Opportunistic Vehicular Assisted Ferrying in Energy Efficient Wireless Mesh Networks

OPPORTUNISTIC VEHICULAR ASSISTED FERRYING IN ENERGY EFFICIENT WIRELESS MESH NETWORKS

BY

KEYVAN REZAEI MOGHADAM,

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2011)	McMaster University
(Electrical & Computer Engineering)	Hamilton, Ontario, Canada

TITLE:	Opportunistic Vehicular Assisted Ferrying in Energy Efficient Wireless Mesh Networks
AUTHOR:	Keyvan Rezaei Moghadam B.Sc., (Electrical Engineering) Iran University of Science and Technology, Tehran, Iran
SUPERVISOR:	Dr. Terence D. Todd
NUMBER OF PAGES:	xiv, 96

To my parents who have made a great effort in their life to teach me how to remain on human standards through difficulties

Abstract

Wireless mesh networks are widely used for various communication purposes and are often deployed in a broad range of outdoor applications. In cases where the deployment area is outside the reach of fixed infrastructure, energy efficient operation of the mesh nodes is very important. Many approaches have been explored to find energy efficient network solutions for these types of scenarios. This thesis proposes power saving mechanisms where the mesh network nodes opportunistically access coexistent vehicular networks. This is referred to as Opportunistic Vehicle Assisted Ferrying (OVAF). The history of using moving particles for message carrying can be found in research on intermittently connected networks. However, this approach has never been considered with fully connected networks.

Two different models are presented to model the OVAF mechanism: **a**. *A flow based model*, and **b**. *A packet based model*. For each model an analytic lower bound is obtained by formulating a linear integer optimization with different cost functions. Heuristics, which simplify the complexity of the problem, are then developed for each model. Arriving vehicles and generated packets are also modeled as random processes under various scenarios using different parameters. Results are presented which demonstrate the superiority of the OVAF routing method compared to conventional multihop forwarding (Up to 60% increase in energy saving).

Acknowledgements

I would like to thank my supervisor Dr. Terence D. Todd for giving me the opportunity to pursue my M.A.SC. studies at McMaster University and for letting me experiencing an autonomous research career, and also for his invaluable guidance throughout my research. He was exactly the type of the supervisor I needed giving enough time to his student to find their way and support them when it was necessary.

I would like to express my sincere gratitude to Dr. Zhao, Dr. Shirani, and Dr. Hranilovic for reviewing my thesis and giving me helpful comments. Dr. Zhao also supported me financially for one semester as my co-supervisor for which I'd like to thank her.

I am also very thankful to the members of Wireless Networking Lab especially Ali Shariat for all his helpful comments and his sincere friendship. I do also appreciate the great support of my friends Nooshin Tajik, Ali Khalatpour and Reza Najafi in the past year without which I might not be able to reach at this stage.

Notation and abbreviations

AP Access Point

BAC Building Automation and Control

DFF Drop-From-Front

DMS Data Mule Scheduling

DP Dynamically-Partitioned

DSRC Dedicated Short Range Communication

DTMN Delay-Tolerant Wireless Mesh Network

DTN Delay Tolerant Network

DTT Delay Tolerant Traffic

EA Energy Aware

EARP Energy-Aware Resource Provisioning

EEPT Energy Efficient Path Table

ETE Expected Transmission Energy

EZF Elliptical Zone Forwarding

FP Fixed Partition

LAN Local Access Network

LM-SPT Local Minimum Shortest-Path Tree

LT Life Time

MF Message Ferries

MIMO Multiple Input Multiple Output

ML Maximum Liftime

MMF Multihop Mesh Forwarding

MULE Mobile Ubiquitous LAN Extension

NMA Next Move Algorithm

OVAF Opportunistic Vehicular Assisted Ferrying

QoS Quality of Service

RFID Radio Frequency ID

SP Shortest Path

VANETs Vehicular Ad-hoc Networks

WAVE Wireless Access in Vehicular Environment

WLAN Wireless LAN

WMN Wireless Mesh Network

WSN Wireless Sensor Network

.

Contents

List of Figures

Chapter 1

Introduction

Wireless mesh networks (WMNs) are communication networks made up of radio nodes organized in a mesh topology. A mesh topology is a type of networking where each node must not only capture and disseminate its own data, but also serve as a relay for other nodes which are categorized as the node's set. These sets are structural units of the network and are called mesh. WMNs are a key technology for next generation networks. They have emerged as a flexible, low-cost and multipurpose networking platform which extends fixed wired Internet infrastructure. Because of their unique features, WMNs have undergone extensive technical development and many applications have emerged from this area. This has motivated researchers to reconsider the protocol design of existing wireless networks from the perspective of WMNs. Industrial standards groups, such as IEEE 802.11, IEEE 802.15, and IEEE 802.16, are all actively working on new specifications for WMNs [6]. There are also many techniques which have been explored in this context to increase the efficiency of these networks and make them more convenient to deploy.

1.1 Energy Efficiency Issues in Wireless Mesh Networks

Wireless local area network (WLAN) mesh networks are currently being used in a variety of network infrastructure applications in which radio coverage is required over expansive outdoor and rural areas. In these cases, continuous power sources are not always available and hence the use of limited battery powered networks are inevitable. Such conditions highlight energy consumption concerns of these networks and has made energy efficiency in WMNs an important issue. As a result, many previous papers have developed methods for reducing energy consumption [7]. These studies have considered both exhaustible (Such as battery powered networks) and renewable energy sources. Solar and wind powered mesh networks are examples of renewable energy sources which are used to make the WMNs more environmentally friendly [2]. Methods like the Local Minimum Shortest-Path Tree (LM-SPT) [8], a distributed localized energy efficient topology control algorithm for WMNs in rural areas, are also used to incorporate an energy aware node placement.

Other studies consider the charge and discharge pattern of the nodes' batteries. Batteries usually discharge more power than needed, and reimburse the over-discharged power later if they have a sufficiently long recovery time. Considering this fact, it is preferable to route data in a way that uses the nodes periodically and gives enough time for batteries to recover [1].

1.2 Delay Tolerant Mesh Networks

Delay-tolerant wireless mesh and sensor networks (DTMNs) are a subset of traditional delay-tolerant networks (DTNs). DTNs are assumed to carry traffic types which can tolerate a wider range of delays for packet delivery than is normally the case. DTNs often occur when the wireless mesh network nodes have insufficient transmission ranges due to mobility, which causes intermitent connectivity. These networks usually experience frequent, long-duration partitioning and may never have an end-to-end path [9]. As a result, the conventional routing protocols considered in ad hoc networks cannot be applied.

