University of Alberta

Non-bifurcated Routing and Scheduling in Wireless Mesh Networks

by

Abdullah-Al Mahmood

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

Examining Committee

Ehab Elmallah, Computing Science

Mike MacGregor, Computing Science

Guohui Lin, Computing Science

Masoud Ardakani, Electrical and Computer Engineering

Damla Turgut, Electrical Engineering and Computer Science, University of Central Florida

Abstract

Multi-hop wireless mesh networks (WMNs) provide a cost-effective means to enable broadband wireless access (BWA) services to end users. Such WMNs are required to support different classes of traffic where each class requires certain quality of service (QoS) levels. The research direction undertaken in this thesis considers the development of enhanced routing and scheduling algorithms that enable WMNs to support various QoS metrics for the served traffic.

A fundamental class of routing problems in WMNs asks whether a given end-to-end flow that requires certain bandwidth, and benefits from routing over a single path (also called non-bifurcated routing), can be routed given that some ongoing flows are being served in the network. In the thesis, we focus on the development of combinatorial algorithms for solving such *incremental non-bifurcated* problems for two types of WMNs:

1. WMNs where mesh routers use contention-based protocol for medium access control (MAC), and

2. WMNs where mesh routers use time division multiple access (TDMA) for MAC.

For WMNs employing contention-based MAC protocols, we present a novel non-bifurcated routing algorithm that employs techniques from the theory of network flows. The main ingredient in our algorithm is a method for computing interference-constrained flow augmenting paths for routing subscriber demands in the network.

For WMNs employing TDMA, we develop a number of joint routing and scheduling algorithms, and investigate the use of such algorithms to maximize

the number of served flows. In chapter 4, we consider a throughput maximization problem in the well-known class of grid WMNs. We present an iterative algorithm that strives to achieve high throughput by considering routing and scheduling a pair of distinct flows simultaneously to the gateway in each iteration.

In chapter 5, we explore joint routing and scheduling in TDMA-based WMNs with arbitrary topologies, and devise an algorithm that can deal with arbitrary interference relations among pairs of transmission links. In particular, our devised algorithm solves a generalized problem where a cost value is associated with using any possible time-slot on any transmission link, and a minimum cost route is sought along which a new flow can be scheduled without perturbing existing slot assignments.

Acknowledgements

I am grateful to Dr. Ehab Elmallah for his guidance and support during the course of my doctoral program. I would like to thank the members of the examining committee Dr. Mike MacGregor, Dr. Guohui Lin, Dr. Masoud Ardakani, and Dr. Damla Turgut for their valuable feedback. I also thank NSERC for providing support for the research work presented in the thesis.

Contents

List of Figures

List of Tables

List of Acronyms

ACK	Acknowledgment
AP	Access Point
AODV	Ad hoc On-demand Distance Vector
BE	Best Effort (service)
BFS	Breadth First Search
BS	Base Station
BWA	Broadband Wireless Access
CAN	Configurable Access Network
CETT	Cumulative Expected Transmission Time
CGF	Closest to Gateway First
CID	Connection Identifier
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTS	Clear To Send
DCF	Distributed Coordination Function
DSR	Dynamic Source Routing
DSSS	Direct Sequence Spread Spectrum
EDCA	Enhanced Distributed Channel Access
ETT	Expected Transmission Time
ETX	Expected Transmission Count
FAP	Flow Augmentation Path
FDD	Frequency Division Duplexing
FHSS	Frequency Hopping Spread Spectrum
FPRS	Flow Pair Routing and Scheduling
GPS	Generalized Processor Sharing
HCCA	HCF Controlled Channel Access
HCF	Hybrid Coordination Function
HWMP	Hybrid Wireless Mesh Protocol
IC-FAP	Interference Constrained Flow Augmenting Path
IR	Infrared
ISM	Industrial, Scientific, and Medical
LAN	Local Area Network
LOS	Line of Sight
LTE	Long Term Evolution

MAC	Medium Access Control
MAN	Metropolitan Area Network
MC-SFRS	Minimum Cost Single Flow Routing and Scheduling
MID	Maximum Interference Distance
MIMO	Multiple Input Multiple Output
MR-LQSR	Multi-Radio Link Quality Source Routing
MUF	Most Utilized First
NBR	Non-bifurcated Routing
NIC	Network Interface Card
NOC	Network Operation Center
nrtPS	Non-real-time Polling Service
OFDM	Orthogonal Frequency Division Multiplexing
PCF	Point Coordination Function
PHY	Physical Layer
PMP	Point-to-multipoint
QoS	Quality of Service
rtPS	Real-time Polling Service
RTS	Request To Send
RTT	Round-trip Time
SC	Single Carrier
SFID	Service Flow Identifier
SFRS	Single Flow Routing and Scheduling
SFS	Single Flow Scheduling
SLR	(Near) Straight Line Routing
SS	Subscriber Station
TCP	Transmission Control Protocol
TDD	Time Division Duplexing
TDMA	Time Division Multiple Access
UDP	User Datagram Protocol
UGS	Unsolicited Grant Service
WCETT	Weighted Cumulative Expected Transmission Time
WGMN	Wireless Grid Mesh Network
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WMN	Wireless Mesh Network
WPAN	Wireless Personal Area Network

Chapter 1

Introduction

Wireless mesh networks (WMNs) can provide broadband wireless access (BWA) service across a diverse geographic region. Effective delivery of such service hinges on the efficient management of wireless resources. The general research direction in this thesis concerns the development and analysis of routing and scheduling algorithms for resource management to support various quality of service (QoS) metrics for traffic flows in WMNs. In this chapter, we give a general overview of WMNs, motivate the research direction undertaken in this thesis, and present an overview of our contributions in the thesis.

1.1 Introduction

In this section, we introduce *multi-hop* wireless networks as a means to provide broadband wireless access (BWA) to subscribers. We discuss *wireless mesh networks* (WMNs) as a special class of multi-hop wireless networks intended to provide BWA. We motivate routing and scheduling methods as network design tools for providing needed quality of service (QoS) measures to flows in such networks.

BWA networks are envisioned to be useful in different deployment scenarios like last mile connection and backhauling traffic from wireless hotspots to distant premises. Health and medical systems, transportation systems, and security systems can also benefit from such networks [1]. BWA networks are expected to support data rates above 100 Mbps with QoS guarantees. Fading of wireless signals over large distances, and interference in the shared wireless medium make the realization of BWA challenging. WMNs form a class of multi-hop wireless network intended to provide BWA over an extended geographic area. To enable BWA, WMNs typically employ routers that have adequate power source and computational capability, and are connected to specialized antennas for communication over relatively large distance. It is also typical that the entire network operates under the control of a single administration domain. Multi-hop wireless networks can be realized using different communication devices that are intended for building wireless personal area networks (WPANs), wireless local area networks (WLANs), or wireless metropolitan area network (WMANs). Among the above technologies WMAN and WLAN technologies are discussed in the literature as enabling technologies for designing WMNs.

Ad-hoc networks and WMNs constructed using WLAN technologies have some similarities and some differences. Some of the notable differences are summarized below.

- *Infrastructure:* Ad-hoc networks do not require any infrastructure. WMNs on the other hand rely on infrastructure deployment for delivery of service.

- *Computational capability and power constraint:* Mesh routers are much more powerful in terms of computational resources than the nodes in ad-hoc networks. Also, nodes in ad-hoc networks are more constrained in power than mesh routers.

- *Communication capability:* An ad-hoc network usually operates on a single radio channel and suffer from significant amount of contention for accessing the wireless medium. In order to achieve better performance wireless mesh networks can use a number of orthogonal channels. There is no interference between transmissions on two orthogonal channels when the corresponding transmission areas overlap.

- *Service requirement:* The performance requirements in terms of data delivery rate and routing capabilities are much higher for the mesh routers than for the nodes in an ad-hoc network.

- *Communication hierarchy:* Ad-hoc networks operate in a peer-to-peer mode. Traffic management is done in a distributed fashion in these networks. Most WMNs on the other hand are deployed in a two tier architecture. The first tier is composed of mesh routers, and the second tier is composed of mesh clients. Both centralized and distributed mechanisms may be required to manage inter-router and router-client communication in these WMNs.

We now describe some architectures proposed in the literature for WMNs. From a conceptual point of view, Akyildiz *et al.* categorize mesh network architectures into three classes [1]:

- *Infrastructure networks*: In an infrastructure network, a number of fixed wireless routers form the mesh backbone. The backbone network provides clients access to outside networks through gateways (*i.e.*, mesh routers with bridging capabilities). Clients connect to the nearest mesh router by single wired or wireless link, and communicate to other clients or resources beyond the mesh network through that router. Figure 1.1 shows an example of an infrastructure network. Many research work assume this architecture in the system model. We also consider this architecture in our contributions mentioned in this thesis.

- *Client mesh networks*: In a client WMN structure, the clients themselves act as routers, and form an operational ad-hoc network. However, the traffic demands and routing requirements call for more computational resource in the mobile clients (routers) than mobile nodes in regular ad-hoc networks.

- *Hybrid mesh networks*: In a hybrid WMN, both the backbone network and the client mesh network operate simultaneously. The presence of mobile routers (*i.e.*, clients) in addition to fixed routers (*i.e.*, mesh routers)

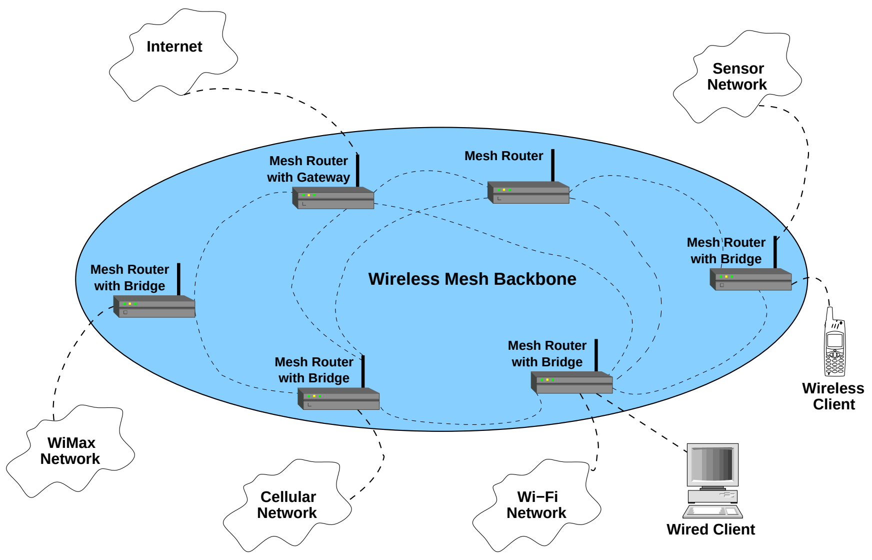


Figure 1.1: An Infrastructure WMN

makes resource management in this type of WMN more complex than either of the other architectures. Figure 1.2 shows an example of hybrid wireless mesh network.

Since WMNs are intended to provide BWA to end-users, the issue of supporting delay-jitter sensitive traffic like streaming traffic, and delay-sensitive traffic like real-time traffic becomes important. Support of such traffic requires QoS assurance. The area of supporting QoS metrics in both wired and wireless networks has received substantial attention in the literature (*e.g.*, [13], [26], [33], [60], [63] and many references therein). Broadly speaking, methods for designing networks that can guarantee QoS metrics include the following (see *e.g.*, the survey in [57]):

- **Constrained routing:** The routing problem is formulated by including the constraints involving QoS metrics on the carried flows as well as constraints arising from the network (*e.g.*, [2], [26]). In multi-hop wireless networks the use of contention-based MAC and time division multiple access (TDMA) MAC impose additional constraints on routing algorithms. In particular in networks employing TDMA MAC we encounter joint routing and scheduling problems. The importance of

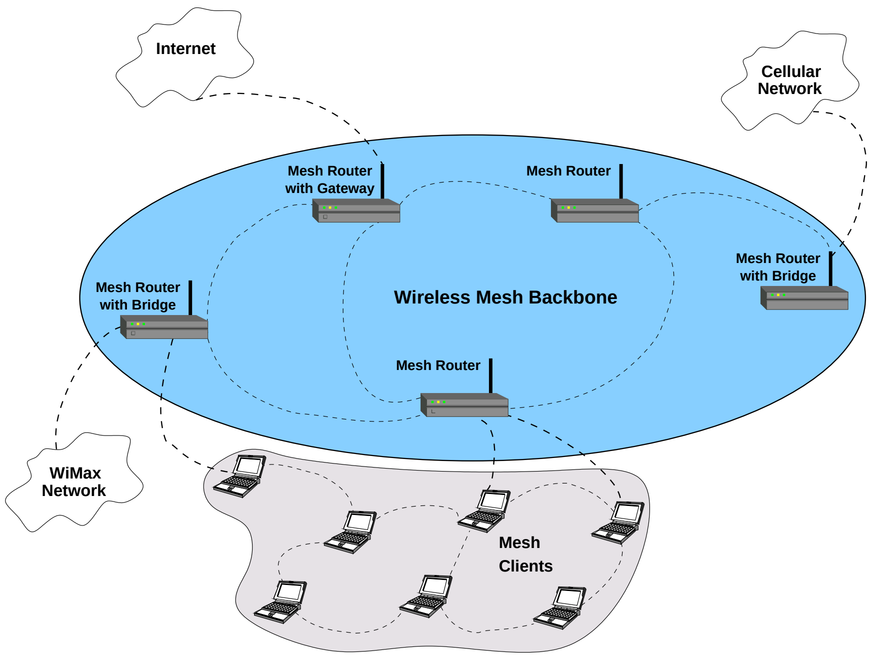


Figure 1.2: A Hybrid WMN

routing decisions motivates research in this direction.

- **Admission control:** Depending on available resources and QoS requirements of new and existing flows, an admission control algorithm decides to accept or reject serving new flow requests (*e.g.*, [32], [62]). A general goal of this design method is to minimize resource allocation while attempting to maximize the number of admitted flows.

- **Traffic shaping:** Traffic shaping algorithms handle bursts of network traffic by buffering techniques, and controlling the flow rate at a sustainable level while catering to QoS requirements of the flows (*e.g.*, [49], [59]).

- **Bandwidth allocation:** During the lifetime of a long-lived flow, a router can allocate different bandwidths to the flow during different intervals, so as to optimize QoS of all flows passing through the router. The resulting class of problems is called bandwidth allocation problems for which there are many research work in literature (*e.g.*, [52], [55]).

- **Packet scheduling based on generalized processor sharing (GPS) model:** The algorithms in this category manage packet queues and decide on the order of outgoing packets so that the QoS requirements of the flows are met (*e.g.*, [4], [8], [44], [61]).

Of the above design methods routing in WMNs (as well as ad-hoc networks and wireless sensor networks) have received attention in literature [2, 3, 7, 10, 11, 19, 26, 27, 31, 33, 47, 51, 56] since

a) routing is a core function of the network,

b) routing has impact on network capacity (*e.g.*, as discussed in sections 2.4.1 and 2.4.4), and

c) routing impacts network delays when considered with other factors, *e.g.*, nodal processing delays, MAC delay, scheduling, bandwidth allocation, and/or call admission control.

In this thesis we focus on routing and scheduling algorithms intended to provide QoS assurance to traffic in WMNs enabling BWA. In the following section we introduce and motivate the concept of *incremental non-bifurcated routing* that is used throughout the thesis.

1.2 Thesis Direction and Motivation

In this thesis we consider routing *end-to-end* flows in a WMN that may utilize a CSMA/CA or a TDMA MAC protocol. Each flow may be an aggregate of one or more TCP (transmission control protocol) or UDP (user datagram protocol) streams from a mesh router connecting one or more subscriber stations to either another mesh router in the WMN, or to a gateway that forwards the flow to the Internet. Multicast flows are not considered explicitly in the thesis. At any instant of time, some streams in a flow may terminate, and other streams may arrive. Thus, *incremental* decrease and increase of the bandwidth allocated to the flow occurs over its lifetime.

One important aspect of the thesis is the consideration of incremental routing problems where the state of the network is induced by existing flows, and possibly interference from other networks (*e.g.*, in cases where our target network operates in the unlicensed ISM band). For this class of incremental routing problems of end-to-end flows, our objective is to devise methods for allocating bandwidth to new flows in a controlled manner so as to enable providing desired QoS levels to the flows (*e.g.*, minimum required bandwidth and maximum tolerable delay jitter). Thus, the obtained algorithms are intended to provide traffic engineering tools for WMNs. To achieve the above objective of controlled bandwidth allocation, the devised algorithms need to deal with contention aspects in CSMA/CA-based WMNs, and joint routing and scheduling (*i.e.*, assigning time-slots to links along a path from the source node to the destination node to carry a flow) in TDMA-based WMNs. In both cases, interference aspects have to be suitably modeled.

Another important aspect in formalizing the routing problems in the thesis is the fact that the number of paths used to route a flow has an impact on the capacity of the network, and QoS levels achieved for the flow. Many algorithms employing different notions of path set selection to route a flow have been discussed in the literature for multi-hop wireless networks. In this regard, the following limiting cases in the spectrum of notions can be identified:

– multi-path routing where the number of paths used to route a flow is potentially unconstrained, and

– single path (also called *non-bifurcated*) routing where only a single path is used.

Note that in either case a path may change over time in reaction to changes that may occur in the network (*e.g.*, changes due to congestion buildup). Examples of situations where increasing the number of paths to route a flow is useful include:

a) Situations where the nodes of the network are resource constrained (*e.g.*, constrained in energy). In such situations, utilizing multiple paths can provide load sharing among the nodes.

b) Situations where node or link failures are frequent, or path disconnections are frequent (*e.g.*, due to mobility). In such situations, utilizing multiple paths can provide higher end-to-end reliability.

c) Either connection-oriented transport layer service is not needed by the streams in a flow, or the service is needed and auxiliary methods are used to provide support over multiple paths (*e.g.*, source coding).

d) The capacity of a network is being analyzed. Here, using multi-path routing allows subdividing the data rate required by a flow into infinitesimally small data rates for routing the sub-flows obtained by splitting the original flow over multiple paths.

For networks where the above factors are not critical, the use of non-bifurcated routing (or by extension, routing over a few paths) offers many advantages. This method of routing is the norm when using many distributed and centralized algorithms (*e.g.*, [3, 25, 46]). Examples of performance gains obtained by utilizing a non-bifurcated routing (or routing over a few paths) include the following.

- *Performance Gains at the MAC Layer*: For WMNs employing CSMA/CA MAC protocols, reducing the number of routes used to route a group of end-to-end flows helps in reducing the contention in delivering the carried packets at junction nodes where two or more routes meet. In the limiting case where all packets are carried over one route, packets that belong to the same flow do not contend with each other.

  For WMNs employing TDMA protocols, reducing the number of routes reduces the control messages needed to allocate and release bandwidth along the routes, simplifies managing the data structures containing the slots to flows assignment, and enables the allocation of multiple consecutive slots to serve a given end-to-end flow.

- *Improved Controllability of QoS*: Single path routing gives us the ability to provide homogeneous treatment (*e.g.*, through application of the same scheduling algorithm) of packets belonging to a flow at intermediate nodes along the route, and simplifies the forwarding operation. Thus we have better control over maintaining the QoS levels for end-to-end flows. Devising single path routing algorithms for WMNs has been mentioned in [27] as a way of combating out-of-order packet delivery problem and loss recovery problems associated with multi-path routing.

- *Performance Gains at the Transport Layer*: The support for TCP traffic plays a key role in profitable deployment of WMNs. The throughput of conventional TCP (*e.g.*, TCP Reno) suffers when the RTT (round-trip time) and consequently the protocol's timeout interval are either overestimated or underestimated. In particular, premature timeouts cause TCP congestion control mechanisms to decrease the protocol's window size, and unnecessarily long timeouts make the protocol react slowly to packet losses. Thus the achieved throughput of TCP over a set of paths is adversely affected by the path with the largest delay, and the delay variance of the ensemble. In this thesis, we pursue the direction of developing algorithms for routing a flow with a given bandwidth requirement indivisibly along a path. Such a path has the potential of achieving low round-trip delay and delay variance.

  Recently a number of modifications to traditional TCP have been proposed to improve TCP performance in wireless networks. The survey of [53] discusses a number of such proposals (*e.g.*, I-TCP, TCP SACK, TCP Veno, and TCP Jersey). Examples of situations where improvements have been attained by the proposed wireless TCP architectures include the following:

  – TCP over a heterogeneous path composed of a wired segment and a wireless segment (*e.g.*, Internet-WLAN connections) where there is a need to differentiate between packet losses due to congestion in the wired network, and random errors in the wireless segment.

  – TCP over frequently disconnected routes (*e.g.*, ad-hoc networks).

  - TCP over long delay channels (*e.g.*, satellite connections).

  We remark that: (a) wireless TCP architectures are not intended to improve performance when using an ensemble of paths with large delay and delay jitter, and (b) similar to the argument presented above with traditional TCP, wireless TCP architectures benefit from careful allocation of paths to flows.

  We also note that research work on performance of TCP over multi-hop wireless networks (*e.g.*, [54]) has identified a number of recommendations to improve TCP performance. One of the important recommendations is to reduce contention between TCP data and TCP ACK packets. The work done in the thesis takes such recommendation into consideration by allocating sufficient bandwidth to *bidirectional* flows for carrying data packets in one direction, and ACK packets in the opposite direction between the two end nodes of the flow.

- *Improved Network Management*: Single path routing facilitates network monitoring and management since it is easier to isolate the source of any malfunction.

The above practical issues motivate research on the fundamental class of non-bifurcated routing problems where sufficient bandwidth should be allocated to each served flow. Yet another scientific motivation comes from the need to understand the combinatorial aspects of routing flows in an environment influenced by interferences where even the transmissions between links on the same route can adversely affect each other.

We next remark that the concept of non-bifurcated routing is parameterized by the magnitude of the unit of flows to be routed indivisibly. The flow unit value parameter then determines the granularity level of the implementation. By decreasing such granularity level, an iterated use of a non-bifurcated routing algorithm degenerates to a conventional multi-path routing algorithm that can achieve high bandwidth utilization at the expense of sacrificing the advantages mentioned above. In contrast, by increasing the granularity level, all packets in a flow can be routed indivisibly over a single path at the expense of lowering bandwidth utilization. In this thesis we are not concerned with determining a suitable granularity level for a given distribution of flow rates, rather we focus on developing mechanisms for non-bifurcated routing for any given level of granularity.

The above factors motivate the research work done in the thesis on the class of **incremental non-bifurcated** routing of end-to-end flows in WMNs. In the next two sections we present an overview of the IEEE 802.16 as an example of WMN architectures that use TDMA MAC protocol, and the IEEE 802.11 as an example of WMN architectures that use CSMA/CA MAC protocol.

1.3 Overview of the IEEE 802.16 Standards

The IEEE 802.16 family of standards, also known as the WirelessMAN (Wireless Metropolitan Area Network) standard, has undergone several revisions. Table 1.1 summarizes the main objectives of some versions of this family of standards.

Standard	Objective	Status
IEEE 802.16-2004	Revision, integrating above extensions	Complete
IEEE 802.16e-2005	Amendment on enhancements to support mobility	Complete
IEEE 802.16-2009	Revision, Air interface for fixed and mobile broadband wireless access [22]	Current
IEEE 802.16j-2009	Multihop relay [23]	Current
IEEE 802.16m	Advanced air interface supporting 100 Mbps data rates for mobile clients and 1 Gbps data rates for fixed clients	Draft

Table 1.1: Major IEEE 802.16 versions

The IEEE 802.16-2004 [20] covers most of the specifications for the physical (PHY) and medium access control (MAC) layers for fixed broadband wireless access networks. The standard defines two architectural modes of operation — *point to multipoint* (PMP) mode and the *mesh* mode. The operation of point to multipoint mode is similar to that within a cell in cellular network.

In the standard, the mesh routers are known as the base stations (BS) and the mesh clients are called subscriber stations (SS). In PMP mode each SS communicates with a particular BS to reach an outside network. In mesh mode any two neighboring mesh routers can communicate with each other. In this mode data is relayed between the SS to and the gateways connecting to the external network in a multi-hop fashion.

We describe some key aspects of the IEEE 802.16-2004 standard in the following subsections.

1.3.1 Physical Layer

The IEEE 802.16-2004 standard supports operation in one of two frequency bands: in 10-66 GHz band for environments with line-of-sight (LOS) propagation and in 2-11 GHz band for environments with non-LOS propagation.

The standard specifies four physical layer (PHY) specifications to support LOS and non-LOS propagation:

1. WirelessMAN-SC PHY
2. WirelessMAN-SCa PHY
3. WirelessMAN-OFDM PHY
4. WirelessMAN-OFDMA PHY

WirelessMAN-SC PHY and WirelessMAN-SCa PHY are based on single carrier modulation. The former supports LOS propagation whereas the latter supports non-LOS propagation.

Both the OFDM PHYs are aimed at supporting non-LOS propagation and employ multiple subcarriers. The WirelessMAN-OFDM PHY uses 256 subcarriers whereas WirelessMAN-OFDMA uses 2048 subcarriers. The latter physical layer specifications facilitates communication over more than one orthogonal channels using multiple radio interfaces, and motivate us to consider multi-channel cases of routing and scheduling problem in chapter 5.

1.3.2 MAC Layer

The MAC layer specification is an important part of the IEEE 802.16-2004 standard. In the following, we discuss the reference model, connection management, QoS provisioning, scheduling services, and bandwidth management in the MAC layer.

Reference Model: The MAC layer has three sublayers:

- *service-specific convergence sublayer*,
- *common part sublayer*, and
- *security sublayer*.

The convergence sublayer is responsible for mapping higher level data units into MAC service data units, and protocol header suppression. The common part deals with core functions of MAC layer, bandwidth allocation, QoS management, and connection establishment and maintenance. The MAC layer supports different PHY specifications through time division duplexing (TDD) and frequency division duplexing (FDD). The common part sublayer deals with the duplexing scheme as well. The security sublayer handles authentication, secure key exchange, and encryption/decryption.

Connection Management: In PMP mode, the MAC layer assigns a 16-bit connection identifier (CID) to each connection between a subscriber station and the base station. On initialization, up to three pairs of management connections are initiated for MAC management. Different QoS levels are employed for these connections. The first management connection carries short and urgent messages, the second one carries longer, delay tolerant management messages and the third one carries protocol specific messages. In mesh mode neighboring nodes are identified through 8-bit link identifiers (Link ID) which in turn become part of CID in the MAC header.

QoS Provisioning: The MAC layer provides different levels of QoS through *service flows*. A service flow is a flow of packets over a single link that is associated with certain QoS parameters like latency, jitter, and throughput. Each service flow is assigned a 32-bit service flow identifier (SFID). Active service flows are mapped to connections and their associated SFIDs are mapped to the corresponding CIDs. Such link level QoS assurance is employed to enable end-to-end QoS provisioning.

