History-based Scheduling Protocol for Ad-Hoc Networks. *Proceedings of the 12th International Conference on Computer Communications and Networks*, pages 223–228, 2003.
- [16] A. Ephremides and T. Truong. Scheduling Broadcasts in Multihop Radio Networks. *IEEE Transactions on Communications*, 38(4):456–460, 1990.
- [17] S. Ergen and P. Varaiya. TDMA Scheduling Algorithms for Sensor Networks. Technical report, EECS Department, Berkeley University, CA, 2005.

- [18] E. Ermel and P. Muhlethaler. Using OLSR Multipoint Relays (MPRs) to Estimate Node Positions in a Wireless Mesh Network. Technical report, Institut National De Recherche En Informatique Et En Automatique, 2006.
- [19] K. Fall, K. Varadhan, et al. The ns-2 Manual. *the VINT Project*, <http://www.isi.edu/nsnam/ns/doc>, 2008.
- [20] Y. Gan, S. Masson, G. Guibe, B. Marin, and C. Le Martret. Cross-Layer Optimization of OLSR with a Clustered MAC. *Military Communications Conference (MILCOM)*, pages 1–7, 2006.
- [21] Hipercom. OLSR Routing Protocol. <http://hipercom.inria.fr/olsr/#code>, 2005.
- [22] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized Link State Routing Protocol for Ad-Hoc Networks. *Proceedings of IEEE International Multi Topic Conference (INMIC)*, pages 62–68, 2001.
- [23] K. Kowalik, B. Keegan, and M. Davis. Making OLSR Aware of Resources. *Wireless Communications, Networking and Mobile Computing Conference (WiCom)*, pages 1488–1493, 2007.
- [24] M. Kuran, G. Gur, T. Tugcu, and F. Alagoz. Cross-Layer Routing-Scheduling in IEEE 802.16 Mesh Networks. *Proceedings of the 1st International Conference on MOBILE Wireless MiddleWARE, Operating Systems, and Applications*, 2008.
- [25] J. Leung. *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman & Hall/CRC, 2004.
- [26] E. Lloyd. Broadcast Scheduling for TDMA in Wireless Multihop Networks. *Handbook of Wireless Networks and Mobile Computing*, pages 347–370, 2002.
- [27] M. Macedo, A. Grilo, and M. Nunes. Distributed Latency-Energy Minimization and Interference Avoidance in TDMA Wireless Sensor Networks. *Computer Networks*, 53(5):569–582, 2009.

- [28] E. Malaguti. The Vertex Coloring Problem and its Generalizations. *4OR: A Quarterly Journal of Operations Research*, pages 1–4.
- [29] S. Mecke. MAC Layer and Coloring. *Lecture Notes in Computer Science*, 4621:63, 2007.
- [30] J. Moy. OSPF Version 2 (RFC 2328). *IETF Internet Draft*, <http://www.ietf.org/rfc/rfc2328.txt>, 1998.
- [31] C. Ngo and V. Li. Centralized Broadcast Scheduling in Packet Radio Networks via Genetic-Fix Algorithms. *IEEE Transactions on Communications*, 51(9):1439–1441, 2003.
- [32] D. Nguyen and P. Minet. Interference Effects on the OLSR Protocol: ns-2 Simulation Results. *Proceedings of the IFIP Med-Hoc-Net Conference*, 2004.
- [33] D. Nguyen and P. Minet. Interference-Aware QoS OLSR for Mobile Ad-Hoc Network Routing. *Proceedings of SNPD/SAWN*, 5:428–435, 2005.
- [34] D. Nguyen and P. Minet. Scalability of OLSR Protocol with the Fish Eye Extension. *Sixth International Conference on Networking (ICN)*, page 88, 2007.
- [35] R. Pal. Alluvion and Cascade: Fast Data Dissemination Schemes in Multihop Wireless Networks. *Proceedings of the 1st International Workshop on Decentralized Resource Sharing in Mobile Computing and Networking: MobiShare*, pages 66–68, 2006.
- [36] G. Pei, M. Gerla, and T. Chen. Fisheye State Routing: A Routing Scheme for Ad-Hoc Wireless Networks. *IEEE International Conference on Communications*, 2000.
- [37] M. Peng, Y. Wang, and W. Wang. Cross-layer Design for Tree-type Routing and Level-Based Centralized Scheduling in IEEE 802.16 Based Wireless Mesh Networks. *Communications, IET*, 1(5):999–1006, 2007.
- [38] V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks. *ACM SenSys*, pages 181–192, 2003.