In most cases, DTMNs are categorized as mobile networks since they may need mobile elements to retain the connectivity between different islands of connectivity. Reasons that may force the wireless network to be modeled as DTNs are given in Reference [10], i.e.,

- Short transmission range of mesh nodes. This is a common assumption in several sensor networks due to limitations in power sources.
- Node mobility.
- Sparse deployment of nodes, which is unavoidable in certain applications.

ZebraNet [11] is a well-known example of a mobile intermittent network and is categorized as a DTN. This network is used to monitor wildlife with sensors attached to animals for collecting information. Due to security and energy consumption considerations these nodes cannot have large transmission ranges and since the number of nodes is limited, it is impossible to maintain a fully connected network.

1.3 Ferrying in Wireless Mesh Networks

A data ferry is often referred to as a data Mobile Ubiquitous LAN Extension (MULE), is the main feature of a typical delay tolerant intermittent network [4]. Ferries and data MULEs in intermittent WMNs and wireless sensor networks (WSNs) function to interconnect the isolated islands of the network [12]. This thesis discusses the possibility of using them to reduce battery usage of mesh nodes for pure multi-hop communication. Data MULEs and ferries can be chosen from the nodes of the primary network which are mobile [13] or they can be chosen from entities, not part of the network infrastructure, which travel through the deployed network area [14]. In the first scenario, the ferry can be controlled and so that the frequency, speed and path of the ferries traveling over the islands, are important issues that should be optimized. In the second scenario, it is usually a fixed-route ferry that cannot be controlled as it is not part of the network and so the problem is how to exploit the ferries more efficiently. DakNet is a very well-known example of ferrying which uses public transportation to implement data MULE's task of transferring data between inaccessible rural areas and the metropolitan networks [15]. All the networks which use ferries and Data MULEs are DTNs. In other words, only those networks can use the advantages of carrying data where some of their traffic is considered Delay Tolerant Traffic (DTT).

1.4 Vehicular Ad-hoc Networks

The concept of using wireless communication in moving vehicles has fascinated researchers since the 1980s. This has led to the development of Vehicular Ad-hoc Networks (VANETs), a technology where moving cars are used as nodes to create a mobile network. This type of system enables applications ranging from road safety, to those involving context-aware advertising and in-vehicle Internet media streaming. Recognizing the importance of VANETs, the FCC has licensed the operation of Dedicated Short Range Communication (DSRC) in the 5.9 GHz frequency band.

In practical situations, VANETs may co-exist with battery powered networks where energy efficiency is highly desirable [16]. This provides the opportunity to exploit these networks to build ferry-assisted networks. Highway traffic is a good example of a case where VANETs can be potentially used to play a ferrying role in delay tolerant WSNs which are deployed in the vicinity of the highway.

1.5 This Thesis Goal

This thesis introduces a new method for fully connected wireless mesh networks which are handling delay tolerant traffic. The new method, opportunistically uses mobile vehicles, which are not part of the network, to carry a portion of the network's traffic load subject to satisfying the Quality of Service (QoS) needed for the mesh network. Due to nature of the method, it is called Opportunistic Vehicular Assisted Ferrying (OVAF). The current work focuses on the network layer and finds the best achievable routing such that both the WMN and the VANET integrates properly. Two different approaches are considered for modeling of the proposed system:

- The flow based model
- The packet based model

In the first approach, the WMN deployment time is broken into smaller time periods, usually an hour, and in each time slot the average node flows are considered.

An optimization problem is formulated by incorporating the flows of the nodes in each time slot and heuristics are also presented to reduce the complexity for practical systems.

In the second approach the optimization problem is formulated to optimize the energy consumption of the network by controlling the path of each packet. The WMN deployment time is broken into the system time clock and the routing is performed in each time epoch for each packet. Different heuristics are presented to reduce the complexity of the algorithm. Results are presented which demostrate the performance advantages of OVAF compared to the conventional multi-hop approach.

The remainder of the thesis is organized as follows. In Chapter 2 recent research in wireless mesh networks is discussed. The energy issues are explored and different ways of improving battery lifetime are reviewed. Chapter 3 reviews ongoing work on the use of moving entities for improving intermittent delay tolerant network performance. The basic ideas of Data Mules and the Ferries is discussed in this section. We present the OVAF idea as well as a problem formulation in Chapter 4. The performance of OVAF is considered under various assumptions in Chapter 5.

Chapter 2

A Brief Review of Wireless Mesh Networks

In the past few decades, WMNs have emerged as flexible, low-cost and multipurpose networking platforms often with a wired backbone infrastructure connected to the Internet ([6], [17]). The nodes themselves can be either fixed or may include mobile nodes. MIT roofnet project [18] is an example in which a neighborhood can easily build a community mesh for Internet access purposes. This is accomplished by setting up mesh routers with flexible connectivity between houses to support distributed storage, data access, and video streaming. In this type of scenario, mesh networks can be used to reduce the cost of installing wired Ethernet cables.

In general, a wireless mesh network consists of three components: Access Points (APs), mesh routers and mesh clients. Mesh client devices are generally more diverse than mesh routers and can include laptops, tablet PCs, PDAs, IP phones, Radio Frequency ID (RFID) readers, sensors, Building Automation and Control (BAC) network controllers, and a wide variety of other devices. Unlike traditional ad hoc