Scheduling Services: An active service flow mapped to a particular connection needs proper scheduling to deliver the level of QoS associated with the flow. For this purpose, each connection is associated with a *data service* characterized by QoS parameters identifying its behavior. The MAC scheduler distinguishes four data service types:

- Unsolicited Grant Service (UGS): The QoS parameters (*e.g.*, maximum latency and tolerated jitter) are designed to support real-time streaming traffic with periodic fixed-size packets.

- Real-time Polling Service (rtPS): The QoS parameters, including minimum traffic rate, tolerated jitter, and latency, support real-time streaming traffic with periodic variable-sized packets (*e.g.*, MPEG video).

- Non-real-time Polling Service (nrtPS): The QoS parameters support delay-tolerant streams of variable-sized packets that require minimum bandwidth assurance.

- Best Effort (BE): This data service type is suitable for services requiring no minimum service level.

The QoS parameters required to map a data service type to a service flow are summarized in table 1.2.

	UGS	rtPS	nrtPS	BE
Minimum reserved traffic rate		$\surd$	$\surd$	
Maximum sustained traffic rate	$\surd$	$\surd$	$\surd$	$\surd$
Maximum latency	$\surd$	$\surd$		
Tolerated jitter	$\surd$			
Traffic priority			$\surd$	$\surd$
Request/transmission policy	$\surd$	$\surd$	$\surd$	$\surd$

Table 1.2: Mandatory QoS parameters

Bandwidth Management: Bandwidth request and allocation mechanisms help in assuring QoS parameters particularly related to traffic rate. The standard specifies a number of mechanisms for requesting bandwidth:

- In PMP mode, a subscriber station sends connection-specific bandwidth request to the base station. The request can be incremental, *i.e.*, for the new connection only, or aggregate, *i.e.*, for all connections. The base station computes allowable bandwidth for the subscriber station based on current allocation and the received request, and sends back grant message to the subscriber station. The allocated bandwidth reflects aggregate allocation, not connection-specific allocation. This mode of allocation is suitable for data traffic.

- Bandwidth for control traffic like request/grant messages can be allocated in polling cycles. A portion of the frame is reserved for polling purposes. When the number of subscriber stations is small, unicast polling is performed. If sufficient bandwidth is not available to poll the subscriber stations individually, multicast or broadcast polling is initiated.

- The standard also describes contention-based bandwidth negotiation schemes for OFDM PHYs.

In mesh mode, there are no bandwidth request/grant mechanism that reserves slots in uplink (or downlink) frames. Instead the routers consider 2-hop neighborhood for constructing transmission schedule. The standard discusses three-way handshake mechanism for both centralized and distributed scheduling.

End-to-end QoS Provisioning: Although the standard provides mechanisms for QoS provisioning at the link level, it does not discuss scheduling algorithms aimed at ensuring end-to-end QoS levels. The standard leaves the innovations in provisioning end-to-end QoS parameters to network providers for promoting competitive service delivery. The routing mechanisms are also beyond the scope of the standard. Our contributions in this thesis makes use of the existing features provided by the standard (*e.g.*, TDD, connection management, incremental and aggregate bandwidth request mechanisms), and attempts to develop innovative solutions for end-to-end QoS provisioning through non-bifurcated routing and scheduling.

1.4 Overview of the IEEE 802.11 Standards

WMNs can also be designed using the IEEE 802.11 standard [21] and its extensions. A number of research work consider problems in mesh networks based on the IEEE 802.11 (see *e.g.*, [46, 10]). The standard has been revised several times since its inception, and amendments and extensions are being developed on a continual basis. Major revisions and extensions of the standard are listed in table 1.3 (cf. Appendix B for a complete list).

Standard	Objective	Status
IEEE 802.11e	MAC enhancements for QoS support	Complete
IEEE 802.11-2007	Integrates a,b,d,e, and g-j	Current
IEEE 802.11n	600 Mbps MIMO at 2.4 and 5 GHz	Current
IEEE 802.11s	Mesh networking	Draft

Table 1.3: Major IEEE 802.11 Standards and Extensions

The standard describes physical layer (PHY) specifications and medium access control (MAC) for wireless local area networks (WLANs). In this section we mention some key aspects of the standard, and the extensions being proposed to support broadband wireless access and mesh networking.

1.4.1 Physical Layer

The IEEE 802.11-2007 standard supports operation in the 2.4 GHz and 5 GHz bands. A number of physical layer specifications are provided for supporting network operation in these bands:

- Frequency-hopping spread spectrum (FHSS) PHY specification for the 2.4 GHz industrial, scientific, and medical (ISM) band

- Direct sequence spread spectrum (DSSS) PHY specification for the 2.4 GHz band designated for ISM applications
- Infrared (IR) PHY specification
- Orthogonal frequency division multiplexing (OFDM) PHY specification for the 5 GHz band
- High rate DSSS PHY specification

Each of these physical layers performs framing of data packets, and physical medium dependent transmission/reception.

1.4.2 MAC Layer

The MAC layer is responsible for ensuring direct transmission of data packets between two adjacent stations in the network. The MAC layer also performs packet ordering, and security services. The security services include encryption/decryption, authentication, and access control in conjunction with layer management.

The functionality of the MAC layer in the IEEE 802.11 is managed by either of the following three coordination functions:

1. Distributed coordination function (DCF),
2. Point coordination function (PCF), or
3. Hybrid coordination function (HCF).

DCF: In DCF, the MAC layer employs carrier sense multiple access with collision avoidance (CSMA/CA). According to this scheme, each station checks to determine if another station is transmitting. If no other station is transmitting, *i.e.*, , the medium is not busy for a predetermined interval, then a station can proceed with its own transmission. Otherwise the station refrains from transmitting till the ongoing transmission comes to an end. In case of transmissions of two stations beginning almost at the same time, collision will take place, and the stations will backoff for a random interval. Before attempting retransmission after deferral or successful transmission, each station adjusts its backoff interval. In order to reduce the number of collisions, the stations employ a four way handshake protocol involving request-to-send, clear-to-send, data, and acknowledgment packets (RTS-CTS-DATA-ACK).

PCF: PCF is suitable for infrastructure network configurations where one of the stations acts as an access point (AP). The AP performs polling operation to control access to the medium by the other stations. The PCF is built on DCF, and therefore share a number of common characteristics in operation. However, the AP can control parameters in the coordination function to give priority to PCF over DCF.

HCF: The HCF is usable only in networks that support QoS assurance. It combines functions from DCF and PCF, QoS-specific mechanisms and frame subtypes to allow a uniform set of frame exchange sequences for data transfer with QoS support. It uses both a contention-based channel access method, called the enhanced distributed channel access (EDCA), and a controlled channel access method, called the HCF controlled channel access (HCCA).

1.4.3 Support for WMN

The IEEE 802.11n amendment is aimed to support high throughput [18]. In this amendment the proposed enhancements in MAC layer achieves the data rate of 150 Mb/s in a single spatial stream. The amendment supports up to four spatial streams using multiple-input multiple-output (MIMO) antennas. The supported data rate is comparable to the IEEE 802.16 standard, albeit over shorter distances, and suitable for providing broadband wireless access.

The IEEE 802.11s aims to support mesh networking using the IEEE 802.11 MAC. One of the design goals in the IEEE 802.11s is to achieve seamless integration with other 802.11 networks. Both infrastructure mesh and client mesh architectures in section 1.1 can be realized with this specification as devices or stations in the network can assume the role of mesh routers. The IEEE 802.11s proposes hybrid wireless mesh protocol (HWMP) as the default routing protocol. The HWMP is based on ad hoc on-demand distance vector routing (AODV) and tree-based routing. However, the IEEE 802.11s also allows vendors to employ other routing protocols. Our contribution in chapter 3 fits nicely in this context.

1.5 Thesis Contributions and Organization

The thesis contributions are as follows:

1. In chapter 3, we consider routing in WMNs that employ a contention-based MAC protocol. Prior to our work in this chapter, the state of the art on multi-hop wireless networks utilizing contention-based MAC protocols include routing algorithms that split traffic among different paths. Some of the more important work in this direction are reviewed in chapter 2. Our goal in chapter 3 is to develop routing algorithms for allocating bandwidth in a controlled manner to traffic requests in WMNs employing CSMA/CA. To the best of our knowledge, no algorithm for allocating bandwidth over a non-bifurcated route between any given source and destination node can provide performance guarantees for the class of contention-based MACs. The work in th chapter 3 tries to fill this gap in the literature. In section 3.3, we formalize a *non-bifurcated* routing problem as an optimization problem that takes into account interference aspects in WMNs. We develop a novel routing algorithm adapted from the theory of maximum flows to solve the problem. The main ingredient in our algorithm is a method for computing *interference constrained flow augmenting paths* (IC-FAPs) in the network in order to route subscriber demands. The work in chapter 3 is the basis of publication [39].

2. In chapter 4, we consider joint routing and scheduling in TDMA-based WMNs to achieve high throughput. Many existing results in literature focus on routing and scheduling in such networks. Among these results, a few distinguished results consider **joint** routing and scheduling problems. Dealing with combined routing and scheduling for multiple flows gives rise to intractable optimization problems. Many approaches rely on decoupling routing from scheduling to get effective albeit non-optimal

solution. In chapter 4, we consider the class of *incremental* routing and scheduling problems. In a typical problem in this class we are given: a WMN, a schedule for routing some existing flows in the network, and a set of new flow requests to be routed. The problem is to route as many new flow requests as possible without perturbing the given schedule of the existing flows. Since grid networks arise in many applications, in chapter 4, we tackle such problem for the class of grid networks. We devise an algorithm that strives to maximize the number of served flow requests by attempting to route a pair of new requests in each iteration. The work in chapter 4 is the basis of publication [37].

3. In chapter 5, we consider routing and scheduling in a generalized context where

   - WMNs can have arbitrary topology, and
   - the maximum interference radius of a node in the network is specified by a given input parameter.

   Prior to our work in this chapter, the hardness of the joint routing and scheduling problems promoted the development of heuristic algorithms. In contrast, our work in chapter 5 characterizes some suitable conditions under which the problem of joint routing and scheduling a flow can be solved efficiently. We consider the problem of routing a new flow request between some two nodes in the network. Using a cost function for utilizing the slots in a schedule, we formalize the joint routing and scheduling problem as an optimization problem that calls for computing a minimum cost schedulable route between the specified end nodes. Our contributions in this chapter include:

   - We obtain graph theoretic characterization of conflict graphs of paths and trees that takes into account the interference distance.
   - We show that using the above characterization and using existing results in the literature we can solve the single flow scheduling problem (SFS) over a given route exactly.

- Moreover, when no route is specified, we show an exact algorithm that solves the single flow routing and scheduling (SFRS) problem in the generalized context described above.

Work based on chapter 5 appears in [36] and [38].

The rest of the thesis is organized as follows: In chapter 2 we discuss a number of research work related to QoS provisioning in WMNs through development of routing and scheduling algorithms. We discuss the covered literature in two threads. The first thread focuses on the development of practical routing and scheduling algorithms in WMNs. The second thread focuses on estimating the throughput capacity in such networks under some idealistic assumptions. In subsequent chapters we present our contributions on non-bifurcated routing and scheduling in WMNs. Finally in chapter 6 we present a summary along with a few possible research directions for future work.

1.6 Summary

In this chapter we present the motivations and the scope of our research on routing and scheduling for QoS provisioning in WMNs. We discuss WMN architectures and the enabling standards in order to elaborate on the relevance of our research in these contexts. We also mention highlights of our contributions in the thesis, and present an outline of the following chapters.

Chapter 2

Literature Review

The amount of research work on routing and scheduling in multi-hop wireless networks is extensive. In this chapter we highlight some important results that are relevant to WMNs. Our presentation classifies the surveyed research results into two broad categories. In the first category, we review research work intended for constructing practical algorithms for scheduling, routing, and channel allocation in WMNs. These algorithms are designed to solve some optimization problem such as maximizing the network throughput. In the second category, we review some algorithms aimed at studying asymptotic results on network capacity defined as the maximum throughput achieved under some idealized assumptions. Algorithms in the latter category concern bandwidth allocation, routing, and channel allocation in multi-channel networks. These algorithms are not typically practical due to assumptions and restrictions of the respective problem formulation.

2.1 Introduction

Efficient routing and scheduling is important for providing BWA and QoS provisioning for end-to-end flows in WMNs. In this chapter, we review some important results on routing and scheduling in WMNs. Broadly speaking, the covered results are classified into two categories: practical algorithms and asymptotic results. A number of the practical algorithms discussed are centralized while the other ones are distributed. We remark that some of the centralized algorithms are intended to provide tools for characterizing network capacity.

In the second category, researchers consider obtaining asymptotic results on achievable throughput. Multi-hop wireless networks experience degradation of throughput with the increase in the number of communicating nodes due to the presence of interference between nodes and the behavior of MAC protocols used for resolving contentions. Research work on asymptotic results address the question of scalability of multi-hop wireless networks. In the following, we summarize some of the aspects that distinguish resource management algorithms for routing, scheduling, and channel assignment intended to derive asymptotic results.

1. Resource management algorithms used in deriving asymptotic results are typically simple (they are amenable to analysis) and efficient (they are used to obtain asymptotic results intended to serve as good lower bounds on the achievable throughput). Such algorithms for deriving lower bounds serve as points of reference when discussing related resource management algorithms.

2. The obtained asymptotic results are typically succinct expressions that capture the effect of few critical parameters on the achieved throughput. Existence of such expressions simplifies the development of intuitive understanding of the behavior of such networks.

3. The approaches used in deriving these results typically abstract away from the limitations inherent in practical MAC protocols with distributed coordination function. This abstraction is achieved by assuming an idealized MAC protocol that does not waste bandwidth for resolving contention. Such abstraction captures the effect of interference only, and thus the obtained results may be viewed as upper bounds on the throughput that can be obtained using a distributed MAC protocol.

On the other hand, the algorithms used to derive asymptotic results may not serve well in practice for the following reasons:

1. The obtained results often focus on some primary parameters (*e.g.*, the number of nodes) and assume uniform distributions of some secondary parameters (*e.g.*, data rate, transmission range, *etc.*). Changing such uniform distribution for secondary parameters may necessitate substantial change in the corresponding resource management algorithms.

2. The assumption pertaining to using idealized MAC protocols that do not waste bandwidth for resolving contention is not practical.

In the next section, we describe a number of interference models commonly encountered in the literature that are referred to in subsequent sections.

2.2 Interference Models

Mathematical modeling of interference in the wireless medium plays an important role in the design of routing and scheduling algorithms for multi-hop wireless networks. The following are some of the commonly used interference models.

Protocol Model: The *protocol model* [17] determines successful transmission based on distance. Following the notation in [17], we denote the distance between two nodes u and v as $|u - v|$. A transmission from a node u to another node v is successful if for any other node w, $|w - v| \leq (1 + \Delta)|u - v|$. The parameter $\Delta > 0$ acts as a guard zone. This model implicitly assumes that each node in the network will adopt a power control mechanism for the transmissions from the node. The *fixed power protocol model* refers to the special case of protocol model when all nodes have the same transmission power and transmission range.

Physical Model: The *physical model* [17] for interference determines successful transmission based on transmission power of transmitting nodes, ambient noise, and reception threshold. If P_u is the transmission power of node u, then a transmission from node u is successfully received by node v if

$$\frac{\frac{P_u}{|u-v|^\alpha}}{N + \sum_{w \neq u, w \neq v} \frac{P_w}{|w-v|^\alpha}} \geq \beta,$$

where N is ambient noise power, α is a signal decay exponent, and β is a reception threshold.

RTS/CTS Model: In networks employing carrier sense multiple access with collision avoidance (CSMA/CA) and the RTS/CTS handshake operation the protocol model implies that two links interfere with each other if the sender or receiver of a link is within interference range of the sender or receiver of the other link. Thus to avoid interference between a pair of transmitter and receiver nodes, all nodes within the interference range of either of the nodes are required to refrain from transmission. The interference relationship between links is modeled in literature as a *conflict graph* where each vertex represents a transmission link in the network and an edge is placed between two interfering transmission links.

2.3 Algorithms for Routing, Channel Assignment and Scheduling in WMN

In this section, we highlight some of the more relevant routing and scheduling algorithms in literature concerning multi-hop WMNs.

2.3.1 The Work of Wang *et al.* [56]

Wang *et al.* [56] present both centralized and distributed algorithms to solve a joint routing and scheduling problem for static multi-hop TDMA wireless networks where a subset of the nodes act as gateways. The authors consider three interference models: the protocol model, the fixed power protocol model, and the RTS/CTS model. The joint routing and scheduling problem under study is called the *max-min-fairness* routing problem, and is formalized as a mixed integer linear program. In the formulation, for all non-gateway nodes, the served traffic computed in the solution is a fraction (denoted λ) of the offered traffic. For a node u in the network the offered traffic is denoted $\ell(u)$, and the served traffic is denoted $f(u)$. The objective of the integer program is to maximize the fraction λ subject to flow conservation constraints, link capacity constraints, and scheduling constraints.

The amount of flow over a link e is denoted $f(e)$, and the capacity of the link is denoted $c(e)$. The served traffic $f(u)$ of node u, and the amount of flow over the set of incoming links (the set denoted by $\Lambda^+(u)$) determine the amount of flow over the set of outgoing links (the set denoted by $\Lambda^-(u)$).

The schedule to be constructed has a period T, *i.e.*, slots in the schedule are numbered $1, 2, \ldots, T$. The binary variable $X_{e,t}$ indicates whether link e is active during slot t. The fraction of time a link e is active in a schedule is denoted $\alpha(e)$. Thus $\alpha(e) = (\sum_{1\le t\le T} X_{e,t})/T$. One may observe that the amount of flow $f(e)$ over link e must be the same fraction $\alpha(e)$ of the capacity $c(e)$ of that link. Following the rule in the RTS/CTS model, if a link $e' \in I(e)$ where $I(e)$ is the set of links interfering with link e, then an interference-free schedule satisfies $X_{e,t} + X_{e',t} \le 1$, for all t. The problem formulation considers all these restrictions. The resulting mixed integer linear program is as follows:

$$
\begin{aligned}
& \max \lambda \quad \text{subject to} & \\
\sum_{e\in\Lambda^+(u)} f(e) - \sum_{e\in\Lambda^-(u)} f(e) &= f(u), & \forall u \notin \text{ gateway nodes} \\
f(u) &\ge \lambda \ell(u) & \forall u \notin \text{ gateway nodes} \\
\alpha(e).c(e) &= f(e) & \forall e \\
\alpha(e) &\ge 0 & \forall e \\
\alpha(e) &\le 1 & \forall e \\
X_{e,t} + X_{e',t} &\le 1 & \forall e' \in I(e), \forall e, \forall t \\
(\sum_{1\le t\le T} X_{e,t})/T &= \alpha(e) & \forall e \\
X_{e,t} &\in \{0,1\} & \forall e, \forall t.
\end{aligned}
$$

The first two constraints in the program are flow conservation constraints, the next three constraints are capacity constraints, and the remaining constraints are scheduling constraints.

Under suitable conditions, the problem formulated above is NP-hard. The authors relax the mixed integer linear program to a linear program in order to obtain an approximate solution. The relaxation is achieved by allowing the variables $X_{e,t}$ to take fractional values and adding some constraints. For example, in one of the relaxed formulations in [56], for each link e, the following constraint is added to ensure the existence of a schedule:

$$\alpha(e) + \sum_{e' \in I(e)} \alpha(e') \leq 1.$$

The above inequality is more restricted than the similar inequality $X_{e,t}+X_{e',t} \leq 1$ in the original problem formulation.

The solution obtained from the relaxed linear program can potentially split a flow along multiple paths. Such relaxation enables the authors to devise centralized and distributed scheduling algorithms that have constant approximation ratios. The approximation ratio of an obtained schedule depends on the interference model, and the ratio between interference range and transmission range.

2.3.2 The Work of Kabbani *et al.* [25]

The authors consider a scheduling problem in wireless backhaul networks found in infrastructure WMNs (cf. section 1.1). As in [56] discussed in the previous section, the nodes in the backhaul network are synchronized in time and employ TDMA. The authors assume the availability of a fixed tree rooted at the gateway node for routing traffic from nodes in the backbone to the gateway. Under suitable conditions, the scheduling problem for obtaining maximum throughput is equivalent to the *maximum weighted independent set* problem. The general maximum weighted independent set problem is NP-complete. However, the authors exploit properties in the *contention graph* (*i.e.*, conflict graph) to obtain a scheduling sequence, and devise a linear time centralized algorithm that solves the scheduling problem optimally on the given routing tree. Guided by the developed centralized algorithm the authors then develop a distributed slotted MAC protocol that can support all feasible arrival rates in the backbone network.

2.3.3 The Work of Wu *et al.* [58]

In contrast to the work in [25], Wu *et al.* consider additional constraints in scheduling in WMNs. In a typical scheduling problem for a TDMA WMN, time-slots allocated to a particular link are not required to be consecutive in the constructed schedule. This type of scheduling is pre-emptive from the perspective of the links appearing in the schedule. In [58], Wu *et al.* consider *non-preemptive* link scheduling in wireless mesh networks, where the non-preemptive constraint requires slots allocated to a particular link in any frame to be consecutive. The non-preemptive scheduling is NP-hard even in the case of tree topologies when each link requires arbitrary number of slots in a frame. The authors present a heuristic algorithm that solves the underlying problem approximately. The approximation ratio is bounded by the maximum degree of the conflict graph.

2.3.4 The Work of Kodialam and Nandagopal [27]

The authors present centralized algorithms to solve joint routing and scheduling problem in WMN where one of the centralized algorithms is a tool for analyzing the capacity of network, *i.e.*, the solution is intended to provide good lower bound on performance of network. They consider a WMN with C mutually orthogonal channels and n fixed routers (or nodes). A router v has $\kappa(v)$ radio transceivers capable of channel switching. There is synchronization among the nodes *i.e.*, routers operate in a time-slotted fashion. The assumed interference model is the protocol model of [17].

The input to the joint routing and scheduling problem is a graph representing the network and interfering links, and the desired data rates of each router. The output is a set of routes, link channel assignments, and associated schedule achieving the rates. In case of infeasibility of the problem instance the output indicates the condition. The problem is NP-hard. The authors suggest a linear programming formulation by relaxing the original problem.

Table 2.1: List of symbols

G:	Network graph
V:	Set of vertices
E:	Set of data links
$E^{\mathcal{I}}$:	Set of interference links
OC:	Set of orthogonal channels
$\kappa(v)$:	Number of radios at node v
$\varrho(e)$:	Maximum number of channels available for link e
$f_i(e)$:	Flow rate of channel i over link e
$c_i(e)$:	Capacity of channel i over link e
$g_i(e)$:	$f_i(e)/c_i(e)$, utilization
$t(e)$:	Transmitting node of link $e \in E$
$h(e)$:	Receiving node of link $e \in E$
$E(v)$:	Set of data links incident on node v
$E^{\mathcal{I}}(v)$:	Set of interference links incident on node v
$y_i^t(e)$:	A binary variable set to 1 if link e is active on channel i in time-slot t

Table 2.1 describes the some of the symbols used in the problem formulation.

The variables in the linear program are $f_i(e)$'s. The objective function can be tailored to reflect optimization of different performance metrics like maximizing network throughput or achievability of demand vector. The constraints in the original problem form the feasibility conditions for the set of given link flow sets $\boldsymbol{f} = \{f_i(e) | i \in OC\}$ for each link $e \in E$:

Link-channel constraint:

$$\sum_{i \in OC} y_i^t(e) \leq \varrho(e), \quad \text{for all } e \in E, \text{ for all } t \tag{2.1}$$

Node-radio constraint:

$$\sum_{e \in E(v)} \sum_{i \in OC} y_i^t(e) \leq \kappa(v), \quad \text{for all } v \in V, \text{ for all } t \tag{2.2}$$

Interference link constraint:

$$\sum_{e' \in E(t(e)) \cup E(h(e))} y_i^t(e') \leq 1, \quad \text{for all } i \in OC, \text{ for all } e \in E \cup E^{\mathcal{I}}, \text{ for all } t \tag{2.3}$$

The relaxed constraints form part of the constraint set of the linear program. These are necessary conditions of feasibility and expressed in terms of utilization:

$$\sum_{i \in OC} g_i(e) \le \varrho(e), \qquad \text{for all } e \in E \tag{2.4}$$

$$\sum_{e \in E(v)} \sum_{i \in OC} g_i(e) \le \kappa(v), \qquad \text{for all } v \in V \tag{2.5}$$

$$\sum_{e' \in E(t(e)) \cup E(h(e))} g_i(e') \le 1, \qquad \text{for all } i \in OC, \text{ for all } e \in E \cup E^{\mathcal{I}} \tag{2.6}$$

The key steps of the rest of the algorithm is presented below:

- The feasibility problem is cast as another linear programming problem called concurrent flow problem. Then a primal-dual algorithm is applied to solve the problem. The algorithm gradually assigns routes to each flow, and in essence solves a shortest path problem.

- The solution to the linear programming problem is used for assigning channels to the links. In static channel assignment load balancing is performed, and at each iteration of channel assignment, preference is given to the channel that creates the least amount of load on all constraint sets. In the dynamic setting, where the channel assignment is changed periodically, flows are packed greedily in the least loaded channel on a link.

- Once channel assignment is complete, scheduling is done by a coloring algorithm. The flows are first scaled to integral flows by multiplication with a large number (and ignoring fractional part). Time-slots are assigned to each channel separately while keeping the number of active links incident on a node at any given time to be at most the number of radio interfaces at that node.

The relaxation approach combined with the objective of throughput maximization can provide a bound on achievable throughput *i.e.*, capacity region of the network under the given conditions.

2.3.5 The Work of Alicherry *et al.* [2]

Alicherry *et al.* discuss joint channel assignment and routing problem [2] for an architecture similar to that in the work of Kodialam and Nandagopal [27]. They also solve a linear programming formulation that is a relaxed version of joint channel assignment and routing problem [2]. The solution to the linear programming is subsequently used in channel assignment, flow scaling, and link scheduling. The linear program models the constraints of channels, flows, and interference in a manner similar to that in [27]. The difference lies in the channel assignment algorithm which operates in multiple stages of approximation. Each stage refines the solutions of a previous stage to bring them closer to feasible assignments. The channel assignments are then further processed by means of linear programming and flow scaling in order to satisfy interference constraints of the original problem. In the final step flows are scheduled on links taking into account available time-slots. The obtained solution is a factor-8 approximation of the optimal solution, and thus provide an estimate of maximum achievable throughput.