- [39] S. Ramanathan. A Unified Framework and Algorithm for Channel Assignment in Wireless Networks. *Wireless Networks*, 5(2):81–94, 1999.
- [40] S. Ramanathan and E. L. Lloyd. Scheduling Algorithms for Multihop Radio Networks. *IEEE/ACM Transactions on Networking*, 1(2):166–177, 1993.
- [41] Z. Ren, Y. Zhou, and W. Guo. An Adaptive Multichannel OLSR Routing Protocol Based on Topology Maintenance. *IEEE International Conference on Mechatronics and Automation*, 4:2222–2227, 2005.
- [42] I. Rhee, A. Warriier, M. Aia, and J. Min. Z-MAC: a Hybrid MAC for Wireless Sensor Networks. *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pages 90–101, 2005.
- [43] I. Rhee, A. Warriier, J. Min, and L. Xu. DRAND: Distributed Randomized TDMA Scheduling for Wireless Ad-hoc Networks. *Proceedings of the 7th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 190–201, 2006.
- [44] I. Rhee, A. Warriier, and L. Xu. Randomized Dining Philosophers to TDMA Scheduling in Wireless Sensor Networks. Technical report, 2004.
- [45] F. J. Ros. UM-OLSR Implementation (Version 0.8.8) for ns-2. *UM-OLSR Project*, <http://masimum.inf.um.es/?Software:UM-OLSR>, 2009.
- [46] S. Shakkottai, T. Rappaport, and P. Karlsson. Cross-Layer Design for Wireless Networks. *IEEE Communications Magazine*, 41(10):74–80, 2003.
- [47] A. Tonnesen. Olsrd: An Ad-Hoc Wireless Mesh Routing Daemon. <http://www.olsr.org>, 2008.
- [48] L. Viennot. Complexity Results on Election of Multipoint Relays in Wireless Networks. Technical report, Institut National de Recherche en Informatique et en Automatique, 1998.
- [49] K. Wang, M. Peng, and W. Wang. Distributed Scheduling Based on Polling Policy with Maximal Spatial Reuse in Multi-hop WMNs. *The Journal of China Universities of Posts and Telecommunications*, 14(3):22–27, 2007.

- [50] Y. Wang and I. Henning. A Deterministic Distributed TDMA Scheduling Algorithm for Wireless Sensor Networks. *Wireless Communications, Networking and Mobile Computing Conference (WiCom)*, pages 2759–2762, 2007.
- [51] H. Wei, S. Ganguly, R. Izmailov, and Z. Haas. Interference-Aware IEEE 802.16 WiMAX Mesh Networks. *IEEE Vehicular Technology Conference, (VTC-Spring)*, 2005.
- [52] K. Xu, L. Qi, and Y. Shu. Enhancing TCP Fairness in Ad-Hoc Wireless Networks Using Neighborhood RED. *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 16–28, 2003.
- [53] S. Xu and T. Saadawi. Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad-Hoc Networks? *IEEE Communications Magazine*, 39(6):130–137, 2001.
- [54] J. Yeo, H. Lee, and S. Kim. An Efficient Broadcast Scheduling Algorithm for TDMA Ad-Hoc Networks. *Computers and Operations Research*, 29(13):1793–1806, 2002.
- [55] S. Zhao, Z. Wu, A. Acharya, and D. Raychaudhuri. PARMA: a PHY/MAC Aware Routing Metric for Ad-Hoc Wireless Networks with Multi-Rate Radios. *Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, pages 286–292, 2005.