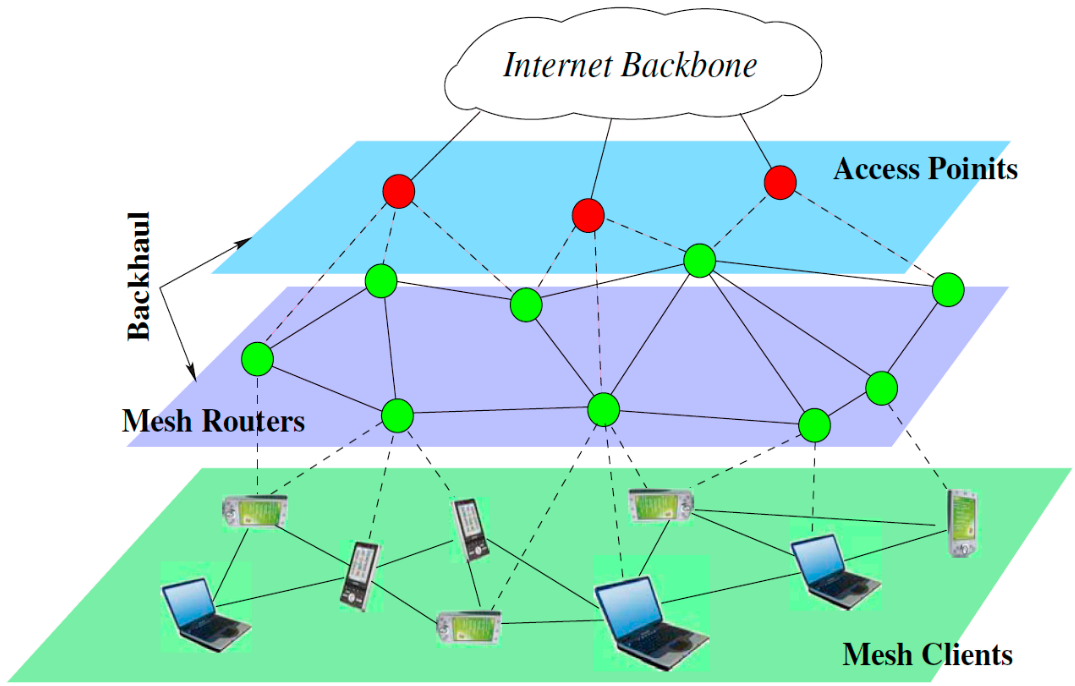


Figure 2.1: Architecture of a wireless mesh network [1].

networks, which are isolated and self-configuring, the architecture of mesh networks are layered in a hierarchical fashion and composed of wireless routers communicating between mesh clients and APs.

A typical wireless mesh network, shown in Fig. 2.1, usually has 30 to 100 mesh routers [1]. It includes APs which are fixed and wired to the Internet to provide high-bandwidth connections. The combination of wireless mesh routers and the APs form a wireless backhaul which provides client nodes with Internet entry points and also helps to ease client information exchange.

From the an architectural viewpoint, WMNs are categorized into three major sets [6]:

1. **Infrastructure/Backbone WMNs:** In this architecture, mesh routers form an infrastructure for clients as shown in Fig. 2.1. The backbone of the WMNs may include various types of radio technologies. Typically the mesh routers are self configured, self healing and connected to the Internet through the gateways.

2. **Client WMNs:** This architecture provides peer-to-peer networking among client devices. In this model, client nodes form the actual network that perform the routing. They are also responsible for configuration as well as providing connectivity for end-user applications. Consequently, the functions of the mesh routers are included in the client nodes. Generally these are single-radio devices similar to that which can be found in conventional ad hoc networks.

3. **Hybrid WMNs:** Hybrid wireless mesh networks are a combination of the previous two types. In this structure the network is accessed by mesh clients through either mesh routers or via direct meshing with other mesh clients. As before, the infrastructure provides connections to other networks such as the Internet, Wi-Fi, WiMAX, cellular, and sensor networks. The routing capabilities of clients provide improved connectivity and coverage inside the WMN.

2.1 Performance Issues in Designing WMNs

In the recent past, many technologies have been developed to improve both the performance and operation of wireless systems. In this section, some of the mechanisms which affect the design of WMNs are briefly summarized [19]:

1. **Radio Technology:** Enhancements in radio technology have had a significant effect on the performance and operation of WMNs. Examples of this include smart antennas, multiple input multiple output (MIMO) systems, multi-radio/multi-channel designs, software defined and cognitive radio. Although many of these technologies are still in their infancy, they are expected to play a central role in future wireless networks.

2. **Scalability:** Scalability in WMNs refers to the capability of maintaining the desired performance as the network scales to large sizes. For instance, without proper scalability, routing algorithms may fail to keep its tables updated which in turn may lead to routing failures. In addition, the energy supplies of the nodes may not be provisioned appropriately or the network may experience throughput reductions caused by unsatisfactory MAC protocol performance. To ensure scalability in WMNs, all the protocols from the MAC layer to application layer need to be scalable.

3. **Mesh Connectivity:** This is an ability which is achieved by incorporating a good mesh nodes deployment design in order that a fully connected network is realized. This may also help to increasing network lifetime by a more uniform node distribution of loading. Algorithms may be required for self-organization and topology control of the mesh nodes to ensure reliable mesh network connectivity.

4. **Security:** Security in WMNs focuses on the protection of data exchange and includes sender authentication. Although many security mechanisms have been developed for wireless LANs, there is still no fully complete schemes that apply

to WMNs. For example, due to the distributed nature of WMNs, there is no centralized trust authority for key distribution.

5. **Compatibility and interoperability:** This is a default requirement for most of WMNs to support network access for mesh client networks. Therefore, WMNs should be developed in a way which is compatible with previous technologies wherever possible.

2.2 Energy Efficiency Issues in WMNs

Outdoor applications require continuously operating networks when battery-powered mesh routers are used. As a result, it has become an important issue to optimize performance from an energy-aware point of view [1]. Various strategies have been considered to increase the energy efficiency and hence the lifetime of WMNs. Some of these methods are introduced in the remainder of this section.

2.2.1 Optimum Energy Aware Routing

One approach to improve energy efficiency is to use an appropriate energy aware routing strategy. Sometimes this is done by formulating an optimization problem which incorporates flow conservation and transmission power consumption of nodes. The objective of the optimization problem usually minimizes the energy consumption of the network or seeks for the maximum achievable lifetime. The obtained optimum result is a bound for this objective.

There are also various heuristics which have been developed to guarantee a satisfactory energy efficiency with lower complexity compared to an optimum solution.

As an example, Dijkstra's Algorithm can be used by setting the weights of each link in proportion to the battery usage of the source node [20]. Other weight labeling strategies may be incorporated as well, based on the performance requirements of the deployed WMN [21]. Another example is [22] in which a capacity control algorithm is introduced. The algorithm is able to improve node outage performance by introducing access point capacity deficit when it is needed. In this method the shortest path is found based on Dijkstra's Algorithm. To avoid outage the capacity of each node is controlled as a function of remaining battery reserves which forces the network to route traffic through nodes with higher battery supplies.

Another strategy for achieving longevity in mesh networks is studied in [23]. The goal of this method is to evenly distribute the load over the mesh nodes in the network. This is achieved by incorporating some modifications to the routing metric in the Hybrid Wireless Mesh Protocol (HWMP). Through experiments, it is verified that this algorithm extends the lifetime of the network [23].