2.3.6 The Work of Bajerano *et al.* [3]

Bajerano *et al.* consider the problem of max-min fair bandwidth allocation in WMNs and offer a centralized approximate solution to the problem. The key features of the system model are:

- The mesh routers are static and communicate among themselves using directional antennas.
- Communication between a user and a router is carried out by means of omnidirectional antennas.
- The solution is periodically computed by a network operation center (NOC) that operates in the wired infrastructure outside the mesh network. The authors refer to such architecture a configurable access network (CAN).
- Communication links have constant capacity.

- The effect of interference is negligible due to presence of directional antennas and a sufficient number of non-interfering channels.

Each router has a number of users associated with it, and a certain amount of bandwidth is allocated to a router or node for routing traffic to/from the users. If d_v is the number of users associated with node (router) v, and b_v is the bandwidth allocated for that node, then the average bandwidth per user is b_v/d_v for $d_v > 0$. The normalized bandwidth is defined as the minimum average bandwidth among all nodes with $d_v > 0$, and the objective of the optimization problem is to maximize this normalized bandwidth.

Unlike multi-path solutions in [27] and [2], the authors consider single-path solution, which is of more interest from practical point of view. Two routing models of single-path routing are considered in this work. In the *aggregate flow model* all traffic from a router to a gateway follows the same path. In the *user traffic model* traffic from a single router may take different paths, but traffic from a single user to the gateway is always routed along a single path. The latter model is more relevant to our research.

The optimization problem of maximizing normalized bandwidth allocation with single path routes is NP-hard when d_v's are distinct. The authors propose approximation algorithms for the problem. The key steps of the basic algorithm are:

- Solve a linear programming problem modeling the constraints and obtain a fractional solution with multi-path routes.

- Round the solution toward an integral solution and then apply the single-source unsplittable flow algorithm of Dinitz *et al.* [9].

- Finally scale down the bandwidth allocation over all nodes to accommodate network constraints.

This work provides a number of approximation ratios for the produced solutions which is not very common in relevant research works. The following are the properties of the solution produced by the basic algorithm in an aggregate flow model:

- If all nodes have the *same bandwidth requirement* and all links have *same capacity* then the algorithm produces optimal integral solution.
- If nodes have *different bandwidth requirements* but capacities of all links are same, then the algorithm produces a 2-approximation.
- If the ratio of maximum and minimum bandwidth requirements of a node is bounded by a constant α, then the algorithm produces a $(1+\alpha)$-approximation. If $\alpha = 1$, *i.e.*, node demands are uniform then the solution is a 2-approximation.
- The algorithm produces 5-approximation for *arbitrary demands* and *arbitrary link capacities.*

In case of user flow model with each user having same bandwidth requirement:

- If all links have the *same capacity* then the algorithm finds optimal integral solution.
- If links have *arbitrary capacities* then the algorithm finds a 2-approximation.

2.3.7 The Work of Raniwala and Chiueh [46]

The authors consider a multi-channel WMN with fixed nodes. Some nodes are connected to end-user devices, some nodes act as gateways to wired network, and the other nodes act as pure routers (figure 2.1). Each node is equipped with multiple network interface cards (NICs). However, the number of available radio channels are more than the number of NICs present at a node. The interference range of a node is assumed to be larger than its transmission range.

The authors propose a distributed algorithm for solving a joint routing and channel assignment problem in multi-channel WMN. The objective of the problem is to assign channels to NICs so that the bandwidth usage is optimized. The authors refer to their implementation as *Hyacinth.* The approach

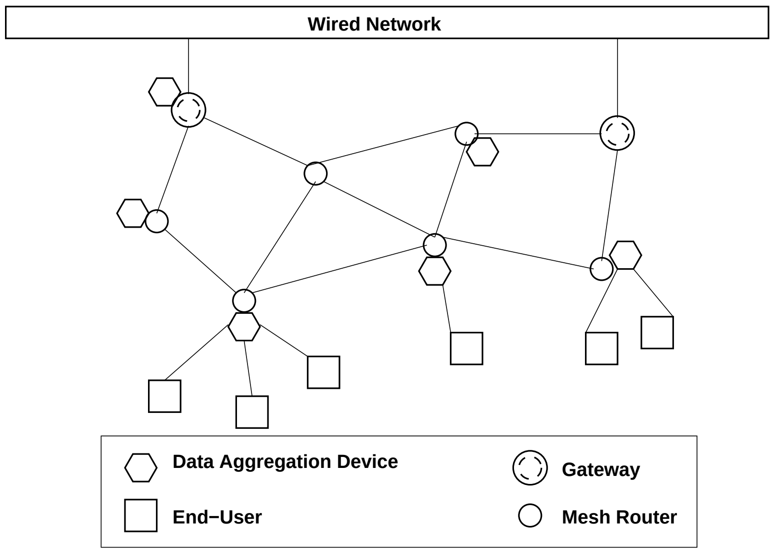


Figure 2.1: The Hyachinth Architecture

is based on the idea of load balancing among both neighbors and available channels.

The authors break down the joint routing and channel assignment problem into two subproblems:

1. Neighbors to NIC assignment problem, and
2. Interfaces to channel assignment problem.

The initial step in solving the first subproblem is to perform neighbor discovery. The next step is to form routing trees by message passing as described below:

- Gateway nodes propagate *advertise* messages along with available bandwidths toward nodes or routers.
- A node may receive *advertise* messages from more that one gateway. It greedily selects (with the objective of balancing load) a gateway as root of routing tree and sends *join* message toward the gateway.
- A gateway (and intermediate nodes) then sends *accept* message to the nodes if they can be accommodated.

- If a node receives *reject* message instead of *accept* message then it tries to be part of another routing tree rooted at a different gateway.

- Each node becomes part of one routing tree under some gateway after an exchange of *advertise*, *join* and *accept/reject* messages.

Depending on traffic load, a node can also change routing tree by sending *leave* message to the previous gateway and *join* message to the new gateway. The authors describe a protocol for managing priorities in direction of message passing with minimal conflicts. The next step in load balancing is achieved by grouping of neighbors and assigning each group to a particular interface. The solution to the second subproblem involves selection of a channel for a particular interface that results in minimum interference.

2.3.8 The Work of Draves *et al.* [10]

Draves *et al.* consider a network with fixed topology. The nodes in the network are assumed to equipped with one or more 802.11 radios. In the case of a node with multiple radios, the radios operate on pairwise orthogonal frequencies. The authors propose a distributed protocol called Multi-Radio Link Quality Source Routing (MR-LQSR) in order to make better utilization of available channel capacity. The protocol is an extension of a DSR-based protocol.

The MR-LQSR protocol performs neighbor discovery, link weight assignment, and propagation of link weights prior to making routing decisions based on the chosen metric. The protocol differs from dynamic source routing (DSR) in the weight assignment stage and metric specification. The authors consider packet delays or waiting times as an indicator of resource usage or congestion, and propose a metric that takes into account delays while routing packets. One component of the metric is cumulative expected transmission time (CETT). For a path consisting of n hops,

$$\text{CETT } = \sum_{i=1}^{n} \text{ETT}_i, \tag{2.7}$$

where ETT_i is the expected transmission time on the i-th hop. The cumulative transmission time is an approximation for end-to-end transmission delay.

The other component of the proposed metric is the transmission time of the bottleneck channel. Thus for a network with k channels, transmission time of the bottleneck channel is:

$$\max_{1 \le j \le k} X_j, \quad \text{where} \quad X_j = \sum_{\text{hop } i \text{ is on channel } j} \text{ETT}_i, \quad 1 \le j \le k \tag{2.8}$$

The authors propose the weighted sum of CETT and transmission time of the bottleneck channel as the routing metric:

$$\text{WCETT} \ = (1 - \beta) \times \text{CETT} + \beta \times \max_{1 \le j \le k} X_j, \tag{2.9}$$

where β is an adjustable parameter in the range $[0, 1]$. Packets are forwarded along the path that minimizes the metric.

Note that the WCETT metric can be viewed as an extended version of the expected transmission count (ETX) metric proposed by De Couto *et al.* [7]. The ETX metric tries to find high throughput paths for outgoing packets in a single channel WMN. Mathematically,

$$\text{ETX} \ = \frac{1}{d_f \times d_r}, \tag{2.10}$$

where d_f is the forward delivery ratio, and d_r is the reverse delivery ratio. Forward delivery ratio is the measured probability that a data packet successfully arrives at the destination, and the reverse delivery ratio is the probability that the acknowledgment packet is successfully received.

2.3.9 Other Algorithms

- Hsiao *et al.* assume interference free point-to-point links between neighboring nodes, and perform load-balanced routing by using heuristic for constructing tree structure(s) for routing [19]. They consider both channel assignment and routing in a multi-channel network.

- Jain *et al.* model interference explicitly in a conflict graph and propose multi-path routing for mesh networks [24].

- Tang *et al.* propose an incremental algorithm for bandwidth aware routing [51]. The algorithm assigns path or route to sessions as they appear.

- Kyasanur and Vaidya present shortest path based routing algorithm that also takes into account the effect of interference [31].
- In [45] Peppas and Turgut take into account varying degrees of node mobility in client mesh networks (cf. section 1.1), and propose a hybrid routing algorithm. The algorithm employs both reactive and proactive routing protocols proposed for ad-hoc networks.
- In [16] Gupta *et al.* consider developing queue length-aware distributed scheduling algorithms whose throughput is close to that of maximal schedules.
- In [14] and [41] the authors consider mixed integer linear programming formulations for joint routing and scheduling. An exact solution to such formulation is feasible for only small sized networks.

2.4 Asymptotic Results on Capacity of Multi-hop Wireless Networks

In this section, we present an overview of asymptotic results on network capacity in single-radio and multi-radio networks.

2.4.1 The Work of Gupta and Kumar [17]

Gupta and Kumar present a number of asymptotic results for network throughput. They consider an ad-hoc network where n nodes are positioned in a static manner. All the nodes in the network are assumed to be placed within a unit radius disk. Each node in the network can act as traffic source and is paired with another node for which the data traffic is destined. Each node has the same transmission range. However, this range can be tuned for optimum performance. Each node is capable of transmitting W bits per second.

The results consider two dimensions of the network settings: topology control and interference model.

1. **Topology control:** Some results in [17] are derived assuming the nodes of a network are placed randomly using uniform distribution. Such networks are called *random networks.* Other results in [17] are derived assuming that the nodes can be placed in a certain way so as to maximize the achieved throughput. These networks are called *arbitrary networks.*

2. **Interference model:** The authors consider two interference models: the protocol model, and the physical model.

In the following discussion, we summarize the method of obtaining asymptotic result for the case of arbitrary network under the protocol model of interference. The input of the problem is the number of nodes n, and the outputs are the topology and the transmission range of the nodes. The key ideas in the construction of the sought after optimum topology and the determination of the optimum transmission range are:

- The plane is tessellated in squares of side $\Theta(\frac{1}{\sqrt{n}})$ and the center of the disk is positioned at the origin. The nodes are placed around the corners of the squares along grid lines.

- The transmission range is adjusted so that each transmitter can successfully transmit to its closest receiver without causing interference according to the protocol model.

Thus, there can be $n/2$ sender-receiver pairs, where only a single hop transmission occurs between each sender-receiver pair. Under the above construction, the achieved throughput of the network has been shown in [17] to be $\Omega(W\sqrt{n})$ bit-meters per second. Since the network has $n/2$ sender-receiver pairs, it follows that the throughput for each pair is $\Omega(\frac{W}{\sqrt{n}})$ bit-meters per second.

In case of random networks the per node average throughput has been shown to be $\Theta(\frac{W}{\sqrt{n \log n}})$ bits/second. Thus, in both arbitrary and random networks the throughput diminishes as the number of nodes increases.

2.4.2 The Work of Tse and Grossglauser [15]

In contrast to the work of [17], Tse and Grossglauser provide asymptotic expression for the achievable throughput in ad-hoc networks with mobile nodes. In the network, n mobile nodes are placed over a disk of unit area. The topology of the network can change significantly over time due to node mobility. Each node in the network transmits data at the same rate and has the same transmission range. Each node transmits data to another randomly chosen node. The interference model adopted in this work is similar to the physical model considered in [17].

The authors show that packets should be relayed in order to improve throughput. In particular, an important result in [15] shows that single-node relaying scheme can improve average throughput to $\Theta(1)$ bits/second per node. According to this scheme the sender transmits a packet to a relay node, the relay moves around in the network, and when the opportunity arrives the relay node transmits the data to the destination node. The approach assumes that

- At each node indefinitely large buffer space is available for guaranteed data delivery.
- The delay in packet delivery can be arbitrarily large, and applications running on the nodes do not have strict delay constraints.

2.4.3 The Work of Neely and Modiano [43]

Unlike the work of [15], where traffic is assumed to be delay-tolerant, Neely and Modiano consider a mobile ad-hoc network where the following restrictions apply:

The network is partitioned into a constant number of non-overlapping equal-sized cells (the dimensions of each cell is assumed to be part of problem input). Each node in the network moves around from one cell to another according to an independent and identically distributed (*i.i.d.*) mobility model. Communication is assumed to be confined to intra-cell transfers only. The transmissions occur in a time-slotted fashion and only one transfer can take place in a cell during one slot. Communication occurs between node pairs chosen from a well-defined set of source-destination pairs. However, during any time-slot, a source and the corresponding destination are not required to be located in the same cell. Communication can be bidirectional between a source-destination pair. Each node is capable of sending a packet along multiple paths toward the associated destination.

The authors propose three relaying schemes for achieving high throughput with bounded packet delay in such networks:

- 2-hop relay algorithm
- 2-hop relay algorithm with redundant transfer
- Multi-hop relay with redundant transfer (fair packet flooding protocol).

The second and third schemes employ more than one relay for improving the delay bound. However, with the improvement in packet delay, achieved throughput can degrade.

The 2-hop relay algorithm achieves $O(1)$ throughput with $O(N)$ delay, where N is the number of nodes in the network. This is a modified version of Tse and Grossglauser's relaying approach. In any cell of the network, if the sender node and the corresponding receiver node are present, priority is given to direct transfer between this source-destination pair. Otherwise two nodes are chosen with one from the set of senders, and the other from the set of receivers. Another node from the remaining nodes in the cell is chosen, which might be a relay node. The third node is randomly assigned the role of a receiving or transmitting relay. In the receiving mode, it receives data to be relayed from the sender node in the cell, whereas in the transmitting mode it sends data to the receiver if it has piggybacked any data previously.

Redundant transfers (for multi-path routing) in 2-hop relaying scheme reduce the delay to $O(1/\sqrt{N})$ but at the same time lower the per node throughput to $O(1/\sqrt{N})$. In this scheme packets are sent to more than one relays within a cell.

Multi-hop relaying can further improve delay bounds. The fair packet flooding protocol in [43] employs multi-hop relaying with redundant packet transfers and achieves a packet delay of $O(\log N)$ with $O(\frac{1}{N \log N})$ per node throughput.

2.4.4 The Work of Kyasanur and Vaidya [30]

Unlike the research works above where each node is equipped with one radio interface, Kyasanur and Vaidya consider a network of n static nodes, each of which is equipped with m interfaces, and each interface can utilize one of c available wireless channels ($1 \leq m \leq c$). The authors refer to such network as an (m, c)-network. The nodes are assumed to be placed on a torus of unit surface area. The torus surface assumption avoids the complications in mathematical analysis arising out of boundary conditions (*i.e.*, edge effects). The authors derive a number of results corresponding to the following cases:

- The authors consider two channel models. In one channel model, the aggregate bandwidth W is fixed, and the bandwidth of each channel is $\frac{W}{c}$. In the other channel model, the bandwidth of each channel W is fixed, and the aggregate bandwidth is Wc (*i.e.*, varies with the number of channels).

- As in [17], the authors also consider arbitrary and random networks (see section 2.4.1) and provide asymptotic results for both cases.

Interference is modeled according to the protocol model in [17] for all cases. In each of the network settings described above, the achievable throughput depends on the values of c and m. Different ranges of values of $\frac{c}{m}$ yield different asymptotic expressions.

Below we present the key steps in the throughput result for random networks where $\frac{c}{m}$ is $O(\log n)$. The cases where $\frac{c}{m}$ is $O(\log n)$ are of practical interest since in real scenarios the number of available channels can be substantially large compared to the number of interfaces installed in a node. The result states that the achieved throughput is $\Theta(W\sqrt{\frac{n}{\log n}})$.

The derivation makes use of the following results pertaining to (m, c)-networks and $(1, c)$-networks:

Lemma 2.1. *An (m, c)-network can support at least the capacity supported by a $(1, \tilde{c})$-network, where m, c, and $\tilde{c}$ are integers, and $\tilde{c} = \frac{c}{m}$.*

Lemma 2.2. *An (m, c)-network can support at least half the capacity supported by a $\left(1, \lfloor \frac{c}{m} \rfloor\right)$-network.*

In the following we show that if $m = 1$, and $c = O(\log n)$, then the achieved throughput is $\Theta(W\sqrt{\frac{n}{\log n}})$. By lemmas 2.1 and 2.2, it then follows that for any $m > 1$, the throughput for a random (m, c) network is $\Theta(W\sqrt{\frac{n}{\log n}})$ when $\frac{c}{m}$ is $O(\log n)$.

Asymptotic Result on Throughput of Random $(1, c)$-network

The obtained result relies on the use of a particular joint routing and scheduling algorithm. Below we sketch the main ideas behind the algorithm. The algorithm can be broken down into two major steps:

1. *cell construction and route planning*, and
2. *transmission scheduling.*

The scheduling process has two sub-steps, namely *routing graph construction* and *interference graph construction.*

Cell construction:

- The given surface of the unit torus is divided into square cells of area $a(n)$ each. In the analysis, the value of $a(n)$ is chosen to limit the number of nodes in each cell to a desirable range.
- In the analysis, the transmission range, $r(n)$, is computed as a function of the cell size $a(n)$ so as to limit the maximum number of interfering cells to be a constant independent of n and $a(n)$.

Route planning:

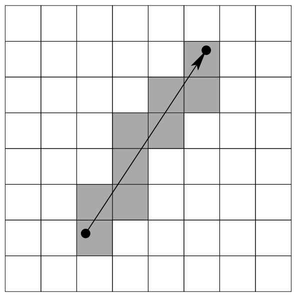


Figure 2.2: Cells selected for routing

- In the construction, the route between a source and the corresponding destination is assumed to pass only through cells that intersect the straight line joining these two nodes (figure 2.2).
- At a cell where a node is either the source or destination of the route, that node is selected among the nodes in the cell.
- At other cells along the route, each flow (*i.e.*, traffic between a source-destination pair) is assigned to a node that is least loaded in terms of the number of flows assigned to it so far. This scheme balances load on nodes, and the number of flows assigned to a node can be expressed asymptotically.

Scheduling Transmissions:

Routing Graph Construction:

- The routing graph is constructed as follows: the nodes correspond to the nodes in the network, and for each flow in the network an edge is placed between a pair of nodes if they appear consecutively in the route for that flow.
- The routing graph is a multigraph which can be edge-colored using a limited number of colors. The number of colors used in the coloring indicate the number of slots in a frame.

- A frame is divided into slots. Each slot corresponds to a channel and is assigned a color from the routing graph. An edge in the routing graph is active during the color-slot associated with it.

Interference Graph Construction:

- The interference graph is constructed as follows: the nodes in the graph correspond to the nodes in the network and an edge is placed between two nodes if the nodes interfere with each other.

- With a constant bound on the number of interfering cells per node, and a bounded number of nodes per cell, the number of edges incident on a node in the interference graph is also bounded. The interference graph can therefore be easily vertex-colored using limited number of colors. The number of colors used indicate the number of mini-slots in each slot of a frame.

- Each edge-color slot obtained from the routing graph is subdivided into mini-slots on every channel.

- When a node is active on a particular edge-color slot, it is allowed to transmit in a precomputed mini-slot on designated channel ensuring non-interfering transmission.

- The bound on the length of each mini-slot and the amount of transferred bits determine the asymptotic expression for network throughput.

Two key values that establish the asymptotic results in this scheme are $a(n)$ and $r(n)$. For the chosen range of $\frac{c}{m}$, the optimal choices for $a(n)$ and $r(n)$ have been shown to be $\frac{100 \log n}{n}$ and $\sqrt{8a(n)}$ respectively.

We remark that the scheduling scheme assumes arbitrarily divisible time-slots or arbitrarily long frame size for scheduling transmission which may not be directly adopted in practical scenarios. The coloring algorithms, however, provide a simple method for obtaining conflict-free schedules where such frames are feasible.

2.5 Summary

In this chapter we present a discussion of the research work that focus on capacity as well as research work aimed at developing routing and scheduling algorithms. The discussion of the work on routing and scheduling in WMNs in this chapter serves in identifying the distinctive aspects of our contributions in subsequent chapters.

Chapter 3

Non-bifurcated Routing in CSMA/CA-based WMNs

The main results in this chapter are original contributions of the thesis [39]. In this chapter we consider traffic routing in the IEEE 802.11-based multi-hop WMNs utilizing CSMA/CA protocols. As mentioned in chapter 1, providing BWA over WMN requires the support of key traffic types like TCP traffic, delay-jitter sensitive traffic, and traffic requiring synchronized delivery to end users. To this end, we consider non-bifurcated routing schemes to avoid introducing wide variations of delay by avoiding routing the same flow on multiple paths. In this chapter we formalize the problem of non-bifurcated routing, while meeting subscriber demands, as an optimization problem that takes into account interference aspects in the network. We present a heuristic that utilizes the theory of maximum flows taking into account interference aspects induced by the CSMA/CA protocol with RTS/CTS messages. To solve the problem we describe a method to compute interference constrained flow augmenting paths (IC-FAPs). Simulation experiments indicate improved achieved throughput, and delay-jitter results obtained by our approach over the use of the well-known Dynamic Source Routing (DSR) algorithm used in ad-hoc networks. Publication [39] is based on the work presented in this chapter.

3.1 Introduction

In this chapter we consider routing in WMNs that employ a contention-based MAC protocol. This type of networks can be built using technologies available for WLANs. As mentioned in chapter 1, routing algorithms for WMNs intended to provide BWA are required to allocate bandwidth to subscribers in a controlled manner, so as to satisfy service agreements. In addition, such mechanisms are required to provide acceptable throughput for key traffic types, such as TCP traffic, delay-jitter sensitive traffic, and multimedia traffic that requires synchronized delivery to end users. Since delays on different routes in such networks may vary widely, routing of the above traffic types can potentially benefit from non-bifurcated routing schemes that do not split flows among multiple paths. The issue of avoiding traffic splitting has been mentioned (but not dealt with), for example, in [27]. We remark that the advantages of non-bifurcated routing have been discussed in [11] for ad-hoc networks and in [50] for sensor networks. Forwarding all packets of a given traffic flow over a single path enables uniform treatment of the packets in each intermediate node which promotes efficient handling of flows requiring QoS guarantees and enables simplified network monitoring and management.

A novel aspect of our work here is the development and evaluation of a non-bifurcated routing algorithm. It is shown that the performance of the resulting algorithm is competitive when compared with the well known dynamic source routing (DSR) algorithm.

3.2 System Model

In this chapter we consider WMNs consisting of fixed wireless routers (the WMN nodes) that are capable of aggregating traffic from subscriber units. Some routers act as gateways to the wired Internet. Routers employ multi-hopping to relay subscriber traffic to (or form) the gateway(s). In the general case, each router node is equipped with a number of 802.11-based radio interfaces, and there is more than one orthogonal wireless channel to utilize. We assume that the routers use the IEEE 802.11 CSMA/CA protocol. To model the interference, and contention of flows in the 802.11-based RTS-CTS-DATA-ACK environment, we adopt the following commonly used assumptions (see, *e.g.*, [27], [35], [46]):

1. All nodes are assumed to have the same transmission range, denoted R_T, and interference range, denoted R_I, where $R_I \geq R_T$.

2. Two nodes that are within the R_T range of each other can receive transmission from each other (*i.e.*, communicate directly).

3. If two nodes are not within the R_T range, but within the interference range (R_I) of each other, then they cannot communicate directly, but they interfere with each other.

4. Two flows using different orthogonal channels do not interfere with each other.

5. Using RTS/CTS model explained in chapter 2, a transmission on a certain link and channel is successful when all potential interferers in the neighborhood of the sender and the receiver are silent on the channel for the duration of the transmission.

Thus, as remarked in [35], under the above assumptions neighborhood and flow contention are commutative properties. In addition, the above assumptions imply that two flows contend with each other if either the sender, or the receiver of one flow coincides with, or lies within the interference range of the sender, or the receiver of the other flow.

In the context of non-bifurcated routing, we define a flow as a sequence of packets that can be uniquely identified at each WMN node. Each flow is required to be routed to (or from) one of the available gateways without splitting at any intermediate node on the route. In addition, each flow requires a certain amount of data rate. We assume that the network operator defines the equivalent data rate of a flow unit, and that applications make requests to their serving WMN nodes in integer multiples of such units.

3.3 Single Channel Problem Formulation

In this section, we consider traffic routing over a single wireless channel operating under the interference model mentioned in section 3.2. In particular, we formulate the problem of maximizing the total amount of flow (throughput) served by the network at any time as a network flow problem.

We denote by $G = (V, E_T, E_I)$ the directed graph on the set V of WMN nodes, of which a subset of nodes $GW \subset V$ serves as gateways. E_T denotes the set of transmission edges, and E_I is the set of interference edges. As noted above, since neighborhood is assumed to be commutative, if a directed edge $(x, y) \in E_T$ (or E_I) then the reverse edge $(y, x) \in E_T$ (respectively, E_I). Moreover, by the above remarks, a flow $f(x, y)$ affects the network in the same way as a flow $f(y, x)$ of equal amount on the reverse edge. Hence, a flow on a route from a mesh node x to a gateway g affects the network in the same way as a flow of equal amount on the same route from g to x. Thus, without loss of generality, we may assume that all flow demands are directed toward the gateway(s). Moreover, since flow demands from a gateway's own subscriber units are routed directly to outside the mesh, we simplify the problem by omitting such demands from the problem formulation.

Notation: We formulate the throughput maximization problem using the following additional notations:

- $D(x)$ (requested flow demands at node x): a vector $(d_i(x) : i = 1, 2, \ldots, |D(x)|)$, where the i-th requested flow demand has value d_i units. The vectors of requested flow demands are assumed to be sent periodically to a central node that is responsible for computing new sets of routes, and subsequently distributing the computed routes to all other nodes.

- $S(x)$ (accepted flow demands at node x): a vector specifying the flow demands of $D(x)$ that are selected for routing.

- f: a vector (computed by an algorithm) specifying for each transmission edge (x, y) a flow of value $f(x, y)$.

Remark: To simplify the presentation below we use the symbol f to denote a number of aggregate nodes flows (*e.g.*, aggregate flows between two sets of nodes, flows generated at one node, and aggregate flows over a set of transmission links). In each use the argument uniquely identifies the intended aggregate flow with no conflict in notation.