2.2.2 Solar Powered Wireless Mesh Nodes

Outdoor WMNs without access to the power grid can be operated from sustainable energy sources such as solar and wind power. Nodes powered by energy sources of this kind are untethered and can be installed and relocated in a cost efficient network deployment [24].

In solar powered WMNs, prior to network installation, the nodes must be provisioned with a solar panel and battery combination that is sufficient to accommodate the anticipated network. The assigned configuration for each node should meet the desired deployment duration of the network without any outage. This is normally

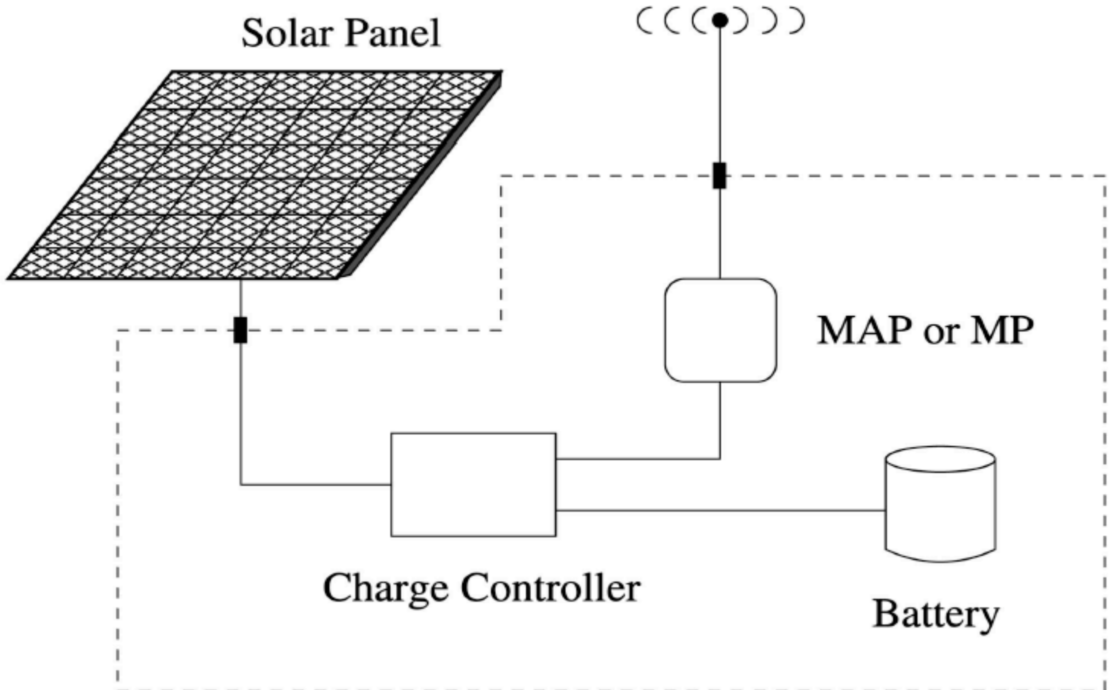


Figure 2.2: Architecture of a solar powered AP [2].

done by considering a temporal load profile for each node and then by assigning resources that are based on historical solar insolation. The insolation data can be obtained using historical meteorological data for the geographic location where the network is to be deployed [2].

Resource provisioning is often done to configure nodes with the least cost in terms of battery and solar panel size, while meeting the desired lifetime of the network. The method that is used for this purpose is called Energy-Aware Resource Provisioning (EARP) and is introduced in [2]. The basic architecture of a solar powered AP is shown in Fig.2.2

2.2.3 Nonlinear Battery Aware Scheme

Recent studies of battery technology have shown that discharging a battery is a nonlinear process. In fact batteries tend to discharge more than needed, and reimburse the over-discharged power later if they have a sufficiently long recovery time [25]. To consider this nonlinear behavior, routing algorithms should be developed to give the nodes enough resting time for recovery. For this purpose, first the relation between the mesh nodes transceiver parameters and battery behavior is inspected. This results in a multiple current battery model that can accurately describe battery behavior with multiple current inputs. As an example, [25] presents two different battery aware mesh scheduling algorithms:

1. *The coverage algorithm:* The main idea of the coverage algorithm is to give neighboring mesh routers the flexibility to collaboratively adjust their transceiver radii and dynamically recover their over-discharged battery power.

2. *The backhaul routing algorithm:* This algorithm adopts multiple models of battery current to calculate battery loss at routers and then uses them to schedule mesh backhaul routing.

2.2.4 Energy Aware Topology Control

Topology control refers to determining mesh nodes' geographical positions prior to their installation. Designing the topology of a network helps the network nodes act energy efficiently and reduce their transmission power while preserving full connectivity. The Local Minimum Shortest-Path Tree (LM-SPT) is an example of an energy efficient topology control scheme which is introduced in [8]. This is a distributed algorithm in which each node is using only the information gathered locally to determine its own transmission power. There are two phases in the implementation: First is the construction of a minimum local shortest-path tree and then the removal of all unidirectional links. The achieved network topology preserves network connectivity as well as possessing some other important features such as:

1. Reductions in the average node degree.

2. Evenly distributed power consumption among the nodes.

3. A reduced total power consumption leading to longer connectivity periods.

Chapter 3

Mobility in Delay Tolerant Networks (DTNs)

Delay tolerant networks are designed to maintain the desired QoS without the need for instantaneous data delivery. The tolerable data delivery delay in a given network varies over a wide range which depends on the network architecture and its useage specifications. Accordingly, DTNs are categorized into two major sets in terms of their tolerable delay range:

1. *The time clock range of delay:* The tolerable delay in these networks is on the order of milliseconds. The network is fully connected and the delivery delay is mainly caused by traffic conjestion. Hence, the objective is mainly focused on finding the best scheduling scheme in order to minimize the packet delivery delay. DT wired backbone networks are a good example of this type of network.

2. *The minute range of delay:* The tolerable delay in this category spans to wider ranges and this allows incorporating other means of data delivery rather than

pure multihop wireless communication. Intermittent wireless networks are the most important class in this category. The main issue in this class of DTNs is in the use of carriers in an efficient way to increase the network connectivity and reduce its load burden. As a result, mobility is an important feature that one may see in the implementation of this class of networks.

This chapter focuses on the later type of DTNs and explores the recent strategies which are used to incorporate mobile entities in data delivery.

3.1 Mobile Entities in DTNs

Until recently, data forwarding in delay tolerant WSNs and WMNs was mainly accomplished through multihop forwarding. However, over the past decade, researchers have also included mobile elements into the DTN infrastructure . Depending on the application of the DTN, the role of the mobile elements can be fulfilled by mobile sensor nodes, vehicles, or animals. The mobile elements can play the role of a base station or function as memory for carrying data from one point to another. The mobile entities in these networks can be categorized into two major sets:

- *Network controlled mobile elements:* In this scenario, the mobile elements are a part of the network and their trajectory and speed can be set by the network based on the network's requirements during each time period.

- *Self controlled mobile elements* In most cases, the mobile elements in these networks are not primarily a part of the network. In fact, they belong to the outer world infrastructure and the network can exploit them for its own benefit.

This section will review different DT wireless network configurations. We will look at various ways that mobility can be embraced as a feature to improve the DTN parameters such as connectivity, cost, lifetime, etc. Overall, recent ideas regarding the use of mobile entities in WSNs and WMNs can be categorized as follows:

- *Data MULEs:* Data MULEs are often used to carry data from nodes to base stations. They can be controlled by the network or they may have random movement. They have long term buffering for data delivery, high latency and the network they are moving in has low energy consumption. These types of networks are fully discussed in: [26], [27], [4], [28], [29], [3], [30] and [31].

- *Controlled Mobile Base Stations:* In this case, the base station has moving ability and is equipped with satellite communication devices. Its mobility is controlled by the network and it has long term buffering for delivering data from source to destination. The packets in these networks can tolerate large delays which in turn can be exploited in favor of low network energy consumption. There are various discussions of their performance which can be found in [32], [33], [34], [35] and [36].

- *Data Salmon:* Data salmon is a type of network controlled mobile entity which is specifically designed for large scale sensor networks. It has short term buffering for node to node data delivery, low delay tolerance, and medium energy consumption. It starts the data gathering tour periodically from the static data processing center, traverses the entire sensor network, gathers data from sensors while moving, returns to the starting point, and finally, uploads this data to the data processing center. This type of mobility in DTNs is extensively discussed in [32].

- *Message Ferries:* Message ferries may be either network controlled or self-controlled. They have long term buffering for delivering data from one island of nodes to other islands which lack connectivity. Their traffic may tolerate large delays which helps achieving low energy consumption. This type of DTN are extensively discussed in [37], [5], [38] and [39].

The rest of this chapter explores the recent work done under the two major topics of data MULEs and message ferries since the structure used in these strategies are closer to that used in this thesis.

3.2 Data MULEs

Data MULEs are moving entities collecting data from sensors when they are in the nodes' communication range. They are usually designed to visit all the nodes and the base station regularly. They buffer data, and drop them off to wired APs which are usually deployed somewhere outside of the sensor node region. This can lead to substantial power savings at the sensors as they only have to transmit over a short range and there is no power usage for multihop routing of packets. Data MULEs have been used in many recent sensor network applications, e.g., a robot in underwater environmental monitoring, an unmanned aerial vehicle (UAV) and in structural health monitoring. The MULEs three tier architecture is depicted in Fig. 3.1 [3].

Depending on the application, a number of tiers in the three-tier abstraction can be collapsed into one device. For example, to reduce latencies in the traffic monitoring application, the MULEs can be equipped with an always-on connection (such as a cellular or satellite phone) which would allow them to act as the top and the middle

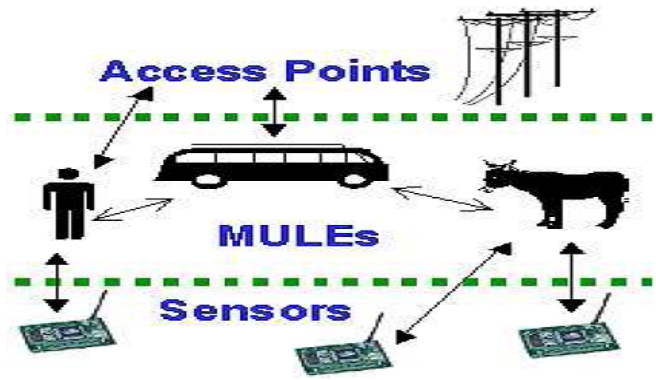


Figure 3.1: MULEs three tier architecture [3].

tier.

The data MULEs are chronologically first presented as self-controlled entities where their arbitrary movements can be exploited to increase network lifetime. Later the concept is extended to network controlled entities. In the following discussion, both types of MULEs are considered.

3.2.1 Self Controlled Data Mules

The role of the data MULEs in this scenario is typically played by either vehicles (cars, buses) outfitted with transceivers or by animals equipped with sensors. The latter case is specific to habitat monitoring applications. To model the random movements of self controlled MULEs, a grid should be fitted in the deployment area as shown in Fig. 3.2 [3].

The main assumptions of this model are [3]:

- The underlying topology on which sensors, MULEs and APs are placed is assumed to be on a discrete and finite 2D grid. Further, for analytical simplicity

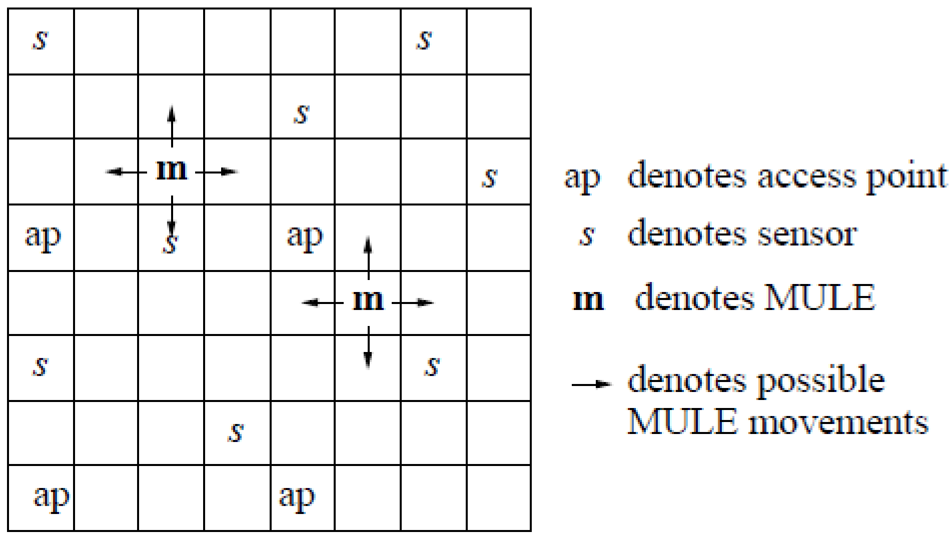
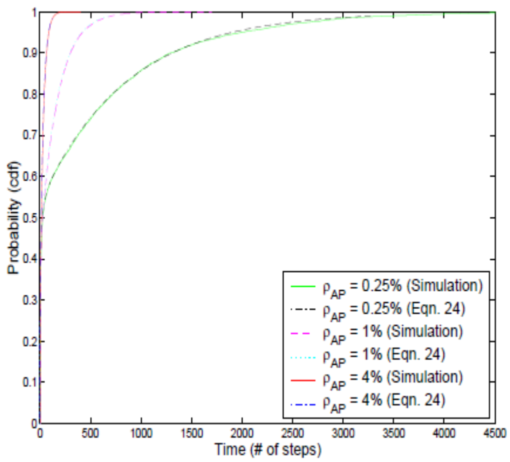