- $f(X, Y)$ (aggregate flow notation): for two sets of nodes $X, Y \subseteq V$, the sum of flows assigned to transmission edges, where each edge has its tail in X and its head in Y. That is, $f(X, Y) = \sum\{f(i, j) : i \in X, j \in Y$, and $(i, j) \in E_T\}$. We also write $f(X, u)$ (or, $f(u, X)$) if one set is just a single node u.

- $E_T^{int}(x)$: denotes the set of transmission edges where each edge has at least one of its end nodes located within the interference range of node x.

- $f(E_T^{int}(x))$: $f(E_T^{int}(x))$ denotes the sum of all flows assigned to transmission edges in $E_T^{int}(x)$.

- $f(D(x))$ and $f(S(x))$: $f(D(x))$ denotes the sum of all demands in the vector $D(x)$. $f(S(x))$ is defined similarly.

- $C(x)$: the available channel capacity at node x (*i.e.*, the highest data rate supported by a wireless network interface at any node in the network). The model allows different nodes to have different channel capacities to account for the possible outside interference on wireless channels in the unlicensed wireless bands.

- $\ell(x)$: To simplify the presentation, we also define the channel loading factor at node x, denoted $\ell(x)$, caused by a given flow vector f, as:

$$\ell(x) = f(V, x) + f(x, V) + f(E_T^{int}(x))$$

where $f(V, x)$ is the sum of all flows entering x, $f(x, V)$ is the sum of all flows leaving x, and $f(E_T^{int}(x))$ is the sum of all interfering flows at x. We note that, $f(S(x))$ (the sum of all flows entering x from its own subscriber units, and accepted for routing) does not contribute to the channel loading factor, since we assume that such flows do not use the same wireless channel used for backhaul communication between the WMN nodes.

Our model hypothesizes that the vectors $(S(x) : x$ is a non-gateway node) of accepted traffic flows admit non-bifurcated routing to the gateway(s) if there exists a flow vector f that satisfies the following constraints:

Channel Capacity Constraint. For any node x, the channel loading factor caused by the flow vector f does not exceed the available channel capacity at the node: that is, $\ell(x) \leq C(x)$.

Flow Conservation Constraint. For any non-gateway node x, the sum of the outgoing flows from x equals the sum of the incoming flows into x, plus the flow demands accepted for routing; that is, $f(x, V) = f(V, x) + f(S(x))$.

Flow Indivisibility Constraint. For any node x, and flow demand $d_i(x) \in D(x)$, a flow of amount $d_i(x)$ is assigned a route from x to a gateway in G.

Finally, the **throughput maximization** problem is to maximize the total flow routed to the gateway(s). That is, using the aggregate flow notation, we want to maximize $f(V, GW)$.

3.4 Background Results and Remarks

We observe the following aspects of the computational complexity of the throughput maximization problem:

1. The problem with arbitrary integer requested flow demands (*i.e.*, the numbers in a D vector), can be shown to be NP-complete. In this general case, the PARTITION problem ([SP12] in [12]) can be transformed into the above problem in polynomial time.

2. We note that the simplified throughput maximization problem where all terms of the form $f(E_T^{int}(x)) = 0$ (*i.e.*, there is no interference), and all demand vectors include unit flows only, is equivalent to a maximum flow

problem with multiple sources and sinks, and capacity constraints on nodes. This latter problem, however, can be solved using an algorithm for solving the standard 2-terminal maximum flow problem (see, *e.g.*, [6]).

3. A prominent class of algorithms for solving the 2-terminal maximum flow problem (*e.g.*, see [6] for the Ford-Fulkerson, and the Edmonds-Karp algorithms), relies on the idea of repeatedly finding a flow augmentation path (FAP), until no such FAP exists. A FAP is a sequence of edges that form an undirected path from a source node s to a terminal node t. Thus, the path may traverse some edges in the forward direction, and traverse other edges in the reverse direction. So, relative to such an undirected path P, some edges are forward edges, and some edges are reverse edges. It is known that if each forward edge admits a flow increase by v units, and each reverse edge admits a decrease of its current assigned flow by v units, then adopting such flow changes along an (s, t) FAP results in a net increase of the total flow from s to t by v units.

3.5 Interference-constrained Flow Augmenting Paths

We extend the idea of flow augmenting paths mentioned in section 3.4 to design a solution to the throughput maximization problem in the context of non-bifurcated routing. We call the new type of paths, interference-constrained FAPs (or, IC-FAPs for short).

Definition 3.1 (IC-FAP of value v). *Given the connectivity graph $G = (V, E)$ of a WMN, with a vector of flow values assigned to the edges, we define an IC-FAP of value v from some demand node x to a network's gateway as an undirected path such that increasing the flow value on each forward edge by v units, and decreasing the flow value on each reverse edge by v units yield a flow vector that does not violate the channel capacity constraints at any node.*

The following example illustrates the above definition.

Example 3.1. Figure 3.1 illustrates a WMN where node g is a gateway. Initially, the available channel capacity at each node is assumed to be $C = 25$

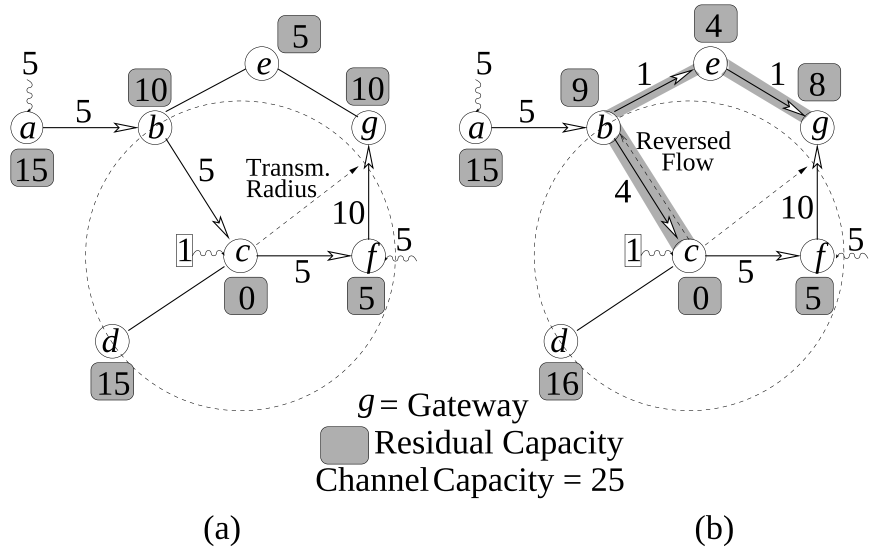


Figure 3.1: An example of an IC-FAP in a wireless mesh network

units. Figure 3.1 illustrates a flow of 5 units sent along the path (a, b, c, f, g), and another flow of 5 units sent along the edge (f, g). The total flow into the gateway is 10 units. The available residual channel capacity at each node appears inside an adjacent square. For example, the load factor at node f, $\ell(f) = f(c, f) + f(f, g) + f(b, c) = 20$ units, where the first two terms are pass-through flows, and the third term is an interference flow. Thus, node f's residual capacity $C(f) = 25 - 20 = 5$ units. Likewise, the load factor at node c, $\ell(c) = f(b, c) + f(c, f) + f(a, b) + f(f, g) = 25$ units, where the first two terms account for the pass-through flows, and the last two terms account for interference flows. Thus, node c's residual capacity $C(c) = 25 - 25 = 0$ unit.

The network in figure 3.1(a) admits the IC-FAP of value $v = 1$, highlighted in figure 3.1(b). The IC-FAP is from node c to the gateway g along the path $P = (c, b, e, g)$. Here, the first edge (b, c) is traversed in the reverse direction (its associated flow is decreased by $v = 1$ unit), and the two edges (b, e) and (e, g) are traversed in the forward direction (their flows are increased by $v = 1$ unit each). Figure 3.1(b) also illustrates the resulting residual capacities at each node after modifying the flows along the indicated IC-FAP. ■

A few remarks now follow in order.

1. We remark that, as in the case of the standard maximum flow FAPs, starting with a feasible flow vector, and making flow changes along an IC-FAP yield a new flow vector that satisfies the flow conservation constraints, as the original flow vector. Thus, starting with the zero flow vector, and repeatedly finding IC-FAPs can be used to obtain feasible flows with higher net flow amounts into the gateway.

2. Given a network and an initial flow vector, the ability to employ the above mentioned iterative scheme to solve the throughput maximization problem hinges on the ability to find an IC-FAP efficiently. Currently, no such efficient exact algorithm appears to be known. Finding such an IC-FAP from some demand node s to a gateway t appears to face the following computational difficulty: if x is a possible intermediate node, $s \neq x \neq t$, on a sought after IC-FAP, then the ability to extend a given path segment from s to x, so as to reach t, appears to depend on the exact distribution of loading factors caused by increasing the flows along the forward edges, and decreasing the flows along the reverse edge of the given segment. Thus, an exact algorithm may have to examine exponentially many paths from s to x. Such apparent computational difficulty does not exist in the search problem for finding a FAP in the standard two-terminal maximum flow problem.

3. In the IC-FAP highlighted in figure 3.1(b), the first edge (b, c) is traversed in reverse direction, and the associated flow is decreased by $v = 1$ unit. This flow augmentation step has the effect of re-routing of $v = 1$ unit of another existing flow f_r. If the magnitude of f_r is just 1 unit then this re-routing step does not violate the non-bifurcated routing of f_r. Otherwise (if f_r is of magnitude larger than 1 unit) this re-routing step violates the non-bifurcated routing. In general, when an IC-FAP of $v > 1$ units is sought, the re-routing process does not violate the non-bifurcated routing of an affected existing flow f_r if f_r is of the same magnitude of v units.

4. In a search for an IC-FAP, if $P = (x_0, x_1, \ldots, x_r)$, $r \geq 2$, is a directed path from some node x_0 to another node x_r along which a unit of flow can be sent by traversing each edge in the forward direction, and if (x_0, x_r) is also an edge in G, then the search algorithm should not consider the longer path P. To see why, let y be any arbitrary node in G, and denote by $\ell(y)$, and $\ell'(y)$ the loading factors that result from sending unit flow along the edge (x_0, x_r), and the path P, respectively. One may then verify that $\ell(y) \leq \ell'(y)$, and hence using the edge (x_0, x_r) is always the better choice. The algorithm presented in section 3.6 uses the above remark by giving preference to extending a path to reach nodes that are as close as possible to the gateway.

3.6 IC-FAP Search Algorithm

In this section we present a search algorithm, called ICFAP_Find (cf., figure 3.3) for the flow augmentation problem mentioned above. Table 3.1 describes the function inputs and output.

Input Parameters:	
G:	The directed graph of a WMN
$flow$:	An array specifying for each directed edge (x, y) a flow value $flow(x, y)$; the values constitute a feasible flow in the network G
cap:	A vector specifying for each node x the residual channel capacity obtained by taking all $flow$ values into consideration
$Nforward$:	An array specifying for each node x two closest neighbors of x to the gateway t, $x \neq t$; if x has one neighbor, the second node is set to null (zero value)
v_{req}:	The required flow increment value of the sought after IC-FAP from demand node s to gateway t
s:	A node with a required unsplittable flow demand of value v_{req}
t:	A target gateway in G
Output:	
P	an IC-FAP from s to t of the required value v_{req}, returned upon a successful search (else, the returned path P is empty)

Table 3.1: Function ICFAP_Find inputs and output

As can bee seen, the function takes as input the connectivity graph G of a WMN, an array $flow$ of current flows routed in the network, and the resulting residual channel capacity at each node. The function searches for an IC-FAP from a given demand node s to the network's gateway t that increases the net flow in the network by the amount specified by v_{req}.

The input array $Nforward$ is one ingredient in a mechanism utilized by the function to bound the number of stored IC-FAPs from s to any intermediate node x during the search. Specifically, if $Nforward[x] = (y_1, y_2)$ (or, $(y_1, 0)$ if x has only one neighbor), then y_1 and y_2 are closest neighbors of x to the gateway t (ties are broken arbitrarily). The algorithm keeps a collection of IC-FAPs from s to x, where each IC-FAP induces a certain distribution of channel loading factors at these two distinguished nodes y_1 and y_2. Two different partial IC-FAPs from s to x that result in the same distribution of channel loading factors at y_1 and y_2 are then considered indifferent by the function. Hence, only one of the two paths is kept in the stored collection.

A second ingredient in bounding the number of examined IC-FAPs from s to x is a table, denoted T_x, maintained for each node x. T_x provides a key-value mapping from pairs of integers (loading factors at the $Nforward[x] = (y_1, y_2)$ nodes), to IC-FAPs from s to x; the net flow along each of the stored IC-FAPs is v_{req} units.

T_x is used in the following way. Initially, T_x is empty (no key exists in the domain of T_x). Subsequently, if $Nforward[x] = (y_1, y_2)$, and $(\ell(y_1), \ell(y_2))$ is a key in T_x (*i.e.*, a pair of channel loading factors at nodes y_1, and y_2, respectively), then $T_x[\ell(y_1), \ell(y_2)]$ is an IC-FAP, denoted P, from s to x; we interpret that the input vector $flow$, combined with the flows assigned to the IC-FAP segment P result in channel loading factors of values $\ell(y_1)$, and $\ell(y_2)$, at nodes y_1 and y_2, respectively. Moreover, if $y_2 = 0$ (the null value), then $\ell(y_2) = 0$. The following example illustrates the above concepts.

Example 3.2. Figure 3.2 illustrates a WMN where node a serves as a gateway. Function ICFAP_Find is assumed to be called with the following settings:

- The input vector $flow$ assigns a unit of flow to each of the four thick edges along the path from node j to the gateway a.

- The residual capacity vector cap is assumed to allow any of the flow augmentations mentioned below.

- The parameter s (the demand node) is set to g, and the required search is for an IC-FAP from g to the gateway a, with value $v_{req} = 1$ unit.

- The array $Nforward$ contains the following values: $Nforward[g] = (f, 0)$, $Nforward[f] = (c, e)$, $Nforward[c] = (b, 0)$, and $Nforward[e] = (b, d)$.

Initially, all tables are empty. Subsequently, the search starts with node g, and considers a partial IC-FAP that sends one flow unit from g to f. Since the flow is assumed admissible, and $Nforward[f] = (c, e)$, the function computes the resulting loading factors at c, and e:
$\ell(c) = 3$ $(= flow(e, b) + flow(b, a) + f(g, f))$, and
$\ell(e) = 5$ $(= flow(j, h) + flow(h, e) + flow(e, b) + flow(b, a) + f(g, f))$,
and inserts the entry $T_f[3, 5] = (g)$, where (g) is the initial part of the path (g, f).

Subsequently, the function considers extending the path from node f by sending a flow unit to each of f's neighbors (other than g), leading to the following cases.

- Since the flow along the path (g, f, c) is assumed admissible, and $Nforward[c] = (b, 0)$, the function computes the resulting loading factor at b:
  $\ell(b) = 4$ $(= flow(h, e) + flow(e, b) + flow(b, a) + f(f, c))$,
  and inserts the entry $T_c[4, 0] = (g, f)$, where (g, f) is the initial part of the path (g, f, c).

- Since the flow along the path (g, f, e) is assumed admissible, and $Nforward[e] = (b, d)$, the function computes the resulting loading factor at b, and d:

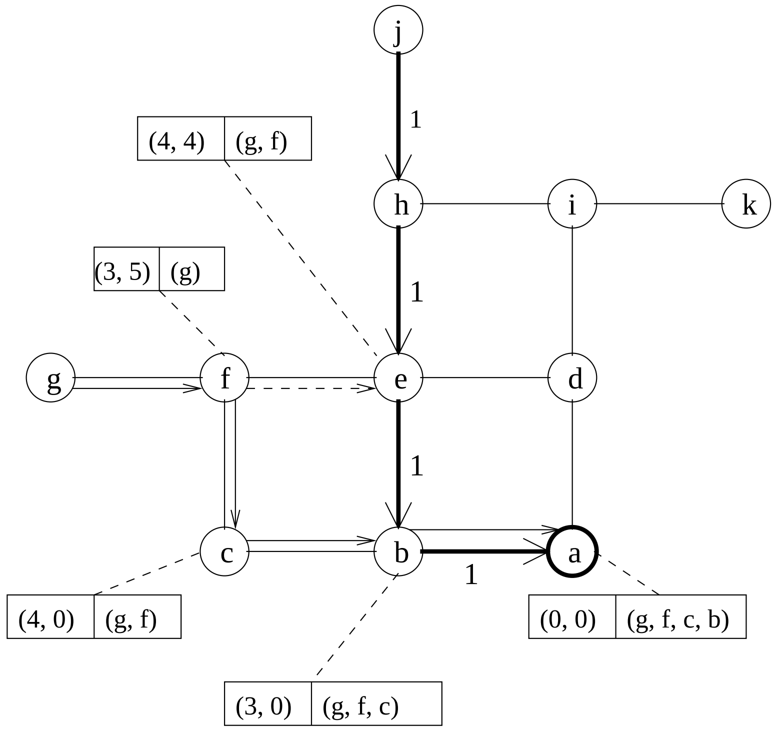


Figure 3.2: Example of the tables maintained by the algorithm

$\ell(b) = 4$ $(= flow(h, e) + flow(e, b) + flow(b, a) + f(f, e))$, and
$\ell(d) = 4$ $(= flow(h, e) + flow(e, b) + flow(b, a) + f(f, e))$,
and inserts the entry $T_e[4, 4] = (g, f)$, where (g, f) is the initial part of the path (g, f, e).

Next, the search continues from node c. Eventually, the search reaches the gateway a via b, and the path stored in $T_a[0, 0] = (g, f, c, b)$, with node a appended, is returned by the function. ■

We now describe the basic steps performed by the function in Fig. 3.3. Step 1 initializes the IC-FAP table T_x at each node x to empty. Step 2 starts the search by setting x (the current node) to the input demand node s. The while-loop in step 3 iterates until an IC-FAP from s to t is found.

Step 3.1 has two nested for-loops: the outer loop expands the search by examining the neighbors of the current node x in a non-decreasing order of their distances to the gateway t. Note that the algorithm (in step 3.2) selects the first neighbor y of x for which an IC-FAP from s to y of value v_{req} is found for further extension. Hence, the above ordering gives preference to extending

```
Function ICFAP_Find    $(G, flow, cap, Nforward, v_{req}, s, t)$:
 Inputs and Outputs: As described in Table 3.1 above.
  1. For each node $x$ in the WMN $G$, initialize table $T_x$ to empty
  2. Start the search from the current node $x = s$
  3. While (the gateway node $t$ is not reached) {
     3.1 for each neighbor $y$ of the current node $x$ (in a non-decreasing order of
         the distances from the gateway), and each index $(\ell_1, \ell_2)$ in the table $T_x$
         (in the order described in the main text) {

         a. Let $P = T_x[\ell_1, \ell_2]$ be the stored candidate IC-FAP from $s$ to $x$ along
              which a flow increment of $v_{req}$ units is possible

         b. If extending $P$ by pushing forward $v_{req}$ units of flow along the di-
              rected edge $(x, y)$ so as to obtain a valid IC-FAP from $s$ to $y$ is
              possible, then update table $T_y$ accordingly; mark the first node
              $y$ for which the above extension is possible

         c. Else if extending $P$ by pushing backward $v_{req}$ units of flow along
              the directed edge $(y, x)$ so that only one existing flow needs to
              be re-routed (so as to obtain an IC-FAP from $s$ to $y$) is possible,
              then update table $T_y$ accordingly; mark the first node $y$ for which
              the above extension is possible
     3.2 If a marked node $y$ has been identified in the above step, then set the
         current node $x = y$, and expand the search for an IC-FAP further by
         continuing the while loop. Else (if no such node $y$ is marked) then exit
         the while-loop
     }
  }
  4. Return the IC-FAP stored at $T_t[0, 0]$
```

Figure 3.3: Pseudo-code for function ICFAP_Find

paths that terminate in close proximity of the gateway.

For a given neighbor y of the current node x, the inner for-loop in step 3.1 examines each of the potential partial IC-FAPs stored in T_x. The IC-FAPs are considered in a particular ordering of their associated keys. The ordering is defined by the following relation: for two different keys (ℓ_1, ℓ_2), and (ℓ'_1, ℓ'_2), we write $(\ell_1, \ell_2) \leq (\ell'_1, \ell'_2)$ if $\max(\ell_1, \ell_2) \leq \max(\ell'_1, \ell'_2)$, or the maximum values are equal, and $\min(\ell_1, \ell_2) \leq \min(\ell'_1, \ell'_2)$. The paths are considered in a non-decreasing order of the above relation on the associated keys. So, a candidate partial IC-FAP from s to x receives higher priority if it minimizes the maximum loading factor at the forward nodes in $Nforward[x]$.

Steps 3.1.b and 3.1.c consider augmenting a path P stored in T_x with a link between the current node x, and one of its neighbors y. If the resulting augmented path satisfies the required flow constraints, and has the required value v_{req}, table T_y is updated with the new augmented path. Care is taken in implementing the above flow augmentation step so that any existing amount $flow(y, x)$ is first reduced as much as possible from the required value v_{req}, and then the possibly remaining amount is sent forward on the edge (x, y). Subsequently, if step 3.1 succeeds in identifying an IC-FAP from s to y, step 3.2 adopts the first such identified node as the current node x from which the search continues.

3.6.1 Running Time

To achieve efficiency in the running time, the function avoids performing exhaustive search, while maintaining awareness of the channel loading factors caused by sending new flows along the selected paths. In the worst case, the while-loop of step 3 iterates once for each node x in G. If we denote the maximum residual channel capacity at each of the $Nforward[x]$ nodes by $c_f(x)$, then the table T_x stores at most $c_f^2(x)$ paths. Additionally, if we denote by $d(x)$ the number of one-hop neighbors of x, then the for-loops in step 3.1 perform at most $d(x).c_f^2(x)$ iterations. In each iteration, step 3.1.b (or 3.1.c) checks channel constraints at each node in G, thus each iteration requires $O(n)$ time. The worst case total running time of the function is thus $O(\sum_{x \in V} c_f^2(x).d(x).n)$. Thus, if m is the number of links in G, and the maximum residual capacity at any node in the network is c_{max} then the running time is $O(n^2.c_{max}.m)$.

3.6.2 Applications and Extensions

Function ICFAP_Find described above provides a tool for tackling a variety of non-bifurcated routing problems on WMNs using conceptually simple algorithmic ideas (*e.g.*, greedy algorithms, search algorithms, *etc.*). In this section, we briefly discuss the use of function ICFAP_Find to tackle three extended problems.

We start by considering the gateway throughput maximization problem for single-channel wireless networks. Given the NP-completeness result of the single-channel problem, as mentioned in section 3.4, the running time of any exact solution of the problem is expected to grow exponentially with the available channel capacity (*i.e.*, the highest data rate supported by a wireless network interface at any node in the network) C, when flow demands assume arbitrary integer values in the range $[1, C]$. A simple framework for tackling the above maximization problem may compute the best result obtained by performing a number of iterations; each iteration starts by fixing an ordering of the set of all (node, flow demand) pairs: $\{(x, d) : x \in V, \text{ and } d \in D(x)\}$, and repeatedly calling function ICFAP_Find to search for an IC-FAP to route the flow demand d from x to the gateway g; each iteration terminates with a net gateway flow value obtained by serving as many flow requests as possible of the given ordering.

Likewise, for the more general problem where the network has a set GW of gateways, and each flow can be served by any gateway, a routing algorithm may start by fixing an ordering of the (node, flow demand, gateway) triplets: $\{(x, d, g) : x \in V, d \in D(x), \text{ and } g \in GW\}$, and repeatedly calling the function as mentioned above. The order of the triplets referred to above may be selected to satisfy some differentiated, or fair service criterion on the flows served from each node, and/or the total flow served by each gateway.

Tackling the more complex problem where the WMN has a number χ of radio channels available at each node can also be approached in the above conceptually simple framework. Briefly, the method entails modifying function ICFAP_Find so that, for each of the available χ channels, the input specifies the current flow vector, and the corresponding residual capacity vector. During each step in which the function tries to extend an IC-FAP by using a certain link (x, y), the function considers achieving this goal by using any of the available channels. That is, the joint assignment of flows to edges, and channels can be integrated within the IC-FAP search mechanism.

3.7 Simulation Results

The non-bifurcated routing problem considered in this chapter concerns WMNs where the transmissions of contending flows are not synchronized in time. Our approach in tackling the problem adopts a set of linear constraints that work at the level of aggregate flows, where each flow is characterized by a requested data rate. The approach of computing routes in such contention environment based on dealing with traffic aggregates raises questions on the effectiveness of the obtained routes. Additionally, the requirement of avoiding traffic splitting is expected to contribute to lower achieved throughput, compared to utilizing routing schemes that do not impose such restrictions. In this section we explore the above performance aspects. In particular, we comment on the achieved average (over all flows) gateway throughput, the minimum node throughput, and delay jitter, as described below.

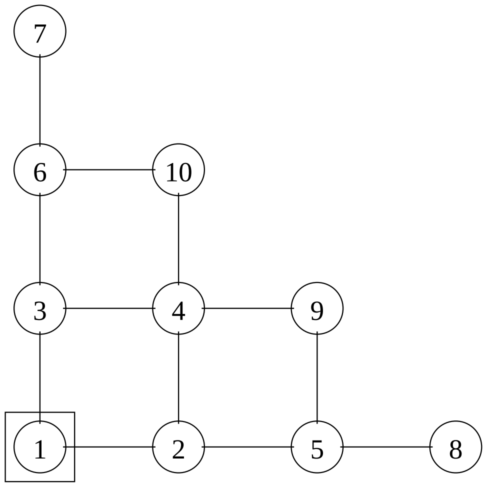


Figure 3.4: The WMN topology

3.7.1 Simulation Environment and Parameters

Performance evaluation is done by implementing two complementary, but logically distinct, software modules: a flow-based algorithm implemented in C++ for route computations, and a network layer routing algorithm that works within the framework of the QualNet 3.9.5 [48] simulator. Our implementation of the non-bifurcated routing (NBR) algorithm employs a simple round robin selection scheme for choosing a flow demand to serve; the goal is to achieve fairness between nodes by maximizing the amount of served flow from each node. Performance of the NBR algorithm is compared with that of the well-known Dynamic Source Routing (DSR) algorithm implemented in QualNet 3.9.5. We note that, in contrast to our algorithm, the operation of DSR does not impose the restrictions of non-bifurcated routing. Additionally, DSR is the basis of some proposed multi-channel routing algorithms (see, *e.g.*, [10]).

WMN Parameters	
Channel Capacity	100 units
Radio Range of Mesh Router	112 m
Radio Range of Subscriber Units	29 m
Maximum Subscriber per Router	10
Flow Demand per Subscriber	1 unit
Traffic Parameters	
1 Unit of flow	40 Kbps (Application Data)
Application-level Packet Size	Uniform: [200, 300] bytes
Packet Inter-arrival Time	Uniform: [30, 50] ms
Lower Layer Parameters	
Router Buffer Size for NBR	1000
Channel Bandwidth/Protocol	11 Mbps/ 802.11b
Data Acquisition Parameters	
Number of Runs per Data Point	8
Simulation Time of a Single Run	10 Min.

Table 3.2: Simulation Parameters

Table 3.2 summarizes the important simulation parameters. The experiments use the network in figure 3.4, where node 1 serves as the gateway. In the network, routers are placed 100 meters apart from each other, and power control is used to set their transmission range to 129 meters. Subscriber units generate traffic flows. Each unit is placed 15 meters away from its serving backhaul router, and its transmission range is set to 29 meters. The experiments are done under the stringent conditions where all traffic flows (of end users, and the backhaul routers) contend for a single 802.11 channel. Each flow unit models a 40 Kbps of application layer traffic. Packets in each flow have size uniformly distributed in the range $[200, 300]$ bytes, and inter-arrival times uniformly distributed in the range $[30, 50]$ milliseconds.

Data is gathered from multiple (typically eight) runs with each run having a simulation time of 10 minutes. The simulation time is long enough to extract stable performance from both routing algorithms.

We remark that the experimental setup described above provides challenges to both the algorithms since the interference graph of the network in figure 3.4 is moderately dense. Picking a network with denser interference graph is expected to make the performance of the DSR algorithm poor since the path discovery phase of the algorithm may compute a heavily loaded path. On the other hand, choosing a network with sparse interference graph is expected to simplify the work of both algorithms.

3.7.2 Minimum and Average Throughput

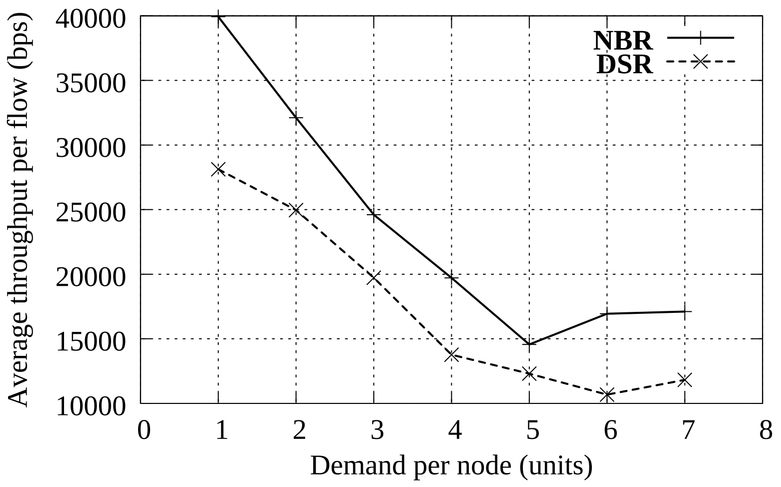


Figure 3.5: Average throughput

We first measure the average (and aggregate) throughput perceived at the gateway under different system loads by means of varying demands of each subscriber units. Figure 3.5 illustrates the achieved average throughput of all application layer flows received at the gateway. As can be seen, the NBR algorithm consistently improves on the DSR throughput by at least 20% at any input load.

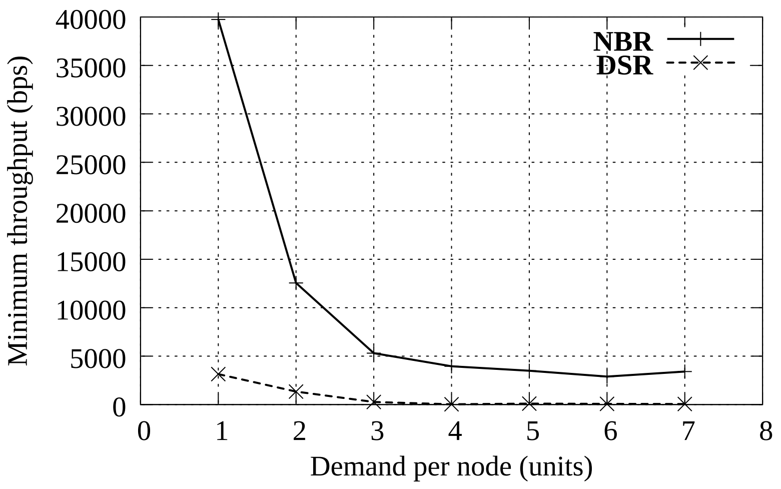


Figure 3.6: Minimum throughput

The ability to allocate bandwidth to subscribers in a controlled way, so as to satisfy the service agreements, is a central and challenging issue in the design of WMNs. To assess the ability of the NBR algorithm to meet the above requirement, the algorithm has been equipped with a simple (but imperfect) mechanism to achieve fair service among all nodes in the network, as described above. Figure 3.6 shows throughput degradation of both algorithms as the offered traffic load increases. The potential benefit of the algorithm illustrated in the figure appears in ensuring non-zero traffic throughput for such discriminated against nodes in the network. In contrast, the DSR algorithm gives zero throughput.

3.7.3 Average Delay Jitter

Delay-jitter sensitive traffic requires low delay jitter. A challenging task in the design of multi-hop wireless networks is to achieve relatively high throughputs, and simultaneously decreasing the average delay, and delay-jitter for a number of contending traffic streams. Figures 3.7, and 3.5 taken together illustrate that the NBR algorithm succeeds in improving over the DSR in both aspects simultaneously.

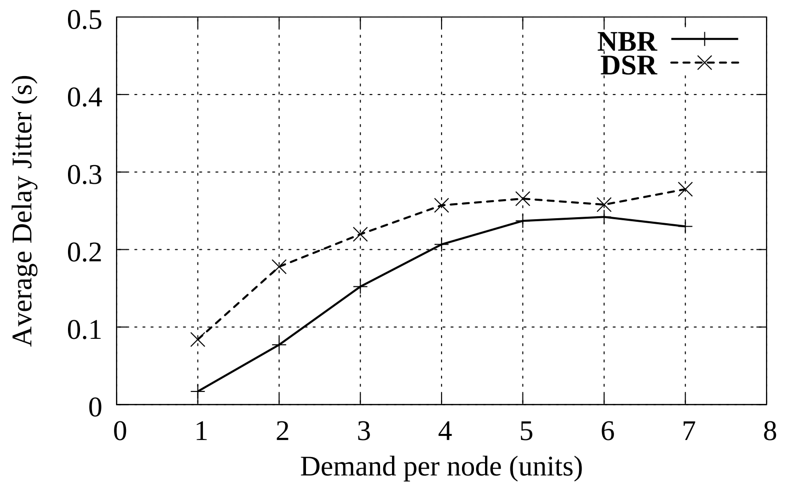


Figure 3.7: Average Delay Jitter

3.8 Concluding Remarks

In this chapter we consider the design of non-bifurcated routing algorithms for serving traffic streams in CSMA/CA-based WMNs. In such networks, each subscriber unit can have one or more streams routed to (or from) the gateway. We formulate the routing problem as a network flow problem over wireless links that are subject to CSMA/CA contention, and devise a solution based on finding interference constrained flow augmenting paths in the network. Simulation experiments show that the devised routing algorithm produces competitive results for application layer traffic.

Chapter 4

Routing and Scheduling in TDMA Grid WMNs

In this chapter we consider non-bifurcated routing in WMNs that employ TDMA. More specifically, we formalize the problem of joint routing and scheduling of flows that can be best served by non-bifurcated routing as an optimization problem. Since grid networks arise in many applications we formalize the problem for the class of grid WMNs. We then discuss the advantages of solving the problem by a strategy that attempts to route and schedule a pair of flows at each step. Subsequently we propose a dynamic programming algorithm to compute such routes. We assess the algorithm analytically and evaluate its performance by simulation. Publication [37] is based on the work presented in this chapter.

4.1 Introduction

Motivated by the results in chapter 3, we consider the joint routing and scheduling problem for traffic in multi-hop mesh networks that employ TDMA as in the IEEE 802.16 standard. Our goal is to develop schemes that achieve high throughput values while simplifying monitoring and management of flows requiring QoS guarantees (*e.g.*, delay sensitive traffic and delay-jitter sensitive traffic). Such goals can be approached by exploring single path (non-bifurcated) routing protocols. Our work here contributes to this direction by considering deployment scenarios where the topology of the network is (or can be approximated by) a grid. Our main contribution is a novel routing and scheduling algorithm that strives to allocate resources at each step for a pair of flow demands simultaneously in such grid networks. We remark that in the context of the IEEE 802.16 WMNs, flows that can benefit from non-bifurcated routing include Unsolicited Grant Service (UGS) flows and Real-time Polling Service (rtPS) flows.

4.2 System Model

As in chapter 3, in this chapter we consider WMNs with fixed mesh routers where one of the mesh routers acts as a gateway to the wired Internet. The mesh routers form a *backbone network* for backhauling traffic from individual subscribers and hotspot access points. However, unlike chapter 3, here we assume that routers are synchronized in time so as to allow data transfers between adjacent routers in well-defined time-slots. End users connect to the backbone network and the gateway through their nearest mesh router. Such backbone WMNs are envisioned to utilize multiple channels and multiple radios at each mesh router. Harnessing the available resources, however, hinges on the effectiveness of utilizing the bandwidth in each available channel. In this chapter we focus on the single channel case as a fundamental and important subproblem of the more general multi-channel case.

We assume that the local communication between subscribers or access points and their nearest mesh router is carried over a secondary wireless channel that is orthogonal to the primary wireless channel used by the backbone WMN. In this chapter we assume that the backbone network employs multi-hopping as in the mesh mode described in IEEE 802.16-2004 [20]. Communication between end users and their closest mesh router may utilize the point-to-multipoint mode.

We consider a wireless grid mesh network (WGMN) of width W units and height H units consisting of $(W + 1) \times (H + 1)$ mesh routers. Each router is assumed to be unit distance apart from its nearest neighbors. Although the grid can be rectangular, the grid cells are assumed to be square and routers are located at the corners of each square. For simplicity, the gateway is assumed to be located at the bottom left corner of the grid. Each router is assigned an (x, y)-coordinate (i, j) where $i \in [0, W]$, and $j \in [0, H]$, and the gateway is located at coordinates $(0, 0)$. We assume that all routers have the same transmission range R_T and the same interference range R_I ($R_I \geq R_T$). The transmission range R_T is adjusted so that each mesh router can directly communicate with its closest neighbors only. In such a WGMN transmission links coincide with grid lines. A transmission link between routers (i_1, j_1) and (i_2, j_2) is denoted $\langle (i_1, j_1), (i_2, j_2) \rangle$.

The interference between routers is determined by R_I. At each time-slot we require that all routers within R_I distance of a sender or a receiver of a transmission to be inactive. This requirement allows bi-directional data transfers over each link since the role of a sender and a receiver can be exchanged without affecting the link interference relations. That is, the computed routes can be used for both uplink and downlink communication with the gateway. For example, in figure 4.1(a), assuming $R_I = R_T$, transmissions over link A interfere not only with transmissions over links b, and B, but also with transmissions over links c and C.

For our purpose, we define the *cross-interference* function of two sets of links.

Definition 4.1 (Cross-interference (CI)). *The cross-interference function of two sets of links is the number of link-pairs (one link from each set) that can interfere with each other under this interference model.*

We denote this function by $f(\cdot)$. For example, in figure 4.1(a), $f(\{A\}, \{a, B, b, C, c\}) = 5$.

In a time-slotted system, a *frame* consists of a number of consecutive time-slots. A *schedule* specifies transmission activity (namely transmission, reception, or inactivity) for each link in each time slot of a frame. The number of time-slots in a frame is referred to as the *schedule length*. The schedule length and the duration of a time-slot is determined from packet size and the channel capacity. The maximum amount of data that can be transmitted over a single link during one time-slot is considered to be a *unit of flow*. Network traffic in our model is measured using such flow units.

In a WGMN each end user can request bandwidth from its nearest mesh router (to which the end user connects) for different applications. We assume a centralized resource management scheme where the mesh routers periodically forward all such requests to a central computing facility (*e.g.*, the gateway), and the facility computes routes, and bandwidth allocations in the form of a schedule. The computed results are conveyed back to the mesh routers. Such centralized resource management scheme aims at analyzing the maximum achievable throughput that the network can deliver using simpler distributed algorithms.

4.3 Problem Formulation

We consider the following throughput maximization problem: given a set of traffic flows requested by end users and traffic aggregation points, we seek to compute the maximum amount of flow that can be jointly routed and scheduled under the *route indivisibility* constraints, *interference* constraints, and *schedule length* constraint described below. The inputs and outputs of the problem are as follows:

Inputs:

- N_{frame} : The maximum allowable length of the sought after schedule.
- $D(i,j)$: A set $\{D_t(i,j) : 1 \le t \le |D(i,j)|\}$ specifying traffic demands corresponding to each flow t at router (i,j). Each demand value $D_t(i,j)$ is an integer (using our flow unit).

Outputs:

- $S(i,j)$: A set specifying the traffic demands accepted for routing at router (i,j) (*i.e.*, $S(i,j) \subseteq D(i,j)$).
- T : A table with at most N_{frame} rows that stores the computed schedule with one row for each time-slot. Row i, $i \in [1, N_{\text{frame}}]$, corresponds

to the transmission activities during the i-th time-slot in each frame. Specifically the i-th row is a list of pairs of values where each pair specifies a flow identification number and a link along which transmission of the corresponding flow takes place.

We now draw some remarks on the problem formulation:

1. The set of accepted flows $\{S(i,j) : i \in [0,W], j \in [0,H]\}$ must satisfy the following constraints:

   Flow Conservation Constraint: For any non-gateway router (i,j), the sum of outgoing flows must equal the sum of incoming flows from other routers and flows accepted from end users.

   Flow Indivisibility Constraint: Any accepted traffic demand $D_t(i,j) \in S(i,j)$ must be assigned an unsplittable route to the gateway.

   Interference Constraint: For each row in T containing a set s of transmissions, the transmissions that occur at the same time-slot corresponding to that particular row must be pairwise cross-interference free, *i.e.*, any two entries $[\mathrm{id}_1, e_1]$, $[\mathrm{id}_2, e_2] \in s$ satisfy $f(\{e_1\},\{e_2\}) = 0$.

   Schedule Length Constraint: The number of rows in table $T \leq N_{\mathrm{frame}}$.

2. The lifetime of a given flow is assumed to be long enough so that for a possible route serving the flow any conflict-free assignment of time-slots to the links of the route produces a feasible solution. That is, the allocated time-slots on the links along the route need not be in any particular order.

3. We remark that various throughput maximization problems in multi-hop wireless networks have been shown to be NP-hard (*e.g.*, see the results in [29, 47]). Such results justify the need to develop effective heuristic algorithms as pursued in this chapter.

4.4 Solution Approach

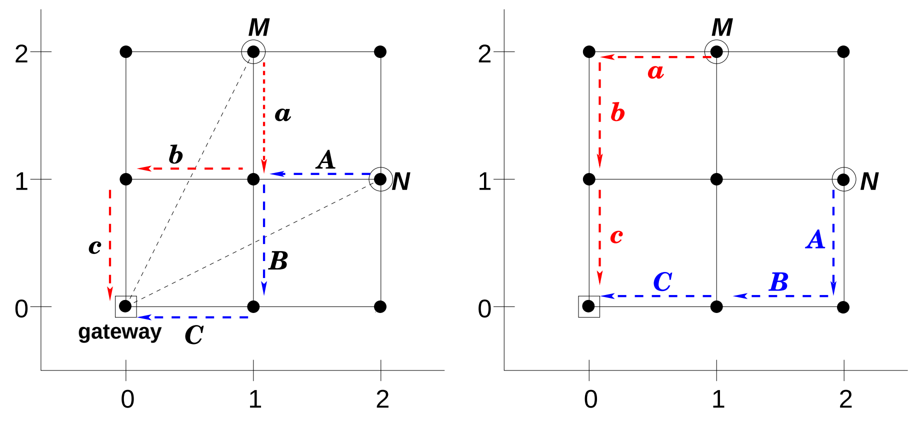
In this section we highlight a number of insights that lie behind the design of our proposed algorithm.

1. A core subproblem in our joint routing and scheduling optimization problem can be stated as follows: given a schedule T that specifies the routing and timing information for a given number of existing flows, and a new flow that is required to be routed along with the existing flows in T, can such a flow be accommodated without perturbing T? We call this problem *single flow joint routing and scheduling* (SFRS, for short) problem. We remark that the SFRS problem for arbitrary interference ranges and arbitrary routes in WMNs arbitrary topology appears to be a computationally demanding combinatorial problem. In chapter 5, we focus on SFRS problem in WMNs with arbitrary topologies for a suitable set of routes. Nevertheless the availability of an efficient solution to the SFRS problem can be used by an iterative algorithm to construct a suboptimal solution to the optimization problem described in section 4.3.

2. A generalized version of the SFRS problem calls for finding whether a given pair of flows can be added to an existing schedule T. We call this generalized problem a *flow-pair joint routing and scheduling* (FPRS, for short) problem. An important insight that underlies our work in this chapter is that using an effective solution to the FPRS problem can achieve a substantial improvement over the use of effective solution to the SFRS problem, as illustrated by the following example.

   Example 4.1. Figure 4.1(a) illustrates a 2×2 WGMN with two source routers M and N, and a gateway. Each router has a flow of one unit to send to the gateway. We require that $N_{\text{frame}} = 4$. For simplicity, we assume that $R_T = R_I$.

   A solution to the SFRS problem that chooses routers close to the straight line between a source and the gateway (*i.e.*, one of the shortest paths) may return path $A - B - C$ or $a - b - c$ in figure 4.1(a). In this case,



(a) SFRS (b) FPRS

Figure 4.1: SFRS vs. FPRS

one may verify that only one flow is admissible under the given schedule length constraint. On the other hand, figure 4.1(b) and table 4.1 illustrate that a solution to the FPRS problem can admit both flows in a schedule of length 4. This is a two-fold improvement in the throughput.

Time-slot	Transmissions
1	$[\mathrm{id}_1, a], [\mathrm{id}_2, A]$
2	$[\mathrm{id}_1, b], [\mathrm{id}_2, B]$
3	$[\mathrm{id}_1, c]$
4	$[\mathrm{id}_2, C]$

Table 4.1: Transmission schedule T

■

3. An effective solution to the FPRS problem can not only achieve higher values of the objective function but also yield schedules of shorter length, as illustrated in the following example.

   Example 4.2. In figure 4.1(a), the iterated use of an algorithm for solving SFRS may compute the two routes $A - B - C$, and $a - b - c$. One may verify that a constructed schedule that utilizes these two routes requires 6 slots. On the other hand, the two routes in figure 4.1(b) that can be computed by an algorithm for solving FPRS problem can be scheduled in 4 slots. This example demonstrates a 33% gain in schedule length resulting from solving the FPRS problem. ■

4. Designing an effective solution to the FPRS problem entails (a) finding good candidate routes for serving the given pair of flows, and (b) determining whether the transmissions along the given pair of routes are schedulable along with existing reservations in input schedule T. For the routing part in (a), our proposed algorithm considers only routes with shortest rectilinear distance to the gateway. Such routes are likely to introduce minimal interference on the neighboring routers, and hence allow more flows to be served.

   For the scheduling part in (b), our proposed algorithm assigns the most-utilized time-slot to transmission over a link without violating the interference constraint. The *most utilized first* (MUF) slot heuristic described above is likely to make the constructed schedule extensible.

5. As a WGMN offers a potentially large number of shortest path routes between a router and the gateway, an effective algorithm to solve the FPRS problem should be capable of using this rich set of routes. Our proposed algorithm uses a dynamic programming approach where the set of available routes is examined in stages. In each stage, a route pair that is considered most extensible is maintained by the algorithm. We remark that a pair of partial routes with a small CI value has a good potential for time-slot reuse, and thus such pair of routes provides a good potential for being extensible. Consequently, the algorithm views a pair of routes (r_1, r_2) with the least cross-interference value $f(r_1, r_2)$ as being the most extensible pair of routes that should be maintained for successive computations.

   Example 4.3. In figure 4.1(a), the two routes illustrated have a CI value of 9 whereas in case of the routes in figure 4.1(b), the CI value is 3. Based on the use of the CI metric, the proposed algorithm discards the pair of routes in figure 4.1(a) in favor of keeping the pair of routes in figure 4.1(b). ■

4.5 The FPRS Algorithm

In this section we outline the main steps of a dynamic programming algorithm to solve the FPRS problem (function FPRS in figure 4.2). The function takes as input a schedule T, and the (x, y)-coordinates (x_M, y_M) and (x_N, y_N) of two routers M and N. Without loss of generality we assume that $x_M \leq x_N$, *i.e.*, of the two routers, M can be considered the leftmost router. For simplicity of description, each router is assumed to be a source of a unit flow that needs to be routed to the gateway at $(0, 0)$. If the pair of flows sourced at routers M and N can be scheduled along with the existing flows in T without perturbing the existing slot assignments, then the function updates T by adding the computed routes and the corresponding slot assignments to serve the new flows.

The computations are done in stages. Graphically the stages correspond to moving a hypothetical vertical scanline from the leftmost column ($x = 0$) in the WGMN to the rightmost column corresponding to $x = x_N$. For a given position of the scanline at a particular x-coordinate x, the algorithm considers every pair of routers at the coordinates (x, y), and (x, y'). It is possible that $y = y'$. The algorithm decides whether the schedule T admits two flows sourced from these two particular routers to the gateway. If the answer is positive, then the algorithm maintains two routes r_1, r_2 with minimum cross-interference value $f(r_1, r_2)$. Computations of such pair of routes utilize routes previously computed by the algorithm.

Below we summarize the key data structures and functions utilized by the algorithm.

1. **Table** A**:** A is a three-dimensional table where the key of each entry corresponds to a triple (x, y_1, y_2) where x is the position of the vertical scanline, and (x, y_1) and (x, y_2) are two routers. The interpretation of the entry depends on the relative position of the scanline with respect to the x-coordinates of the routers M and N as follows:

    - If the scanline position $x \leq x_M$, then the value of $A[x, y_1, y_2]$ is a pair of routes from the two routers (x, y_1) and (x, y_2) respectively to the

```
Function FPRS $(T, M, N)$

  Inputs:   i) Coordinates $(x_M, y_M)$ and $(x_N, y_N)$ of the source routers $M$ and
            $N$ respectively where $x_M \le x_N$, and ii) Schedule $T$
  Outputs:  i) A pair of non-bifurcated routes in $A[x_N, y_M, y_N]$ or null (in case
            of failure), and ii) Updated $T$ if the route pair is schedulable

 1. Gateway initialization: $A[0,0,0] \leftarrow \Big([(0,0)], [(0,0)]\Big)$

 2. Leftmost scanline $(x = 0)$ initialization:
    for $(i = 0$ to $y_M$, $j = 0$ to $y_N$, and $i + j > 0)$ do
      case $i, j > 0$      : $A[0,i,j] \leftarrow \text{extend}(A[0,i-1,j-1], \Big(\text{Down}(0,i),\ \text{Down}(0,j)\Big))$
      case $i = 0, j > 0$: $A[0,i,j] \leftarrow \text{extend}(A[0,i,j-1], \Big(\emptyset,\ \text{Down}(0,j)\Big))$
      case $i > 0, j = 0$: $A[0,i,j] \leftarrow \text{extend}(A[0,i-1,j], \Big(\text{Down}(0,i), \emptyset\Big))$
    end for
 3. Bottom grid line initialization:
    for $(x = 1$ to $x_N)$ do
      case $x \in [1, x_M]$      : $A[x,0,0] \leftarrow \text{extend}(A[x-1,0,0], \Big(\text{Left}(x,0),\ \text{Left}(x,0)\Big))$
      case $x \in (x_M, x_N]$: $A[x,0,0] \leftarrow \text{extend}(A[x-1,0,0], \Big(\emptyset,\ \text{Left}(x,0)\Big))$
    end for

 4. Non-boundary entries:
    for $(x = 1$ to $x_N)$ do
      4.1 case $x \in [1, x_M]$:
          for $(i = 0$ to $y_M$, $j = 0$ to $y_N$, and $i + j > 0)$ do
            $A[x,i,j] \leftarrow \text{best-pair}\Big\{$ $\text{extend}(A[x-1,i,j], \Big(\text{Left}(x,i),\ \text{Left}(x,j)\Big))$,
                                  $\text{extend}(A[x,i-1,j], \Big(\text{Down}(x,i), \emptyset\Big))$,
                                  $\text{extend}(A[x,i,j-1], \Big(\emptyset,\ \text{Down}(x,j)\Big))$,
                                  $\text{extend}(A[x,i-1,j-1], \Big(\text{Down}(x,i),\ \text{Down}(x,j)\Big))\Big\}$
          end for
      4.2 case  $x \in (x_M, x_N]$:
          for $(j = 1$ to $y_N)$ do
            $A[x,i,j] \leftarrow \text{best-pair}\ \{$ $\text{extend}(A[x-1,y_M,j], \Big(\emptyset,\ \text{Left}(x,j)\Big))$,
                                  $\text{extend}(A[x,y_M,j-1], \Big(\emptyset,\ \text{Down}(x,j)\Big))$ $\}$

          end for
    end for

 5. If $A[x_N, y_M, y_N]$ is not empty (i.e., the pair of routes is schedulable), then
    update $T$ and return $A[x_N, y_M, y_N]$, else return null.
```

Figure 4.2: Pseudo-code of function FPRS

gateway such that on each route a unit of flow can be transmitted without violating interference constraints.

- If the scanline position $x > x_M$ then the value at $A[x, y_M, y_2]$ stores routes from routers M and (x, y_2) to the gateway such that on each route a unit of flow can be transmitted without violating interference constraints.

2. **Functions *Down* and *Left***: These two functions identifies links adjacent to a router that are horizontally to the left and vertically down with respect to the position of the router. In particular, Down(i, j) refers to the link $\langle (i, j-1), (i, j) \rangle$ and Left(i, j) refers to the link $\langle (i-1, j), (i, j) \rangle$.

3. **Function *extend*:** This function takes two parameters. The first parameter is an entry in table A (say $A[x, y_1, y_2]$) and the second one is an ordered pair of grid edges (say e_1 and e_2). The function checks whether the edges e_1 and e_2 can be appended to the first and second routes at $A[x, y_1, y_2]$, respectively, so that the transmissions over the extended routes can be scheduled along with existing transmissions in T. If so, the function returns the pair of extended routes. Otherwise the function indicates failure. In case the first (or second) route need not be extended we set $e_1 = \emptyset$ (or $e_2 = \emptyset$ respectively).