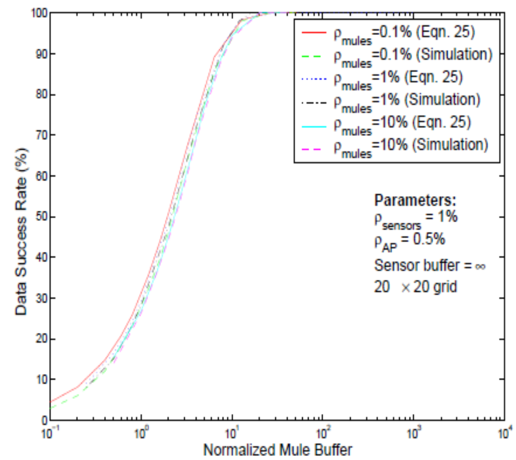
Figure 3.2: Two dimensional grid for modeling the network [3].

the planar topology is assumed to be the surface of a torus (i.e., the grid is wrapped in both the north-south and the east-west directions).

- Only a fraction of the grid points are occupied by sensors and APs. The access points are modeled to be uniformly spaced on the grid while the sensors are randomly distributed.
- The network evolves synchronously with a global clock. At every clock tick the following events take place:
  - Sensors generate one unit of data
  - Every MULE moves one step in the grid
- The MULEs communicate with the sensors or APs only when they are co-located at the grid points.



(a) Cdf of the return times (R_AP) for the access point set (20 × 20 grid).



(b) Data success rate vs. normalized MULE buffer size for ρ_{mules} =0.001; 0.01 and 0.1 (20 × 20 grid).

Figure 3.3: Network performance under different assumptions [3].

- The MULEs are assumed to perform a simple symmetric random walk on the grid.

The average delay or average data loss is the main important performance index of these networks. It is affected by different parameters such as the number of MULEs and the number of APs. Some of the general trends showing the effect of different parameters on the system performance are depicted in Figure 3.3 [3].

Fig. 3.3(a) depicts the probability of revisiting an access point by data MULEs for different densities of APs. The number of packets which are not dropped due to a lack of space in the buffers is called the data success rate and Figure 3.3(b) shows this variant versus the normalized MULE buffers for different MULE densities. This plot, along with many other experiments in the literature have confirmed following conclusions.

1. The sensor buffer requirements are inversely proportional to the density of the MULEs (ρ_{mules}).

2. The MULE buffer requirement is inversely proportional to both ρ_{mules} and density of APs (ρ_{AP}).

3. When the sensor buffer is large the buffer capacity on each MULE can be traded-off with the number of MULEs to maintain the same data success rate.

4. The change in the buffer capacity on each sensor needs to be greater than the change in the number of MULEs to keep the same data success rate.

3.2.2 Network Controlled Data MULEs

Network controlled MULEs are attractive alternatives to the multihop forwarding method in WSNs. In this structure, there are one or more mobile nodes which have unlimited energy supply, moving ability and they receive their controlling data directly from the base stations. This is contrary to the self controlled data MULEs which are originally parts of the outer space environment.

The most important disadvantage of data MULEs is the generally long time that they need for collecting data, which in turn incurs larger data delivery latency. Accordingly, optimizing data delivery latency is vital in the data MULE approach to make them useful in practice. In the DTN structures which are taking advantage of self controlled data MULEs, the only controllable parameters are ρ_{mules} and ρ_{AP}. However, having an increase in any of these variables affect the network cost significantly. On the other hand, in the network controlled MULE scenario there are more variables to control. The controllable parameters include a wide range, from the path selection of the data MULE to the speed control for each of them. This can help decrease the data delivery delay and data loss of the network with a reasonable cost.

To control a data MULE for data collection, the path and speed of the data MULE should be determined and also the scheduling part should be managed carefully. Scheduling in this context refers to the order of the sequence of time slots at which the data MULE leaves for a node to collect data. Nevertheless, optimizing all of these parameters is an NP-hard problem, which is a result of the NP-hardness of the simplified path selection problem. As a result, in all of the studies in this context, either the problem is simplified or some heuristics are used. Some of the simplifications are performed by employing assumptions that restrict the capabilities of sensor nodes and data MULEs, such as considering constant speed for the data MULEs. Although these assumptions have simplification effects, they often make the formulation only applicable to a few specific applications and settings.

The most recent work under this topic is [4] which designs a Data Mule Scheduling (DMS) problem to provide a comprehensive and flexible problem framework in which it is possible to fully exploit the networking and mobility capabilities. For this purpose, first the problem should be decomposed into following three subproblems (See Figure 3.4 [4]).

1. *Path selection:* Determines the trajectory of the data MULE such that it will pass within the communication range of each sensor node for a minimum number of one in each round trip.

2. *Speed control:* Determines how the data MULE changes its speed along the path, so that it spends enough time within each node's communication range to collect all the data.

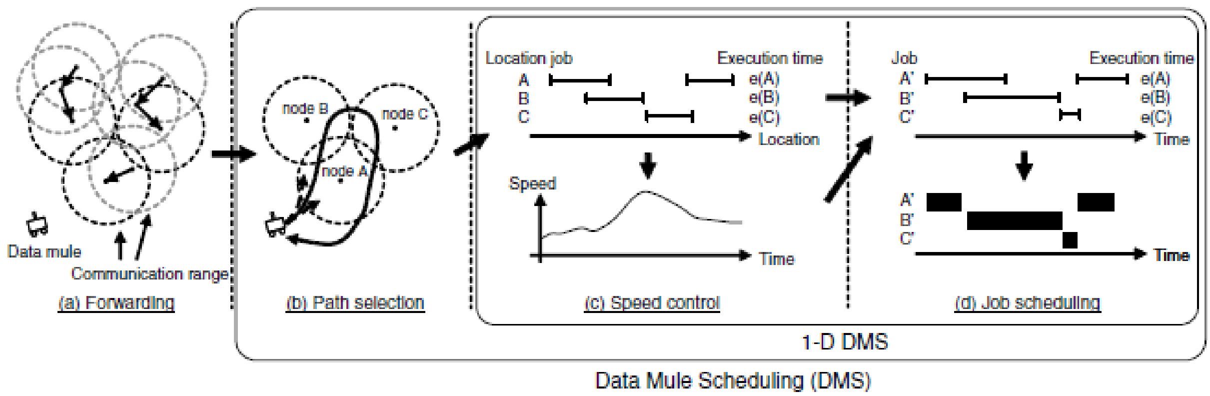


Figure 3.4: Different subsections of the DMS problem [4].

3. *Job scheduling:* Determines the schedule of data collection from each node.

The last two subproblems are solved jointly as a scheduling problem with both location and time constraints in [30] where it is called the 1-D DMS problem.

The path selection subproblem is formulated as an independent problem. The objective is to find a path such that the shortest travel time of the data MULE in the 1-D DMS problem induced by that path is minimized. However, finding a smooth path is computationally expensive and maneuvering the data MULE along such a smooth path is often difficult as well. For these reasons, it is advantageous to design and analyze a simplified path selection problem. It is a wise strategy to consider a complete graph having vertices at sensor nodes' locations and assume the data MULE moves between the vertices along a straight line. Each edge is associated with a cost and a set of labels, where the latter represents the set of nodes whose communication ranges intersect with this edge. This way, the data MULE can collect data from nodes in the labels associated with it, while traveling along an edge. Considering this approach, the objective will change to finding a label-covering tour that minimizes the total cost of the edges in the tour. Euclidean distance may be used as the cost metric, since it has been observed in the experiments that it has a strong positive correlation with the shortest travel time in the induced 1-D DMS problem. Since the simplified problem which is stated above is still an NP-hard optimization, heuristics still need to be considered [31].

When the path is determined, the speed scheduling problem can be easily solved based on delay criteria, link capacities and the amount of data which should be carried from each sensor node to the base stations.

There is also a recent approach which introduces a method for data MULEs to minimize the data delivery latency at the expense of more network energy consumption [4]. This is the closest thing to the OVAF approach as it combines the multihop data forwarding and the data carrying at the same time. Contrary to the pure data MULE approach, in which each node sends its data only to the data MULE, they can forward their data to other neighboring nodes as well. More importantly, if a node decides to forward all data to other nodes, the data MULE does not need to collect data directly from this node. Thus the data MULE can possibly take a shorter path to reduce the travel time. The problem of forwarding data in each period is formulated as follows.

- λ_i: Data generation rate of Node i.
- E_{limite}: Energy consumption limit at each node per unit time.
- E_r, E_s: Energy consumption for receiving and sending unit data.
- R: Bandwidth, i.e., maximum data rate that each node can communicate with other nodes and the data mule.

- x_{ij}: Amount of data sent from Node i to j per unit time.

Objective: Minimize $\sum_i d_i \lambda'_i$ where d_i is the distance between Node i and the base station and λ'_i is the data rate that Node i sends directly to the data mule. We have $\lambda'_i = \sum_i x_{ij} + \lambda_i - \sum_j x_i j$, which is the difference between the incoming and outgoing data rates.

Constraints:

- $x_{ij} = 0$

- For $i \neq j, x_{ij} \geq 0$ if node j is in the communication range of node i. If not then $x_{ij} = 0$

- Flow consevation: $\lambda'_i \geq 0$

- Energy constraint for each Node i:

$$E_r \sum_j x_{ij} + E_s(\sum_j x_{ij} + \lambda_i) \leq E_{limit}, \tag{3.1}$$

- Bandwidth constraint for each link capacity:

$$2 \sum_j x_{ij} + \lambda_i \leq R \tag{3.2}$$

With this strategy, the delay is decreased to a great extent while the energy constraints of the sensor nodes is preserved. The obtained results form this approach show the delay for different levels of energy in Figure 3.5 [4]. As can be seen by increasing the energy constraint limit, the average delay decreases.

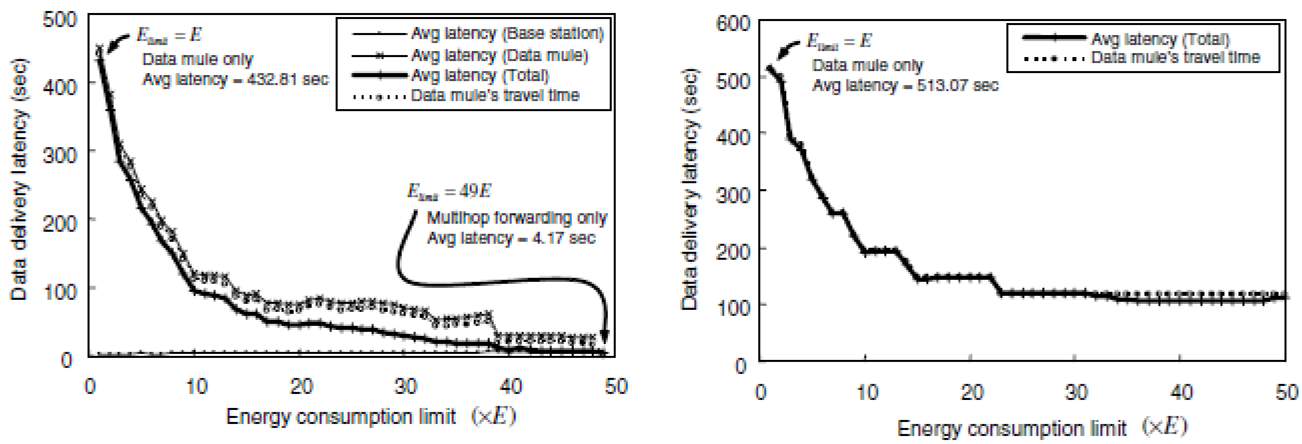


Figure 3.5: Data delivery latency for different energy limit constraints for 2 scenarios: connected and disconnected networks [4].