4. **Function *best-pair*:** The best-pair function takes a list of route pairs as argument, computes the CI value of each route pair, and returns the route-pair with the minimum CI value.

The overall steps of the algorithm can be described as follows: Step 1 initializes the entry $A[0, 0, 0]$ associated with the gateway to a pair of zero-length routes. Step 2 initializes boundary values $A[0, i, j]$ corresponding to the leftmost column, *i.e.*, scanline $x = 0$, by using downward links at $x = 0$. Step 3 initializes the boundary values $A[x, 0, 0]$ corresponding to the bottom row of the grid by using horizontal links at $i = j = 0$. Step 4.1 computes entries of the general form $A[x, i, j]$, $x > 0$. Step 4.2 deals with the case when the scanline has moved to the right of router M.

We remark that WGMNs are not expected to be large in size. However, they are expected to carry a large number of flows. We next show that the running time of our algorithm grows linearly with the number of carried flows.

4.5.1 Running Time

For a $W \times H$ WGMN, a single run of function FPRS requires $O(WH^2)$ entries in table A to be filled. Each entry stores a pair of routes with $O(W+H)$ links. Each entry requires a constant number of routes to be examined for extension by testing schedulability and computing pairwise-conflicts. If the number of scheduled flows in T is f_s then the number of links in T that need to be tested for conflicts is $O(f_s(W+H))$. Computation of each entry therefore requires $O(f_s(W+H)^2)$ time. The total running time to compute a schedulable route pair is therefore $O(f_s WH^2(W+H)^2)$. If the grid is almost square, and the number of routers in the grid is n then W and H both are $\Theta(\sqrt{n})$. In that case the running time of the algorithm is $O(f_s n^{5/2})$.

4.6 Remarks on Route Generation and Selection

Given two routers M and N in a WGMN, each having a flow of a certain amount to communicate with the gateway, and a schedule T of previously scheduled flows in the network, function FPRS strives to find a pair of routes that serves the flows from M and N along with the existing flows in T without changing T's slot assignments. We call such pair of routes *feasible*.

An important feature of the solution approach adopted by function FPRS is the generation and maintenance of feasible pair of partial routes that exhibit relatively small CI value. As mentioned earlier, partial route pairs that have the small CI property are more likely to admit extensions to complete routes from each of M and N to the gateway. In this section we present key observations that show the following:

a) the ability of the algorithm to generate any pair of routes from M and N to the gateway, where each route is composed of Down and Left links only, and

b) the ability of the algorithm to select a pair of routes with a small CI value (the selected pair may not be optimum in some cases).

4.6.1 Route Generation

The main result in this section is theorem 4.1 that shows that function FPRS can generate any pair of routes composed of Down and Left links from the given routers M and N to the gateway.

Notation. For ease of presentation, we denote by $R_{FP}(M,N)$ the set of all possible pairs of routes from routers M and N with coordinates (x_M, y_M) and (x_N, y_N), respectively, such that

- $x_M < x_N$, or $x_M = y_M$ and $y_M \leq y_N$, and
- each route is composed of Left and Down links only.

Theorem 4.1. *Given two routers $M = (x_M, y_M)$ and $N = (x_N, y_N)$ where $x_M < x_N$, or $x_M = x_N$ and $y_M \leq y_N$, and a valid schedule T for routing some other existing flows, the function FPRS(T, M, N) can generate any pair of routes in the set $R_{FP}(M,N)$.*

Proof. We use induction on $x_N \in [0, W+1]$.

Basis $x_N = 0$. For the case $x_N = 0$, the function FPRS(T, M, N) considers all possible routes in $R_{FP}(M,N)$ composed of Down links in step 2.

Inductive hypothesis. For any pair of routers $M' = (x_{M'}, y_{M'})$ and $N' = (x_{N'}, y_{N'})$ satisfying the following relations:

– Relation between M' and N': $x_{M'} < x_{N'}$, or $x_{M'} = x_{N'}$ and $y_{M'} \leq y_{N'}$,

– Relation between M' and M: $x_{M'} \leq x_M$ and $y_{M'} \leq y_M$, and

– Relation between N' and N: $x_{N'} < x_N$, or $x_{N'} = x_N$ and $y_{N'} < y_N$,

the function FPRS(T, M, N) considers all routes in $R_{FP}(M', N')$.

Induction step. Let r_m and r_n be any possible pair of routes in $F_{RP}(M,N)$ connecting routers M and N, respectively, to the gateway. We let $r_m =$

$e_m + r_{m'}$ and $r_n = e_n + r_{n'}$ where e_m and e_n are links incident to routers M and N, respectively, and $r_{m'}$ and $r_{n'}$ are the remaining parts of the routes r_m and r_n connecting routers M' and N', respectively, to the gateway.

Case $x_N \in [1, x_M]$: We identify the following sub-cases:

1. **Case $e_m =$ Left(M), $e_n =$ Left(N)**: The function considers such extensions using the first *extend* functions in step 3 and step 4.1 with the entry $A[x_N - 1, y_M, y_N]$.

2. **Case $e_m =$ Down(M), $e_n =$ Left(N)**: The function considers such extensions using the second *extend* function in step 4.1 with the entry $A[x_N, y_M - 1, y_N]$. Note that the entry $A[x_N, y_M - 1, y_N]$ considers the link Left(N) in a previous iteration.

3. **Case $e_m =$ Left(M), $e_n =$ Down(N)**: The function considers such extensions using the third *extend* function in step 4.1 with the entry $A[x_N, y_M, y_N - 1]$. Note that the entry $A[x_N, y_M, y_N - 1]$ considers link Left(M) in a previous iteration.

4. **Case $e_m =$ Down(M), $e_n =$ Down(N)**: The function considers such extensions using the last *extend* function in step 4.1 with the entry $A[x_N, y_M - 1, y_N - 1]$.

Case $x_N \in (x_M, W + 1]$: Similar to the above presentation, the analysis is based on the operations in step 4.2 as well as in the second extend function in step 3. ■

4.6.2 Route Selection

In this section we present observations that collectively show that route pairs generated by the algorithm enjoy the small CI property. The presentation considers the computations done in three different phases as explained below.

Phase 1. The scanline ℓ is in the leftmost position $\ell = 0$. Here, the algorithm considers all possible choices of router positions y_1 and y_2 on the vertical line $x = 0$ (*i.e.*, $y_1 \in [0, y_M]$ and $y_2 \in [0, y_N]$). For each such router, y_i, $i = 1$ or

2, the algorithm considers the unique route r_i made of vertical edges to the gateway. The algorithm keeps routes r_1 and r_2 in entry $A[\ell = 0, y_1, y_2]$ if they are feasible, otherwise the entry is null. Thus the algorithm stores a pair of feasible routes having the smallest possible CI value, if such pair exists.

Phase 2. The scanline $\ell \in [1, x_M]$. The algorithm computes entries of the form $A[\ell, y_1, y_2]$ for each possible setting of $y_1 \in [0, y_M]$ and $y_2 \in [0, y_N]$. For each such pair of values y_1 and y_2 the algorithm aims at computing two feasible partial routes r_1 and r_2, where r_1 connects router (ℓ, y_1) to the gateway, and r_2 connects router (ℓ, y_2) to the gateway. We write $r_1 = r_1' + e_1$ (similarly, $r_2 = r_2' + e_2$) where e_1 (respectively, e_2) is either a down link or a left link incident with node (ℓ, y_1) (respectively, (ℓ, y_2)). Routes r_1' and r_2' are the remaining parts of route r_1 and r_2, respectively, that have been computed in a previous step. Consequently, the pair (r_1', r_2') is assumed to enjoy the small CI property.

We say that edge e_1 lies above e_2 if $y_1 > y_2$, or if $y_1 = y_2$ and the y-coordinate of the other end vertex of e_1 is greater or equal to its counterpart of e_2. We draw the following remarks assuming that e_1 lies above e_2 (the argument is symmetric if e_2 lies above e_1):

2.a. The contribution of edges e_1 and e_2 to $f(r_1, r_2)$ is $f(e_1, r_2') + f(e_2, r_1')$.

2.b. In all cases, $f(e_1, r_2') \leq 2$, and the exact value of $f(e_1, r_2')$ is independent of the layout of r_2 in all cases except for one case (when $y_1 = y_2$).

2.c. For $f(e_2, r_1')$, we have $f(e_2, r_1') = 0$ if $y_1 \geq y_2 + 3$, independent of e_2 and r_1'.

2.d. We also observe that:
$f(e_2, r_1') \leq 1$ if $y_1 = y_2 + 2$, and
$f(e_2, r_1') \leq 3$ if $y_1 = y_2$ or $y_2 + 1$.
In these cases the exact value depends on e_2.

Phase 3. The scanline ℓ moves to the right of router M, *i.e.*, $\ell \in [x_M + 1, x_N]$. Here the algorithm utilizes entries of the form $A[\ell, y_M, y_2]$ where $y_2 \in [0, y_N]$.

For each possible value of y_2, the algorithm aims at computing a pair of routes r_1 and r_2, and storing them in entry $A[\ell, y_M, y_2]$ where r_1 is a route from M to the gateway, and r_2 is a route from (ℓ, y_2) to the gateway. We write $r_2 = r'_2 + e$ where e is either a down link or a left link incident with router (ℓ, y_2), and r'_2 is the remaining part of r_2. We remark that

3.a. For all $\ell \geq x_M + 2$, $f(e, r_1) \leq 1$ and the exact value is independent of r_1.

3.b. For $\ell = x_M + 1$, $f(e, r_1) \leq 2$. The exact value depends on r_1 only when y_2 is sufficiently close to y_M.

Thus, although at each step the algorithm computes a feasible pair of routes (if such pair exists) with the minimum encountered CI value, this value may not be optimum in some cases when the routes are required to be within the interference range of each other, as described in remarks 2.d and 3.b above. As the majority of pairs of routes that are candidates of being feasible are expected to be close to each other only in a limited number of locations, we conclude that all such pairs of routes computed by the algorithm possess a small CI value as desired.

4.7 Experimental Results

In this section we evaluate the effectiveness of our algorithm with respect to the ability to (a) maximize network throughput, and (b) construct schedules of relatively short length. We compare its performance against two commonly used algorithms that route one flow in each step. The first algorithm uses a near straight line routing (SLR) heuristic to route each flow under consideration. The second algorithm forwards packets to the neighboring router closest in Eucledian space to the gateway (denoted the CGF algorithm below).

We also compare performance against the ICFAP_Find algorithm in chapter 3 that does not require routers to be synchronized (*i.e.*, does not use time-slots). Its performance, however, is expected to be weaker than the performance of algorithms utilizing time-slotted transmissions.

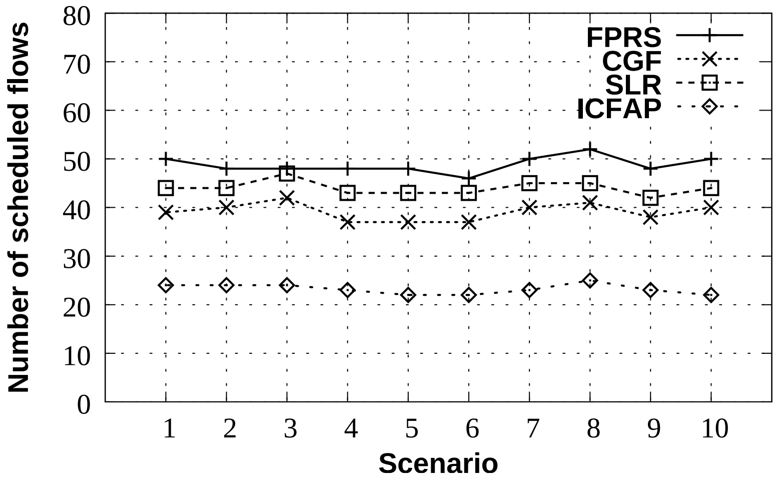


Figure 4.3: Achieved throughput

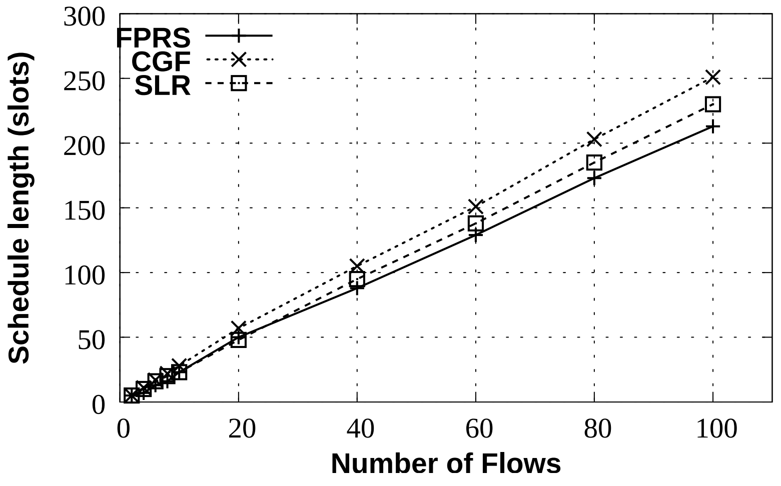
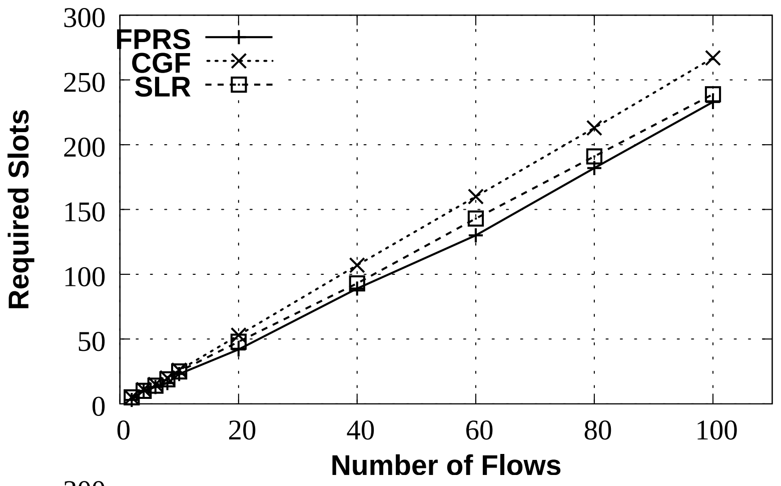


Figure 4.4: Achieved schedule length

All test cases uses a 9×7 grid network with one gateway and 62 other mesh routers. The generated traffic demand has 60 randomly generated flows, each of unit value. We remark that random generation of traffic challenges the work done by the algorithm without creating bias toward any particular algorithm. Figure 4.3 illustrates the achieved throughput (*i.e.*, the number of admitted flows) in 10 different scenarios where each scenario corresponds to a random way of generating 60 flows by the 62 mesh routers, and we seek to construct a schedule of length $N_{\text{frame}} = 100$. As can be seen, our FPRS algorithm consistently outperforms other methods (*e.g.*, by as much as 12% over SLR



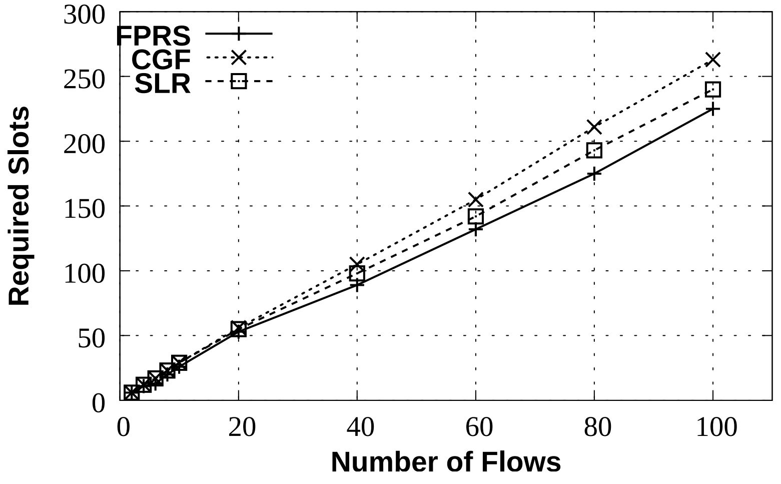


Figure 4.5: More examples of achieved schedule length

and 22% over CGF). Figure 4.4 illustrates the effectiveness of the algorithm in constructing schedules of short length (*e.g.*, SLR and CGF produces schedules that require 8% and 17% more slots respectively). Here we incrementally add a number of flows to a network so as to vary the total number of demands from 0 to 100. After each addition, we compute a schedule with minimum length that accommodates all flows. Figure 4.5 shows two more improvement scenarios for the FPRS algorithm in terms of reducing the number of required slots.

4.8 Concluding Remarks

In this chapter we consider a joint routing and scheduling problem for routing traffic in a TDMA-based WMN deployed in a grid configuration. Motivated by the advantages of non-bifurcated routing, we develop a joint non-bifurcated routing and scheduling algorithm to solve the underlying cross-layer combinatorial optimization problem. The devised algorithm acquires its strength from dealing with the combinatorics of serving pairs of flows at each step.

Chapter 5

Incremental Routing and Scheduling in TDMA WMNs

In this chapter we explore a fundamental joint routing and scheduling problem in wireless mesh networks (WMNs) that employ time division multiple access (TDMA). The problem deals with incremental update of transmission schedules necessitated by dynamic arrival of new flows and termination of existing flows during the operation of the network. In the problem, we are given a multi-hop WMN, a set of ongoing flows, a transmission schedule for the ongoing flows, a set of costs associated with links, and a new flow demand. All flows contend for using one of the available wireless channels. The problem asks for finding a non-bifurcated route with minimum cost along which the new flow can be scheduled without perturbing slot assignments in the given schedule, if such route exists. Our main contribution is an efficient algorithm for solving the problem for arbitrary interference relations among pairs of transmission links in networks with arbitrary topologies. Among other classes of routes, our algorithm is exact over the class of shortest routes. Publications [36] and [38] are based on the work presented in this chapter.

5.1 Introduction

In this chapter we consider the development of efficient exact algorithms for routing and scheduling traffic in the WMN backbone. Bandwidth allocation for network traffic requires changes over time as flows originate and terminate dynamically. Rescheduling all existing traffic carries both computation and communication overhead, and may not be possible due to QoS assurances committed earlier. In such scenarios it is practical to investigate incremental **joint** routing and scheduling algorithms.

Handling routing and scheduling in multi-hop TDMA-based WMNs has been approached in many papers by decoupling the route selection process from the scheduling process. Such decoupling can be done, *e.g.*, by choosing a spanning tree of a given WMN to perform routing during a given interval of time. Examples of recent work that use such decoupling approach include [16, 25, 34, 40, 42, 56, 58] (and many of the references cited therein). The disadvantage of such approach is that the combined solution is often an approximation of the optimal solution.

In contrast with the decoupling approaches in [2], [27], and [58] we consider both of the routing aspect and scheduling aspect jointly, and devise efficient and exact algorithms. To this end, we introduce the following problems for **incremental** routing and scheduling:

- single flow scheduling (SFS),
- joint single flow routing and scheduling (SFRS), and
- minimum cost single flow routing and scheduling (MC-SFRS).

The above problems are fundamental problems that enable the incremental update of an existing schedule to serve a new flow. The problems assume that at a given instant of time a schedule for routing some ongoing flows in a WMN is given together with a new flow demand that needs to be routed along with the ongoing flows.

For the SFS problem, in section 5.4, we derive observations on the following aspects:

a) A relation between the SFS problem and the problem of *list coloring* the nodes of the underlying conflict graph (cf. section 2.2).

b) The *treewidth* of conflict graphs that arise in solving the SFS problem.

The derived observations allow us to obtain an efficient solution to the SFS problem.

The MC-SFRS problem is a generalization of the SFRS problem where we associate a cost value of using any possible time-slot on any transmission link. The MC-SFRS problem is to find a minimum cost route along which the new flow can be scheduled without perturbing existing slot assignments in the given schedule, if such route exists. The formalized problem allows for better control of the structure of the resulting schedule. In section 5.6, we present an algorithm that solves the MC-SFRS problem without decoupling. Our algorithm is exact and efficient for certain classes of routes. In addition, our proposed algorithm produces non-bifurcated routes for arbitrary WMN topologies, and arbitrary interference relations.

5.2 System Model

As in chapter 4, we adopt a WMN model that is commonly used in the literature (*e.g.*, [2, 27]). We consider WMNs with fixed mesh routers where one of the mesh routers acts as a gateway to the wired Internet, and other mesh routers form a backbone network for backhauling traffic from individual subscribers and hotspot access points. Mesh routers can utilize multiple channels and multiple radios. We assume that routers are synchronized in time.

End users connect to the WMN through their nearest mesh router. Routers in the backbone network employ multi-hopping to carry network traffic to/from the gateway. We assume that the local communication between subscribers or access points and their nearest mesh router is carried over a secondary wireless channel that is orthogonal to the primary wireless channel(s) for the backbone WMN, and utilize the point-to-multipoint mode.

We denote transmission range as R_T and the interference range as R_I

($R_I \geq R_T$). At each time-slot we require that all routers within R_I distance of a sender or a receiver of a transmission to be inactive. This requirement allows bi-directional data transfers over each link since the role of a sender and a receiver can be exchanged without affecting the link interference relations. That is, the computed routes can be used for both uplink and downlink communication with the gateway.

Example 5.1. Throughout this chapter we make references to the network $G = (V, E_T, E_I)$ on 11 nodes and 15 links in figure 5.1. The network assumes

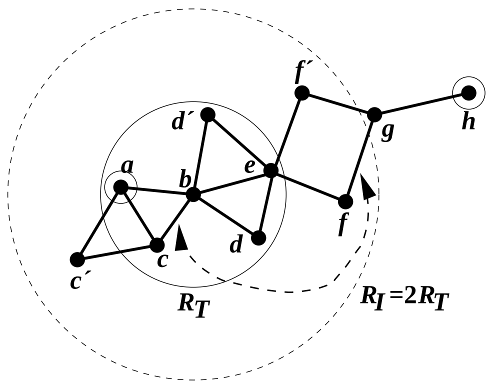


Figure 5.1: A WMN G with 11 nodes

that R_T equals the radius of the smaller circle centered at node b. In this example, we also assume that $R_I = 2R_T$. Thus the set of interference edges E_I contains all link pairs of G except the following:

– Link (g, h) does not interfere with any of the links (a, b), (b, c), (a, c), (a, c'), and (c, c').

– Link (f, g) does not interfere with links (a, c), (a, c'), and (c, c').

– Similarly, link (f', g) does not interfere with links (a, c), (a, c'), and (c, c').

■

Maximum Interference Distance

We introduce the notion of *maximum interference distance* (MID) of a route as a way of classifying routes.

Definition 5.1 (Maximum Interference Distance (MID))**.** *Given a route* $\mathcal{R} = (e_1, e_2, \ldots, e_m)$ *we define the MID of* $\mathcal{R}$ *to be the largest integer* k *such that* *transmissions over links e_i and e_j interfere with each other only when $|i-j| \leq k$.*

So, in such route two links separated by k or more links do not interfere with each other.

Example 5.2. In the network G of example 5.1, the MID of route $\mathcal{R} = ((a,b), (b,e), (e,f), (f,g), (g,h))$ is 3, since links (a,b), and (f,g) interfere with each other (the Euclidean distance between b and f is less than R_I), and yet they appear in $\mathcal{R}$ separated by 2 links. ■

We remark that

- If $\mathcal{R}$ is an induced route in G (*i.e.*, no transmission link exists between any two non-adjacent nodes in $\mathcal{R}$), and $R_I = R_T$ then the MID of $\mathcal{R}$ is 2.

- Since shortest length routes are all induced routes, it follows that shortest length routes form a subset of routes that have MID = 2.

5.3 Problem Formulation

We introduce the following fundamental problems dealing with incremental update of schedule:

- the single flow scheduling (SFS) problem,

- the single flow routing and scheduling (SFRS) problem, and

- the minimum cost single flow routing and scheduling (MC-SFRS) problem.

In the SFS problem, we are given a WMN G with some existing flows that are assigned routes and transmission slots according to a given schedule T having N_{frame} slots per frame, and a route $\mathcal{R}$ between two nodes u and v in G. We ask whether a flow of f units can be scheduled along $\mathcal{R}$ without perturbing the existing slot assignments of T. In the SFRS problem, no specific route $\mathcal{R}$ is given.

In this chapter we deal with the more general MC-SFRS problem where each link (or, each slot in each link) carries a cost. Thus, in an instance of the MC-SFRS problem, we are given

- A WMN $G = (V, E_T, E_I)$, where V is a set of nodes, E_T is a set of transmission links, and E_I is a set of interference edges between pairs of transmission links.

- A table T with at most N_{frame} slots (the number of slots in a frame) that stores a transmission schedule of existing flows in the network. Each row in T corresponds to a time-slot in the TDMA frame. The values stored in each row determines the active links during the corresponding slot.

- A cost $cost(e, c)$ associated with using slot c on link e. In case the cost metric is slot independent, we use $cost(e)$ to denote the cost of using link e.

- A flow demand of unit value between two nodes denoted u and v.

The cost of a route is the sum of the costs of slots assigned to each of its links. A route that serves a flow between nodes u and v is *feasible* if transmission on each link can be assigned a time-slot such that no two mutually interfering transmissions (either in the existing schedule T or along the route) are assigned the same time-slot. A solution to the problem is a minimum cost feasible route. As can be seen, the above problem is a fundamental problem to perform incremental updates of a schedule for handling dynamic arrival and termination of flows.

We now draw the following remarks on the problem formulation.

1. The lifetime of the given flow is assumed to be long enough so that if r is any possible route for serving the flow then any conflict-free assignment of its links to time-slots produces a feasible solution.

2. For a given link e, the formulation supports different costs for using different time-slots. This capability is intended to provide better control

on the computed schedule. Later in section 5.5 we quantify this aspect using some numerical examples.

5.4 The SFS Problem and Treewidth of Conflict Graphs

In this section we establish a relation between the SFS problem and the problem of list coloring the nodes in a graph. In addition, we recall the definition of the class of *partial k-trees* and their treewidth property. We then derive an upper bound on treewidth of the conflict graphs that arise in solving the SFS problem. The derived observations enable us to obtain an efficient solution for solving the SFS problem.

5.4.1 Conflict Graphs and List Coloring

A common approach to tackle routing and scheduling problems is to define the conflict graph of a given WMN G as a graph G_C where each node in G_C corresponds to an edge in G, and each edge in G_C corresponds to two links in G whose transmissions interfere with each other [24]. Figure 5.2 illustrates

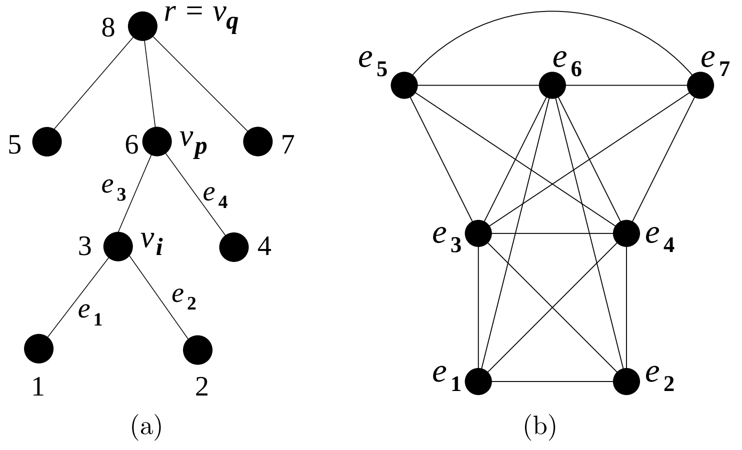


Figure 5.2: A tree T and its conflict graph $G_C(T)$

a routing tree rooted at node r, whose nodes are labeled according to *reverse level* order, *i.e.*, nodes farthest from the root r are labeled first. Assuming $R_I = R_T$, and the interference model in section 5.2 (also in section 2.2), figure 5.2(b) represents T's conflict graph.

Scheduling problems of the above type are often modeled in the literature as node (or edge) coloring problems of a conflict graph where the slots are viewed as colors (see *e.g.*, [56]). In our problem formulation, determining whether a given (u, v) route is feasible is modeled as a *list coloring* problem [28] where we associate with each link e a set, denoted *available*(e), of available time-slots (colors) that do not conflict with any time-slot (color) in the existing schedule T.

As can be seen, the complexity of coloring such a (u, v)-route depends on the interference relations between links on the route. In particular, the larger the maximum distance between two interfering links on the route (*i.e.*, the larger the MID of the route), the more work the algorithm has to perform.

5.4.2 Partial k-trees and Treewidth

In this section we recall the class of partial k-trees (also known as graphs with treewidth $\leq k$) that has received attention in literature due to development of polynomial time algorithms for solving many important NP-complete graph problems on partial k-trees with fixed k. We briefly recall the following definitions [5] pertaining to the class of k-trees for $k \geq 1$.

Definition 5.2 (k-tree)**.** *Given an integer k, $k \geq 1$, a k-tree is defined as follows:*

1. *A k-*clique *(complete graph on k nodes) is a k-tree.*
2. *If G is a k-tree on n nodes, then so is a graph G' on $n+1$ nodes obtained by adding a new node and making it adjacent to every node of an existing k-clique in G.*

The treewidth of a k-tree is k. Note that trees are k-trees with $k = 1$. A *leaf* of a k-tree is a node whose neighbors induce a clique. A partial k-tree is a subgraph of a k-tree obtained by possibly removing some edges.

Reversing the recursive construction in the above definition, one can show that a graph on n nodes is a partial k-tree by showing a *perfect elimination sequence* $v_1, v_2, \ldots, v_{n-1}$ that iteratively reduces the graph G into graphs $G =$

$G_1, G_2, G_3, \ldots, G_n$ where G_n is a single node. The perfect elimination of node v_i from graph G_i is done by possibly adding new edges to its neighbors to make its neighbors a clique of size at most k (thus v_i becomes a leaf after adding the edges), and then removing v_i. The resulting graph is denoted G_{i+1} in the above sequence.

Example 5.3. For the tree T shown in figure 5.2(a), one may verify that $G_C(T)$ in figure 5.2(b) is a 4-tree. A perfect elimination sequence for $G_C(T)$ is $e_1, e_2, \ldots, e_6$. ■

5.4.3 The Main Theorem

In this section we prove a theorem on the upper bound of the treewidth of the conflict graph of paths and trees.

Theorem 5.1. *The conflict graph of a tree $T = (V, E)$ composed of routes with MID = 2 is a partial k-tree where $k = \max_{(u,v)\in E}(deg(u) + deg(v) - 2)$.*

Proof. Choose one of the nodes with maximum degree as a root node r. Viewing T as a rooted tree, one can then fix the *parent* relationship between nodes in T with respect to the root node r. Let $v_1, v_2, \ldots, v_n$ be the nodes of T according to reverse level order traversal of T (thus v_1 is a leaf of T). Let G be the conflict graph of T where each vertex $e_i = (v_i, parent(v_i))$. We show that the sequence $e_1, e_2, \ldots, e_{n-2}$ is a perfect elimination sequence of G that results in the sequence of graphs $G_1, G_2, \ldots, G_{n-1}$ where $G = G_1$ and each vertex e_i has at most degree k in G_i. To see this, note that the following holds in G_i, $1 \leq i \leq n-1$:

(i) all vertices in the prefix $(e_1, e_2, \ldots, e_{i-1})$ have been eliminated without adding any new edge to the original graph G, and

(ii) the neighbors of vertex e_i in G_i correspond to undeleted edges in T incident with $v_p = parent(v_i)$ and (if v_q exists) $v_q = parent(v_p)$. For example, in figure 5.2(a), if we take node 3 as v_i, then $v_p = parent(v_i)$ is node 6, and $v_q = parent(v_p)$ is the root node 8. The neighbors of edge $e_i = e_3$ in figure 5.2(b) after perfectly eliminating e_1 and e_2 are nodes

$e_4, e_5, \ldots, e_7$ that correspond to a subset of the tree edges incident to v_p and v_q.

Thus the degree of e_i in G_i is at most $deg(v_p) + deg(v_q) - 2$. ■

In the context of solving the SFS problem, theorem 5.1 implies that if $\mathcal{R}$ is a route with MID = 2 then its conflict graph is a 2-tree. As mentioned earlier, the observation enables us to design a linear time algorithm for solving the SFS problem. However, we do not describe the solution since we present an efficient solution to the more general MC-SFRS problem in section 5.6.

5.5 Performance Benefits

We recall that the MC-SFRS problem is a generalization of the SFRS problem by associating a cost with allocating a time-slot on a transmission link. We now present examples that illustrate the benefits of solving the SFRS and MC-SFRS problems in maximizing network throughput (examples 5.4 and 5.5) and shortening the schedule length (examples 5.6 and 5.7) over the competing technique of using a fixed routing tree.

Example 5.4. To show that routes produced by solving the SFRS problem improves throughput over routes of a fixed tree T, we consider first the 3×3 grid network of figure 5.3. Assume that the desired schedule length has 5 slots, the network initially has no flows, and the routing tree of figure 5.3(a) is used in the comparison. In addition, assume that two unit flows between node u and the gateway need to be served. Using T, only one flow can be served

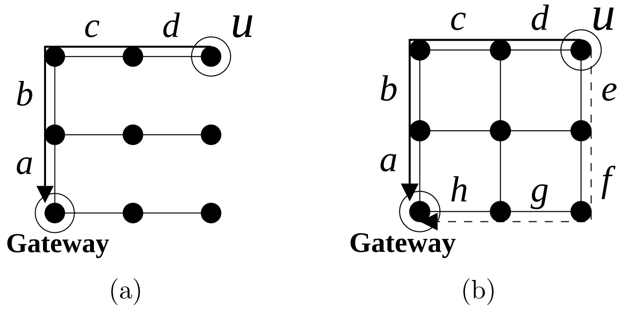


Figure 5.3: Scheduled Flows over Tree and Grid

Slot	Link
1	a, d
2	b
3	c
4	-
5	-

Table 5.1: Schedule for 1 flow in tree

Slot	Link
1	a, d
2	b, e
3	c, f
4	g
5	h

Table 5.2: Schedule for 2 flows in grid

(table 5.1 illustrates the resulting schedule). In comparison, solving the SFRS problem twice produces routes that accommodate the two flows (table 5.2 illustrates the resulting schedule). ■

Example 5.5. This example shows that iterated use of an algorithm that solves the MC-SFRS problem can produce a schedule that improves throughput in a non-grid WMN over a schedule produced by using a spanning tree T. Figures 5.4(a) and 5.4(b) illustrate a 7 node WMN and an associated breadth first search (BFS) tree T rooted at the gateway, respectively. Here, $R_I = 1.5R_T$ (so, *e.g.*, link A interferes only with links B, C, a, and b). We assume that the desired schedule length is 5 slots. The network initially has no flows. Two unit-valued flow requests are required to be routed from node u to the gateway.

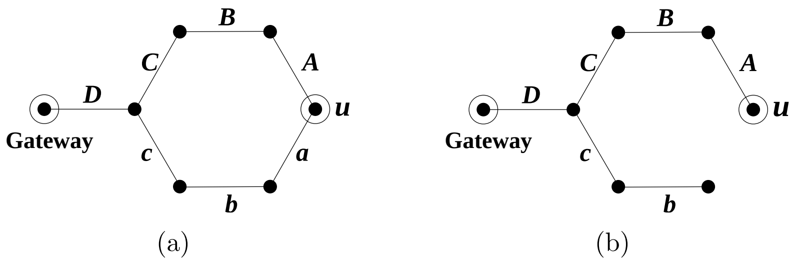


Figure 5.4: (a) A WMN backbone and (b) a BFS routing tree T

We observe that using the fixed routing tree T, only one flow from u can be sent (table 5.3 shows a possible schedule). In contrast, solving the MC-SFRS problem twice for the two flows, both flows can be accommodated in a single frame. The corresponding schedule is shown in table 5.4. Thus, solving the MC-SFRS problem yields a two-fold improvement in throughput in this example. ■

Slot	Link
1	A, D
2	B
3	C
4	-
5	-

Table 5.3: Schedule using routing tree

Slot	Link
1	A, D
2	B, b
3	C, a
4	c
5	D

Table 5.4: Schedule using SFRS

Example 5.6. To show that routes produced by solving the SFRS problem can result in a shorter schedule length than routes of a fixed tree T, consider the 4×4 grid of figure 5.5. Assume that the network initially has two unit flows from node v and w to the gateway node, and the routing tree T of figure 5.5(a) is used for comparison. In addition, assume a new unit flow from u to the gateway need to be routed. Using T, one obtains the schedule in

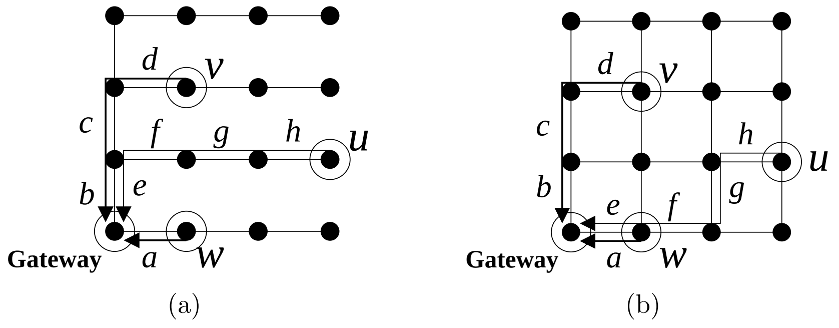


Figure 5.5: Routes over Tree and Grid

Slot	Link
1	a, d, h
2	b
3	c
4	e
5	f
6	g

Table 5.5: Schedule for tree

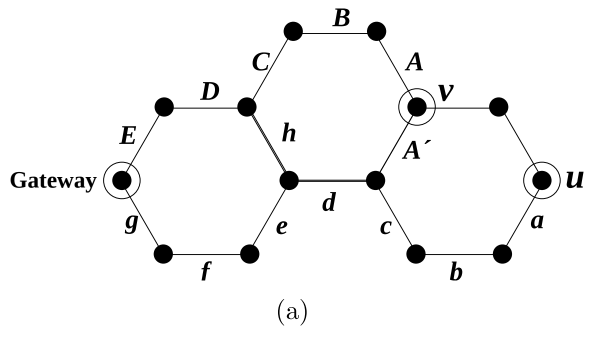
Slot	Link
1	a, d, h
2	b, g
3	c, f
4	e
5	-
6	-

Table 5.6: Schedule for grid

table 5.5 of length 6. In comparison, solving the SFRS problem for each flow produces the schedule in table 5.6 of length 4. ■

Example 5.7. This example shows that iterated application of an algorithm that solves the MC-SFRS problem can be controlled to produce a schedule with larger number of unused slots in a non-grid WMN than a schedule obtained by using a routing tree T.

Figures 5.6(a) and 5.6(b) illustrate a WMN and an associated routing tree T. In this example we assume $R_I = 1.75R_T$. So, *e.g.*, in figure 5.6(a) link g interferes with all remaining links in its hexagon (*i.e.*, links f, e, h, D, E), and also links C, and d. We assume a schedule length of 8 slots. The network initially has a unit flow routed from u along the route (a, b, c, d, e, f, g). This flow occupies 4 slots in a schedule (see tables 5.1 or 5.2). Using the slot assignments of this flow, we set the cost of using any of the already used slots (1 through 4) to 1, and the cost of using any of the unused slots (5 through 8) to 2, so as to discourage the use of the unused slots. A new flow demand of unit value needs to be routed from node v to the gateway.



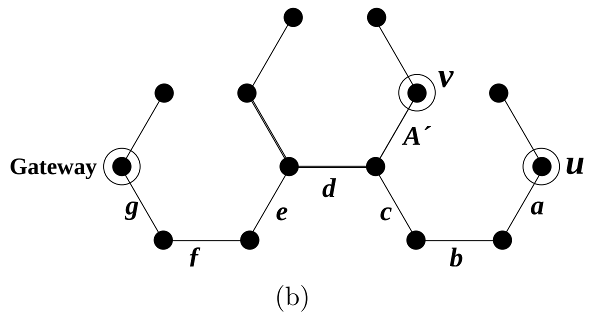


Figure 5.6: (a) A WMN and (b) a routing tree T

In case of the routing tree T, the flow follows the unique route (A', d, e, f, g). We observe that using T, the two flows occupy all 8 slots. Table 5.7 shows a possible schedule for this case. In contrast, solving the MC-SFRS problem, where the cost metric is the total number of slots required, can produce a schedule with larger number of unused slots. Table 5.8 shows a

Slot	Link
1	a, e
2	b, f
3	c, g
4	d
5	A', g
6	d
7	e
8	f

Table 5.7: Schedule using routing tree

Slot	Link
1	a, e
2	b, f, B
3	c, g
4	d
5	A, E
6	C
7	D
8	-

Table 5.8: Schedule using MC-SFRS

possible schedule corresponding to this case where the new flow is assigned the route (A, B, C, D, E). Thus, there is an opportunity of obtaining schedules of shorter length by solving the MC-SFRS problem. ■

5.6 The MC-SFRS Algorithm

In this section we present an algorithm for solving the MC-SFRS problem for classes of routes characterized by a given maximum interference distance (MID) value. Our algorithm is exact when applied to any such class of routes including the set of shortest routes. Our presentation below emphasizes the following aspects:

(a) The usage of a total order relation defined over a subset of nodes of G. The relation is used in generating the routes considered in solving the optimization problem (section 5.6.1).

(b) An algorithm for computing an upper bound on the MID of the set of routes considered by the algorithm. The computed upper bound serves as the parameter k used in the algorithm (section 5.6.2).

(c) An overview of the main algorithm (section 5.6.3).

(d) The concept of state vectors used by the dynamic program in the main algorithm (section 5.6.4).

5.6.1 Node Ordering

The main algorithm considers a particular set of (u, v)-routes in computing an optimal solution to the problem. The structure of such set of routes is controlled by selecting a permutation π that defines a total order relation over a subset of nodes in the given WMN G, and a subset E'_T of transmission links of G. The resulting set of routes is denoted $R(\pi, E'_T)$. Both of the two parameters π and E'_T can be changed to explore different sets of routes.

For our purpose, the selected permutation $\pi = (\pi_1, \pi_2, \pi_3, \ldots, \pi_{n'})$ is any permutation on any subset of n' nodes, $n' \geq 2$, in G that includes nodes u and v where nodes $\pi_1 = u$ and $\pi_{n'} = v$. In addition, E'_T is selected to be any set of links between the nodes in π. A route is *valid* (for π and E'_T) only if its sequence of nodes is a subsequence of π (*i.e.*, if $(\pi_i, \pi_{i'})$ is a link in a route then $i \leq i'$), and all its links exist in E'_T. The set $R(\pi, E'_T)$ is the set of all valid (u, v)-routes.

Example 5.8. For the network G of example 5.1, suppose that we want to route a flow between nodes a to h. In addition, suppose we choose π to be the order of nodes encountered in a possible BFS traversal of G from a to h, say *e.g.*, $\pi = (a, c, b, c', d, d', e, f', f, g, h)$. Furthermore, suppose we set E'_T to the subset of links where each link has its two end nodes in π except that we omit *cross* links encountered in the BFS traversal (*i.e.*, links (b, c), (c, c'), (d, e), and (d', e)). Then $R(\pi, E'_T)$ is exactly the set of all shortest length routes between nodes a and h. ■

Generalizing the above example, we remark that if the permutation π is the BFS ordering of nodes in T starting at node u and ending at node v, and E'_T is the subset of links where each link has its two end nodes in π except *cross* links encountered in the BFS traversal, then the set $R(\pi, E'_T)$ of routes corresponds to a subset of induced routes from u to v in G. By the definition of induced routes, each such (u, v)-route has a minimal set of nodes. Thus, using such set of routes guarantees finding a shortest length route, if one exists, as claimed above.

Notation. For simplicity of presenting the details of the algorithm we henceforth refer to nodes π_1, π_2, ..., $\pi_{n'}$ simply as $1, 2, \ldots, n'$. So, any such node label x lies in the range $[1, n']$.

5.6.2 Maximum Interference Distance

We now discuss a simple method for computing an upper bound on the MID of a given set $R(\pi, E'_T)$ of routes. The computed value is used in the main algorithm. To this end, we introduce the following notation using the new labels $1, 2, \ldots, n'$ fixed above for the nodes in π:

- Denote by E'_I the set of possible interference edges between pairs of transmission links in E'_T.
- Let $e_I \in E'_I$ be an interference edge between two links $e_i = (i, i')$, $e_j = (j, j') \in E'_T$ that appear on at least one valid route $\mathcal{R}$ (thus, $i < i' \leq j < j'$ in π).
- Denote by $d_I(e_I, \pi)$ the maximum number of links separating e_i and e_j on any such valid route $\mathcal{R}$.
- For any node $x \in [1, n']$, let

  $d_{\max}(x)$ = the maximum number of links in any valid route between nodes 1 and x.

  $d_{\min}(x)$ = the minimum number of links in any valid route between nodes 1 and x.

Example 5.9. For the network G of example 5.1, suppose we want to route a flow $f(a, h)$, and we choose $\pi = (a,\ c,\ b,\ c',\ d,\ d',\ e,\ f',\ f,\ g,\ h)$ and E'_T to include all links whose endpoints exist in π. Then we get the following values.

Node	a	c	b	c'	d	d'	e	f'	f	g	h
$d_{\min}$	0	1	1	1	2	2	2	3	3	4	5
$d_{\max}$	0	1	2	2	3	3	4	5	5	6	7

Table 5.9: Calculating $d_{\max}$ and $d_{\min}$

One may verify that if $e_I = ((a,b),(f,g))$ then $d_I(e_I,\pi) = 3$ (see, *e.g.*, the route $\mathcal{R}$ of example 5.2). ■

Our main observation here is that arrays $d_{\max}(1,\ldots,n')$ and $d_{\min}(1,\ldots,n')$ can be computed in $O(|E'_T|)$ time by an iterative algorithm that processes the nodes in the increasing order $(1,2,\ldots,n')$. The algorithm initializes $d_{\max}(1) = d_{\min}(1) = 0$. Subsequently, it processes node x, $x \geq 2$, by computing:

- $d_{\max}(x) = 1 + \max\{d_{\max}(w) : w < x, \text{ and } (w,x) \in E'_T\}$, and
- $d_{\min}(x) = 1 + \min\{d_{\min}(w) : w < x, \text{ and } (w,x) \in E'_T\}$.

We also observe that the following inequality for $e_I = ((i,i'),(j,j'))$ gives an upper bound on $d_I(e_I,\pi)$:

$$d_I(e_I,\pi) \leq d_{\max}(j) - d_{\min}(i').$$

In example 5.9, $d_I(e_I,\pi) = 3 \leq d_{\max}(f) - d_{\min}(b) = 5 - 1 = 4$. By computing such an upper bound value for each edge in E'_I, and taking the maximum value, we obtain an upper bound on the value MID – 1 of $R(\pi, E'_T)$.

5.6.3 Overview of the Main Algorithm

The main algorithm illustrated in figure 5.7 takes as input a permutation π that induces the labeling $1,2,\ldots n'$ fixed in section 5.6.1, a subset E'_T of transmission links, link cost information, a schedule T for serving existing flows in the network, and the MID (or an upper bound) of $R(\pi, E'_T)$ denoted k. The algorithm computes a minimum cost feasible route between u and v.

The main loop of the algorithm processes the nodes in the order (1, 2, 3, …, n'). In addition, we associate with each such node x, $x \in [1,n']$, a separate array R_x for storing intermediate results. Each array R_x provides a mapping from *state vectors* to (route, route cost) pairs of values. At the x-th step, $x \geq 2$, the main loop of the algorithm in step 2 starts with precomputed values stored in arrays $R_1, R_2, R_3, \ldots, R_{x-1}$. The step then considers all links of the form $(w,x) \in E'_T$ where $w < x$. Each such link is considered as a candidate for extending each of the routes stored in R_w. If the decision is to consider such extended route for subsequent iterations, then a state vector summarizing the important properties of such route is computed and used as a key value for storing the extended route in array R_x.

5.6.4 State Vectors

Suppose that $r = (x_1, \ldots, x_i)$ is a partial route in the set $R(\pi, E'_T)$ where x_1 is the node u (so, $x_1 = 1$), and each link in r is assigned a transmission slot. We define the state vector associated with the partial route and its associated slot assignment as follows. First, write the links of r as encountered in a traversal from x_i to x_1 as $e_1 = (x_i, x_{i-1})$, $e_2 = (x_{i-1}, x_{i-2})$, *etc.*, and denote the slots assigned to these links as $c_1, c_2, \ldots, c_{i-1}$, respectively. If the route has k or more links (*i.e.*, $i > k$) where k is the MID of the set $R(\pi, E'_T)$ of routes considered by the algorithm then the corresponding state vector is $(c_1, c_2, c_3, \ldots, c_k)$.

For routes that have fewer than k links, the missing slots in the corresponding vector are represented by a special value denoted $\varnothing$ which does not correspond to any valid slot. So, any state vector has k components, regardless of the length of the route with which it is associated.

The intuitions behind the above definition are:

(a) any partial valid route from node 1 to some node x in π (with an associated slot assignment) maps to a unique state vector, and

(b) while many potentially different partial routes with their associated slot assignments may map to the same state vector, all such routes admit similar extensions (edges and slot assignments) to produce a route from u to v.

Hence, for a given state vector and a given node x in π, it suffices to store a least cost partial route from node 1 to node x that is associated with this vector, and ignore the remaining partial routes from node 1 to node x. The algorithm stores such a representative partial route in step 2.4.

5.6.5 Correctness and Running Time Analysis

One can use induction to prove the following theorem.

```
ALGORITHM MC-SFRS($\pi, E'_T, cost, T, k, u, v$)

Input: as described above
Output: a minimum cost feasible $(u,v)$-route (if one exists) in the set
$R(\pi, E'_T)$ of routes

1. Initialize $R_1[\varnothing, \ldots \varnothing] = ((u), 0)$. That is, $R_1$ stores the trivial one-node
   route $(u)$ with cost zero.

2. for $(x = 2, \ldots, n')$ do
     2.1 Initialize $R_x$ to empty
     2.2 for each link $(w, x)$ in $E'_T$ where $w < x$ do
         /* Check whether link $(w, x)$ can be profitably added to some
         route in $R_w$ */
         2.3 for each (key $(c_1, \ldots, c_k)$ in $R_w$) and (slot $c$ in $available(w, x)$)
             do
               /* Denote the (route, cost) pair stored in $R_x(c_1, \ldots, c_k)$ by
               $(r, cost(r))$ */
               2.4 if link $(w, x)$ can be scheduled to transmit in slot $c$
                   without conflicting with slots $c_1, \ldots, c_k$, and $cost(r)$
                   + $cost((w, x), c)$ is best possible for the entry
                   $R_x(c, c_1, \ldots, c_{k-1})$ then store the extended route
                   and its associated cost in $R_x(c, c_1, \ldots, c_{k-1})$.
             end for
         end for
   end for

3. if ($R_v$ is non-empty) then return the least cost route in $R_v$ with its
   associated schedule.
```

Figure 5.7: Pseudocode for algorithm MC-SFRS

Theorem 5.2. *Algorithm MC-SFRS computes a least cost feasible (u, v)-route in the set $R(\pi, E'_T)$ (if such route exists).*

Proof. We induct on the label x for the node processed in step 1 (for node $x = 1$) or step 2 (for nodes $x \in [2, n']$) to show that at the end of the processing step the algorithm computes in table R_x a least cost valid route from node 1 to x for each possible state vector associated with any such valid route.

The basis for $x = 1$ holds since step 1 initializes R_1 with the empty route containing node u (node 1) and having cost zero.

For $x > 1$, we assume that the algorithm satisfies the induction hypothesis for every node in the range $[1, x-1]$ before the start of iteration x. For each possible edge (w, x), where $w < x$ and $(w, x) \in E'_T$, the algorithm exhausts all possibilities of extending least cost partial routes stored in R_w by adding the link (w, x), and stores a least cost route associated with the resulting state vector in table R_x. ■

Running Time. Let M be the number of links in the subset E'_T of transmission links, N_{frame} be the number of time-slots in the TDMA frame, and k be the MID for the set $R(\pi, E'_T)$ of routes. Each iteration of step 2.3 requires $O(N_{\text{frame}}^{k+1})$ time, and the total number of iterations is $O(M)$. Thus the running time of the algorithm is $O(MN_{\text{frame}}^{k+1})$. ■

We remark that the algorithm can be extended to handle routing multi-unit flows using the same route (*i.e.*, non-bifurcated routing). In addition, the algorithm admits speedup in cases of equal (or slot independent) cost values, and when the given schedule T is sufficiently sparse.

5.7 Experimental Results

In this section we present experimental results to show the effectiveness of our MC-SFRS algorithm. We compare our exact algorithm with scheduling over a fixed routing tree. Many existing research work consider using a BFS tree T rooted at a gateway (see, *e.g.*, [25], [58]). However, the exact scheduling algorithm in [25] assumes restriction on interference. In this section we show that, in comparison with such tree-based routing, our MC-SFRS algorithm is able to

(a) route more flows between a gateway and other nodes,

(b) route more flows between arbitrary nodes,

(c) route additional flows when a routing tree becomes saturated.

We use the 19 node example network topology of figure 5.8(a), and the routing tree in figure 5.8(b). We also use a 9×9 grid, and one of its BFS trees, where the gateway is located at one of the corners of the grid. We consider

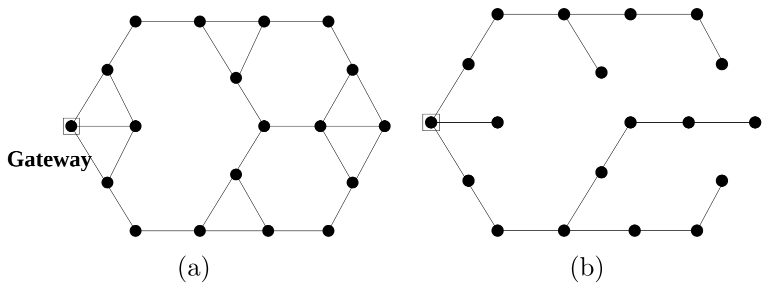


Figure 5.8: (a) Example topology and (b) its BFS routing tree T

randomly generated flow requests. In the example topology, we consider flow requests from nodes more than 2 hops away from the gateway (non-uniform distribution), and in the grid, we consider flow requests from all nodes (uniform distribution). All flow requests are of unit value, and all links are assumed to have equal costs for each time-slot.

We remark that the choice of topology and traffic pattern in experiments does not create bias toward the MC-SFRS algorithm. For any topology, the set of routes determined by the algorithm contains all the routes in a BFS tree, and therefore the algorithm performs at least as good as the BFS tree. The experimental setup for the topology in figure 5.8 aims for the algorithm to exercise search for alternative paths that exist for traffic sent from a node that lies more than 2 hops away from the gateway.

Traffic to Gateway: In figure 5.9, we consider traffic between mesh routers and the gateway for both the 19 node example topology with $R_I = 1.75R_T$ (plots Topology and Topo-tree), and the 9×9 grid with $R_I = 2R_T$ (plots Grid and Grid-BFST). The length of the required schedule is fixed in each experiment to a value in the range $[20, 80]$. For each schedule length, we consider 100 sets of random sequence of flow requests from the routers to the gateway. We count the number flows scheduled until a flow request cannot be served. Since the gateway is a common bottleneck for all flow requests, irrespective of the set of routes under consideration, the first failure in routing a flow request is an indication of imminent saturation of the network. In figure 5.9, for each schedule length considered, we plot the average and standard deviation of throughput over the same sets of flow requests. We observe that the

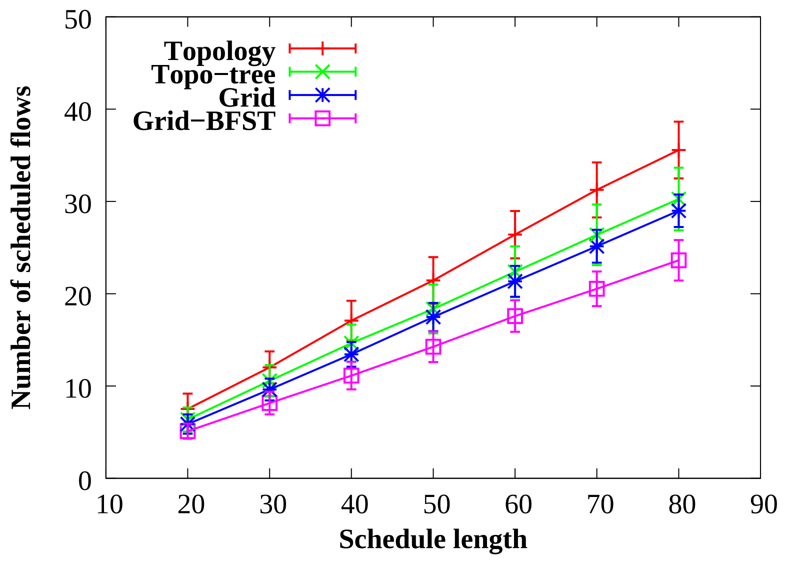


Figure 5.9: Throughput comparison for gateway traffic

throughput for MC-SFRS algorithm is higher than the tree-based counterpart.

Inter-router Traffic: In the second set of experiments, we run our algorithm using the same range of schedule lengths as in the first set of experiments.

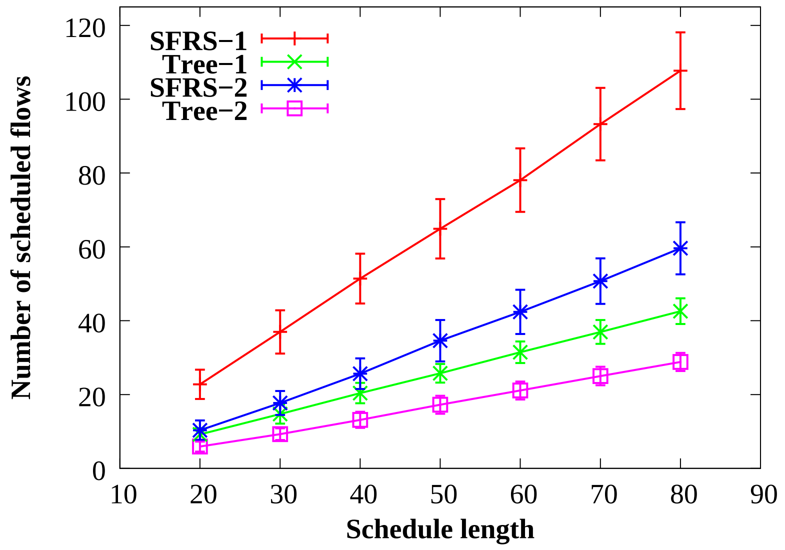


Figure 5.10: Throughput comparison for inter-router traffic

For each schedule length, we consider 100 sets of random sequence of flow requests between non-gateway routers. We compare our MC-SFRS algorithm with tree-based routing for both $R_I = R_T$, and $R_I = 2R_T$ (plots with suffix 1 and 2 respectively in figure 5.10). We observe the number of scheduled flows until a request in the sequence cannot be served. In contrast with the scenarios involving gateway traffic, there is no single bottleneck in the network. Thus, the first failure to route a flow request in this case is indicative of impending saturation only for the case of tree-based routing. In figure 5.10, the MC-SFRS algorithm shows improvement in throughput than the tree-based algorithm for both cases considered.

Additional Traffic over Tree-based Routing: In this experiment we consider flows in the grid topology using two interference ranges $R_I = R_T$, and $R_I = 2R_T$ (plots with suffix 1 and 2 respectively in figure 5.11). We first

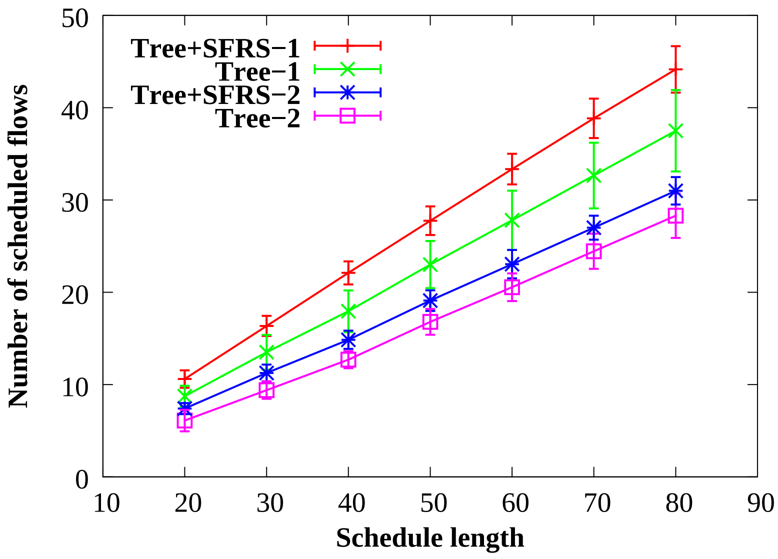


Figure 5.11: Augmenting flows with MC-SFRS

saturate the BFS tree with flow requests. Then we run our MC-SFRS algorithm for handling the flow demands that failed to be scheduled using the routing tree T. For each schedule length we consider 20 sets of random flow requests destined to the gateway. We find that there are opportunities for routing additional flows for both interference ranges considered (figure 5.11). Such opportunities of routing additional flows arise since the MC-SFRS algorithm is able to schedule more transmissions along non-tree links. Based on this finding, one may opt to use a two-level scheduler where the first level employs a simple tree routing algorithm, and the second level employs a more sophisticated MC-SFRS algorithm. The second level is invoked only when the first level fails to satisfy a given flow.

5.8 Concluding Remarks

In this chapter we consider the SFS, SFRS, and MC-SFRS problems in TDMA-based WMNs. These problems are fundamental problems in incremental routing and scheduling of a flow.

We identify the relationship between the SFS problem and list coloring, and derive an upper bound on treewidth of conflict graphs that arise in solving the problem. The observations allow us to design an efficient solution to the SFS problem.

We then present examples that demonstrate the advantages of solving the SFRS and MC-SFRS problems. Since MC-SFRS is a generalization of the SFRS problem, we develop an efficient solution for the MC-SFRS problem. To this end, we present a flexible parameterized way of defining several classes of routes that can be used in solving the problem. The important class of shortest length routes is a special class of the defined route classes. We develop an exact efficient algorithm for solving the MC-SFRS problem over any such route class. In addition, our algorithm can handle networks with arbitrary topologies and interference relations between pairs of links in the network (*i.e.*, cases where $R_I \geq R_T$).

Chapter 6

Conclusion and Future Work

6.1 Summary

The development and proliferation of WMNs as a key technology for BWA has triggered research interest in both academic and industrial setting. Many research work aim at improving the performance of WMNs using a number of methods. In this thesis, we investigate and formalize routing and scheduling problems in WMNs for QoS provisioning while serving demands of the end users.

We consider routing and scheduling problems in two types of WMNs. In WMNs of the first type, mesh routers employ CSMA/CA protocol, and hence the routers are not required to be synchronized in time. In WMNs of the second type, routers are synchronized in time and employ TDMA. In chapter 3, we formalize a throughput maximization problem in the **CSMA/CA-based WMNs**. We utilize concepts from the theory of network flows to route integer valued flows, and develop a heuristic routing algorithm that discovers IC-FAPs for routing new flow demands in a non-bifurcated manner in such WMNs. Iterative application of the algorithm yields improvement in terms of throughput and delay-jitter over the widely referenced DSR protocol for multi-hop wireless networks that incorporates a number of optional features.

Motivated by the improved support for jitter-sensitive traffic through non-bifurcated routing, in chapter 4, we investigate joint non-bifurcated routing and scheduling algorithm in **TDMA** WMNs deployed in a **grid** configuration. We devise a heuristic algorithm that tries to route a pair of flows to (or from) a gateway in each iteration with the goal of improving network throughput.

In chapter 5, we investigate interference-aware dynamic routing and scheduling in **TDMA** WMNs with **arbitrary topologies**. In particular, we consider a general incremental joint routing and scheduling problem that deals with updates of schedule and routes necessitated by arrival of new flows and termination of some of the existing flows. We characterize routes using interference relation among links using the theory of graphs with bounded treewidth, and present an algorithm for scheduling a new flow along a non-bifurcated route without perturbing the slot assignments of existing flows.

6.2 Future Work

The success and popularity of WMNs in recent years motivate further research to improve their architecture, and to better harness new advances in wireless communication. The development in WMN technology now enables serving mobile end-users with high data rate. WiMAX (Worldwide Interoperability for Microwave Access) and LTE (Long Term Evolution) are two types of architectures envisioned for serving mobile users requiring both Internet connectivity and high bandwidth multi-media services. A significant number of research work focus on different aspects of serving mobile users in WMNs. As future work, we consider the research direction of supporting user mobility with QoS assurance. To this end, we consider non-bifurcated routing as one possible method of QoS provisioning for mobile end-users. Non-bifurcated routing with mobility support in WMNs poses a challenging set of research problems, *e.g.*, reducing handoff latency and call admission control for providing QoS assurance. The design of centralized algorithms can be a possible approach to tackle such problems. However, in some scenarios, the performance of the centralized algorithms can be inadequate for the real-time services required by mobile users. Consequently, we also consider the development of distributed algorithms for routing and scheduling to serve the requirements of such real-time applications as future work.

Supporting mobility in WMNs also introduce challenges in a number of related research problems including router placement, and gateway selection in multi-gateway networks. Research in this area entails estimating user requirements in a geographic region, and determining placement of WMN routers and gateways for serving the users in the best possible way while keeping the deployment cost low. We envision adequate modeling of the problem and development of a solution framework as a possible future research work that is significant from both theoretical and practical points of view.

Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang. Wireless mesh networks: a survey. *Computer Networks and ISDN Systems*, 47(4):445–487, 2005.

[2] M. Alicherry, R. Bhatia, and L. Li. Joint channel assignment and routing for throughput optimization in multiradio wireless mesh networks. *IEEE Journal on Selected Areas in Communications*, 24(11):1960 – 1971, November 2006.

[3] Y. Bajerano, S.-J. Han, and A. Kumar. Efficient load-balancing routing for wireless mesh networks. *Computer Networks*, 51(10):2450–2466, July 2007.

[4] J. Bennett and H. Zhang. WF^{2}Q: worst-case fair weighted fair queueing. In *INFOCOM '96. Proceedings of the Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation.*, pages 120–128, March 1996.

[5] A. Brendstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: A Survey.* SIAM, 1999.

[6] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*, chapter 26, pages 643–700. The MIT Press, 2nd edition, 2001.

[7] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris. A high-throughput path metric for multi-hop wireless routing. In *MobiCom '03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 134–146, 2003.

[8] A. Demers, S. Keshav, and S. Shenker. Analysis and simulation of a fair queueing algorithm. In *SIGCOMM '89: Symposium proceedings on Communications architectures & protocols*, pages 1–12, 1989.

[9] Y. Dinitz, N. Garg, and M. Goemans. On the single-source unsplittable flow problem. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 290–299, November 1998.

[10] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. In *MobiCom '04: Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*, pages 114–128, 2004.

[11] Y. Ganjali and A. Keshavarzian. Load balancing in ad hoc networks: Single-path routing vs. multi-path routing. In *INFOCOM 2004: Proceedings of the twenty-third Annual joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 1120–1125, March 2004.

[12] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, San Francisco, 1979.

[13] S. Ghazal, L. Mokdad, and J. Ben-Othman. A real time adaptive scheduling scheme for multi-service flows in WiMAX networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, pages 1–5, November-December 2008.

[14] C. Gomes and H. Rivano. Fair joint routing and scheduling problem in wireless mesh networks. Technical Report 6198, INRIA, May 2007.

[15] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, August 2002.

[16] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. In *INFOCOM 2007: Proceedings of*

the 26th IEEE International Conference on Computer Communications, pages 1631–1639, May 2007.

[17] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[18] G. Hiertz, D. Denteneer, L. Stibor, Y. Zang, X. Costa, and B. Walke. The IEEE 802.11 universe. *IEEE Communications Magazine*, 48(1):62–70, January 2010.

[19] P.-H. Hsiao, A. Hwang, H. T. Kung, and D. Vlah. Load-balancing routing for wireless access networks. In *INFOCOM 2001: Proceedings of the twentieth Annual joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 986–995, April 2001.

[20] IEEE std 802.16-2004. IEEE standard for local and metropolitan area networks – part 16: Air interface for fixed broadband wireless access systems, June 2004.

[21] IEEE std 802.11-2007. IEEE standard for information technology – telecommunications and information exchange between systems – local and metropolitan area network – specific requirements – part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, June 2007.

[22] IEEE std 802.16-2009. IEEE standard for local and metropolitan area networks — part 16: Air interface for broadband wireless access systems, May 2009.

[23] IEEE std 802.16j-2009. IEEE standard for local and metropolitan area networks — part 16: Air interface for broadband wireless access systems — amendment 1: Multihop relay specification, June 2009.

[24] K. Jain, J. Padhye, V. N. Padmanabhan, and L. Qiu. Impact of interference on multi-hop wireless network performance. In *MobiCom '03:*

Proceedings of the 9th Annual International Conference on Mobile Computing and Networking, pages 66–80, 2003.

[25] A. Kabbani, T. Salonidis, and E. W. Knightly. Distributed low-complexity maximum-throughput scheduling for wireless backhaul networks. In *INFOCOM 2007: Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 2063–2071, May 2007.

[26] J. Kazemitabar, V. Tabatabaee, and H. Jafarkhani. Global optimal routing, scheduling and power control for multi-hop wireless networks with interference. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, pages 1–5, November-December 2008.

[27] M. Kodialam and T. Nandagopal. Characterizing the capacity region in multi-radio multi-channel wireless mesh networks. In *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 73–87, 2005.

[28] J. Kratochvíl and Z. Tuza. Algorithmic complexity of list colorings. *Discrete Applied Mathematics*, 50(3):297–302, May 1994.

[29] V. S. A. Kumar, M. V. Marathe, S. Parthasarathy, and A. Srinivasan. End-to-end packet-scheduling in wireless ad-hoc networks. In *SODA '04: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1021–1030, 2004.

[30] P. Kyasanur and N. H. Vaidya. Capacity of multi-channel wireless networks: impact of number of channels and interfaces. In *MobiCom '05: Proceedings of the 11th Annual International Conference on Mobile Computing and Networking*, pages 43–57, 2005.

[31] P. Kyasanur and N. H. Vaidya. Routing and interface assignment in multi-channel multi-interface wireless networks. In *IEEE Wireless Communications and Networking Conference, 2005. . WCNC 2005.*, volume 4, pages 2051–2056, 2005.

[32] S. Lee, G. Narlikar, M. Pal, G. Wilfong, and L. Zhang. Admission control for multihop wireless backhaul networks with QoS support. In *Wireless Communications and Networking Conference, 2006. WCNC 2006.*, pages 92–97, 2006.

[33] E. Leonardi, M. Mellia, M. Marsan, and F. Neri. Optimal scheduling and routing for maximum network throughput. *IEEE/ACM Transactions on Networking*, 15(6):1541–1554, December 2007.

[34] Q. Li and R. Negi. Prioritized maximal scheduling in wireless networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, pages 1–5, November-December 2008.

[35] H. Luo, S. Lu, and V. Bharghavan. A new model for packet scheduling in multihop wireless networks. In *MobiCom '00: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 76–86, 2000.

[36] A.-A. Mahmood and E. S. Elmallah. Incremental routing and scheduling in wireless grids. In *Global Telecommunications Conference, 2009. IEEE GLOBECOM 2009.*, pages 1–6, November-December 2009.

[37] A.-A. Mahmood and E. S. Elmallah. Joint non-bifurcated routing and scheduling in wireless grid mesh networks. In *BROADNETS 2009: Proceedings of Sixth International Coference on Broadband Communications, Networks, and Systems*, pages 1–7, September 2009.

[38] A.-A. Mahmood and E. S. Elmallah. An algorithm for incremental joint routing and scheduling in wireless mesh networks. In *IEEE Wireless Communications and Networking Conference, 2010. . WCNC 2010.*, pages 1–6, April 2010.

[39] A.-A. Mahmood, E. S. Elmallah, and A. Kamal. Non-bifurcated routing in wireless multi-hop mesh networks. In *LCN '07: Proceedings of the 32nd IEEE Conference on Local Computer Networks*, pages 279–286, October 2007.

[40] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proceedings SIGMETRICS Perform. Eval. Rev.*, volume 34, pages 27–38, 2006.

[41] P. S. Mogre, N. d'Heureuse, M. Hollick, and R. Steinmetz. A case for joint routing, scheduling, and network coding in TDMA-based wireless mesh networks: a cross-layer approach. In *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 1–3, October 2007.

[42] G. Narlikar, G. Wilfong, and L. Zhang. Designing multihop wireless backhaul networks with delay guarantees. In *INFOCOM 2006: Proceedings of the 25th IEEE International Conference on Computer Communications*, pages 1–12, April 2006.

[43] M. J. Neely and E. Modiano. Capacity and delay tradeoffs for ad-hoc mobile networks. In *BroadNets 2004: Proceedings of First International Conference on Broadband Networks*, pages 428–438, October 2004.

[44] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[45] N. Peppas and D. Turgut. A hybrid routing protocol in wireless mesh networks. In *Proceedings of Military Communications Conference (MILCOM)*, pages 1–7, October 2007.

[46] A. Raniwala and T.-c. Chiueh. Architecture and algorithms for an IEEE 802.11-based multi-channel wireless mesh network. In *INFOCOM 2005: Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 3, pages 2223–2234, March 2005.

[47] A. Raniwala, K. Gopalan, and T.-c. Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh net-

works. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):50–65, April 2004.

[48] Scalable Network Technologies. QualNet. `www.scalable-networks.com`.

[49] G. Sharma, N. B. Shroff, and R. R. Mazumdar. Joint congestion control and distributed scheduling for throughput guarantees in wireless networks. In *INFOCOM 2007: Proceedings of the 26th IEEE International Conference on Computer Communications*, pages 2072–2080, May 2007.

[50] S. Tai, R. Benkoczi, H. Hassanein, and S. Akl. A performance study of splittable and unsplittable traffic allocation in wireless sensor setworks. In *2006 IEEE International Conference on Communications*, volume 8, pages 3432–3437, June 2006.

[51] J. Tang, G. Xue, and W. Zhang. Interference-aware topology control and QoS routing in multi-channel wireless mesh networks. In *MobiHoc '05: Proceedings of the 6th ACM International symposium on Mobile ad hoc networking and computing*, pages 68–77, 2005.

[52] J. Tang, G. Xue, and W. Zhang. Maximum throughput and fair bandwidth allocation in multi-channel wireless mesh networks. In *INFOCOM 2006. Proceedings of the 25th IEEE International Conference on Computer Communications.*, pages 1–10, April 2006.

[53] Y. Tian, K. Xu, and N. Ansari. TCP in wireless environments: problems and solutions. *IEEE Communications Magazine*, 43(3):S27–S32, March 2005.

[54] L.-P. Tung, W.-K. Shih, T.-C. Cho, Y. S. Sun, and M. C. Chen. TCP throughput enhancement over wireless mesh networks. *IEEE Communications Magazine*, 45(11):64–70, November 2007.

[55] J. Wang, H. Li, W. Jia, L. Huang, and J. Li. Interface assignment and bandwidth allocation for multi-channel wireless mesh networks. *Computer Communications*, 31(17):3995–4004, 2008.

[56] W. Wang, Y. Wang, X.-Y. Li, and W.-Z. Song. Interference-aware joint routing and TDMA link scheduling for static wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 19(12):1709–1726, December 2008.

[57] D. Wu. QoS provisioning in wireless networks. *Wireless Communications and Mobile Computing*, 5:957–969, 2005.

[58] Y. Wu, Y. J. Zhang, and Z. Niu. Nonpreemptive constrained link scheduling in wireless mesh networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008*, pages 1–6, November-December 2008.

[59] Y. Xi and E. M. Yeh. Distributed algorithms for spectrum allocation, power control, routing, and congestion control in wireless networks. In *MobiHoc '07: Proceedings of the 8th ACM International symposium on Mobile ad hoc networking and computing*, pages 180–189, 2007.

[60] G. Xue, W. Zhang, J. Tang, and K. Thulasiraman. Polynomial time approximation algorithms for multi-constrained QoS routing. *IEEE/ACM Transactions on Networking*, 16(3):656–669, June 2008.

[61] L. Zhang. VirtualClock: a new traffic control algorithm for packet-switched networks. *ACM Transactions on Computer Systems*, 9(2):101–124, May 1991.

[62] D. Zhao, J. Zou, and T. D. Todd. Admission control with load balancing in IEEE 802.11-based ESS mesh networks. *Wireless Networks*, 13(3):351–359, June 2007.

[63] J. Zou and D. Zhao. Connection-based scheduling for supporting real-time traffic in wireless mesh networks. In *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008.*, pages 1–6, November-December 2008.

Appendices

Appendix A

IEEE 802.16 Standards

Standard	Objective	Status
IEEE 802.16-2001	Air interface for 10-66 GHz	Complete
IEEE 802.16a-2002	Amendment including 2-11 GHz	Complete
IEEE 802.16c-2003	Amendment including 10-66 GHz profiles	Complete
IEEE 802.16-2004	Revision, integrating above extensions	Complete
IEEE 802.16e-2005	Amendment on enhancements to support mobility	Complete
IEEE 802.16f-2005	Amendment on management information base for fixed systems	Complete
IEEE 802.16g-2007	Amendment on mobile management information base	Complete
IEEE 802.16k-2007	Amendment on bridging	Current
IEEE 802.16-2009	Revision, Air interface for fixed and mobile broadband wireless access	Current
IEEE 802.16j-2009	Multihop relay	Current
IEEE 802.16h	Improved coexistence mechanisms for license-exempt operation	Draft
IEEE 802.16m	Advanced air interface supporting 100 Mbps data rates for mobile clients and 1 Gbps data rates for fixed clients	Draft

Table A.1: The IEEE 802.16 versions

Appendix B

IEEE 802.11 Standards

Standard	Objective	Status
IEEE 802.11-1997	WLAN MAC and PHY	Complete
IEEE 802.11a	54 Mbps OFDM PHY at 5 GHz	Complete
IEEE 802.11b	11 Mbps DSSS PHY at 2.4 GHz	Complete
IEEE 802.11d	Compliance with regional regulations	Complete
IEEE 802.11g	54 Mbps OFDM PHY at 2.4 GHz	Complete
IEEE 802.11h	5 GHz operation in Europe	Complete
IEEE 802.11i	Security Enhancements	Complete
IEEE 802.11j	5 GHz operation in Japan	Complete
IEEE 802.11e	MAC enhancements for QoS support	Complete
IEEE 802.11-1999	Internationalization	Complete
IEEE 802.11c	MAC bridging	Current
IEEE 802.11-2007	Integrates a,b,d,e, and g-j	Current
IEEE 802.11k	Measurements of wireless channel	Current
IEEE 802.11r	Fast hand-off for roaming	Current
IEEE 802.11y	3.65 GHz (FCC band) operation	Current
IEEE 802.11w	Security for management frames	Current
IEEE 802.11n	600 Mbps MIMO at 2.4 and 5 GHz	Current
IEEE 802.11z	Extensions to direct link setup	Draft
IEEE 802.11p	Vehicular wireless access	Draft
IEEE 802.11v	Network management	Draft
IEEE 802.11s	Mesh networking	Draft
IEEE 802.11u	Convergence of 802.11 and GSM	Draft
IEEE 802.11aa	Audio video streaming	Draft
IEEE 802.11af	TV whitespace	Draft
IEEE 802.11ae	QoS Management	Draft
IEEE 802.11ac	Gbps throughput in $<$ 6 GHz bands	Draft
IEEE 802.11ad	Gbps throughput in 60 GHz band	Draft

Table B.1: The IEEE 802.11 Standards and Extensions