3.3 Message Ferries

Message Ferries (MF) are mobile entities used more often in intermittent DT networks. Unlike the networks with data MULEs, in the intermittent network structure the nodes are connected in small islands. These islands are not connected with each other and message ferries play the role of mobile entities to connect them (Fig. 3.6 [5]). They may be either controlled by the network or self controlled with predicted mobility, like bus routes. It has been shown that message ferry routing can achieve reasonable performance even when the nodes are highly disconnected.

Message Ferrying is studied extensively in [37] where the basic design of the MF scheme is proposed and a general framework to classify variations of the MF systems is developed. There are five major dimensions to the design of a message ferrying system, namely,

- The mobility of ferries

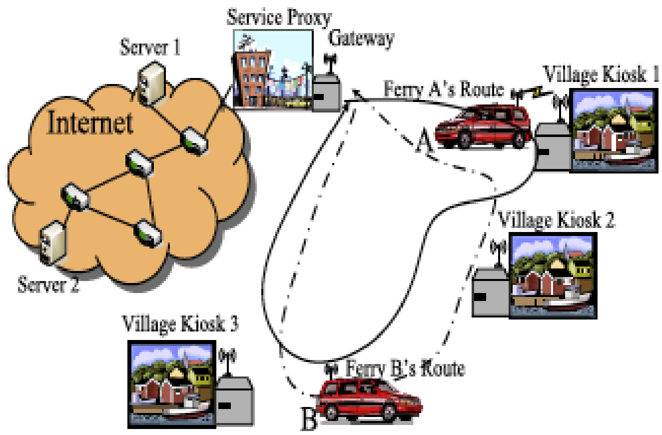


Figure 3.6: The general topology of ferries [5].

- The mobility of regular nodes
- The number of ferries
- The level of coordination
- Ferry designation

Clearly, the system behavior, the delivery performance, and hence the optimal strategy, are dependent on the assumptions made.

In a more developed MF scheme that is presented in [5], the method allows urgent messages with more stringent delay requirements to be delivered earlier than regular messages. As a result, the system analyzes these two types of messages according to their priority and arrival distribution. It also considers the limitations on buffer sizes at the both ferries and nodes. Then it designs an elliptical zone forwarding (EZF) scheme for the ferry route. They study how three different buffer allocation schemes affect the fraction of delivered urgent messages that meet the deadlines. They also

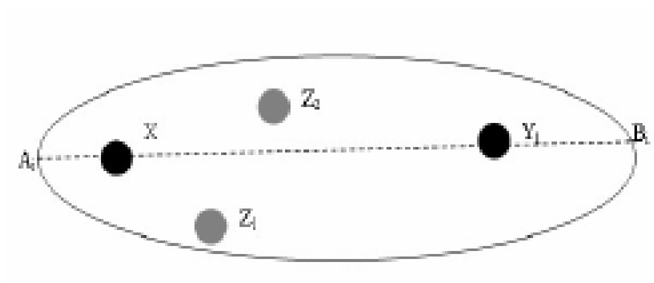


Figure 3.7: Divided region of network area used for path selection [5].

explore the dropping rates of urgent and regular messages due to buffer contention when used together with the EZF scheme.

The three simple buffer allocation schemes that are normally implemented in routers, namely, the Drop-From-Front (DFF) scheme, Fixed Partition (FP) scheme and Dynamically-Partitioned (DP) scheme are usually adopted for MF buffering systems. In the DFF scheme, older messages will be dropped to accommodate new messages when the buffers are full. For the FP scheme, $x\%$ of the buffer space is reserved for urgent messages. In the DP scheme, the regular messages are allowed to occupy at most $(1-x)\%$ of the buffer space but they can be preempted by urgent messages. In this protocol the ferry maintains a list of nodes that need to be visited either because messages need to be picked up or dropped off. This list is ordered based on the deadlines of the messages (assuming that the reservation message indicates the message deadlines). The ferry maintains a list of destinations that need to be visited and another list of reservations that it has received. After the ferry has visited a node, it checks its destination list and performs the following computation to decide if there is any node in the reservation list that it can visit before visiting a node in the destination list with the earliest deadline.

Assume that there are 3 nodes in the destination list of the ferry, namely Y_1, Y_2, Y_3

and that currently the ferry is at Node X. For the destination with the earliest deadline, Y_i , the ferry uses X and Y_i as the force, calculates the delay requirement R_i for Y_i and uses R_i as the length of the major axis to construct an ellipse. For example, as shown in Figure 3.7 [5] the length of A_iB_i is equal to R_i. Then, the algorithm looks for a possible node, Z, within the ellipse to visit. This node Z is chosen as the next node to be visited if we can guarantee that the traveling time for $XZ + ZY_i < R_i$. If such a node exists and the ferry has enough buffer space to accommodate its request, the ferry will visit that node. If more than one node exists, then the node with the minimum $XZ + ZY_i$ will be chosen to be visited next. Otherwise, the ferry will just visit the next node in the destination list with the earliest deadline. Such an approach is taken in an attempt to reduce the overall message delay while maintaining a high delivery ratio for urgent messages.

Chapter 4

Opportunistic Vehicular Ferrying

Vehicular ad-hoc networks support a variety of inter-vehicle and vehicle-to-roadside applications. In many cases these networks will also co-exist with battery operated networks such as those deployed for sensor and mesh network applications. In these latter types of networks, mesh node energy efficiency is often of paramount importance. This motivates the idea of incorporating vehicular data ferrying into the operation of wireless mesh networks.

This chapter proposes improvements in mesh network energy efficiency by opportunistically ferrying delay tolerant traffic through co-existent vehicular networks. This is accomplished by a mechanism referred to as Opportunistic Vehicle Assisted Ferrying (OVAF). This approach dynamically routes traffic through the vehicular network when this forwarding is consistent with the Quality of Service (QoS) requirements of the forwarded traffic flows. An illustrated example of the OVAF scheme is shown in Fig. 4.1 in which a WMN with seven nodes is depicted with one base station. According to the figure, to route a flow from the depicted sensor node to the base station, two options exist. The one which is shown in a dotted line is the pure

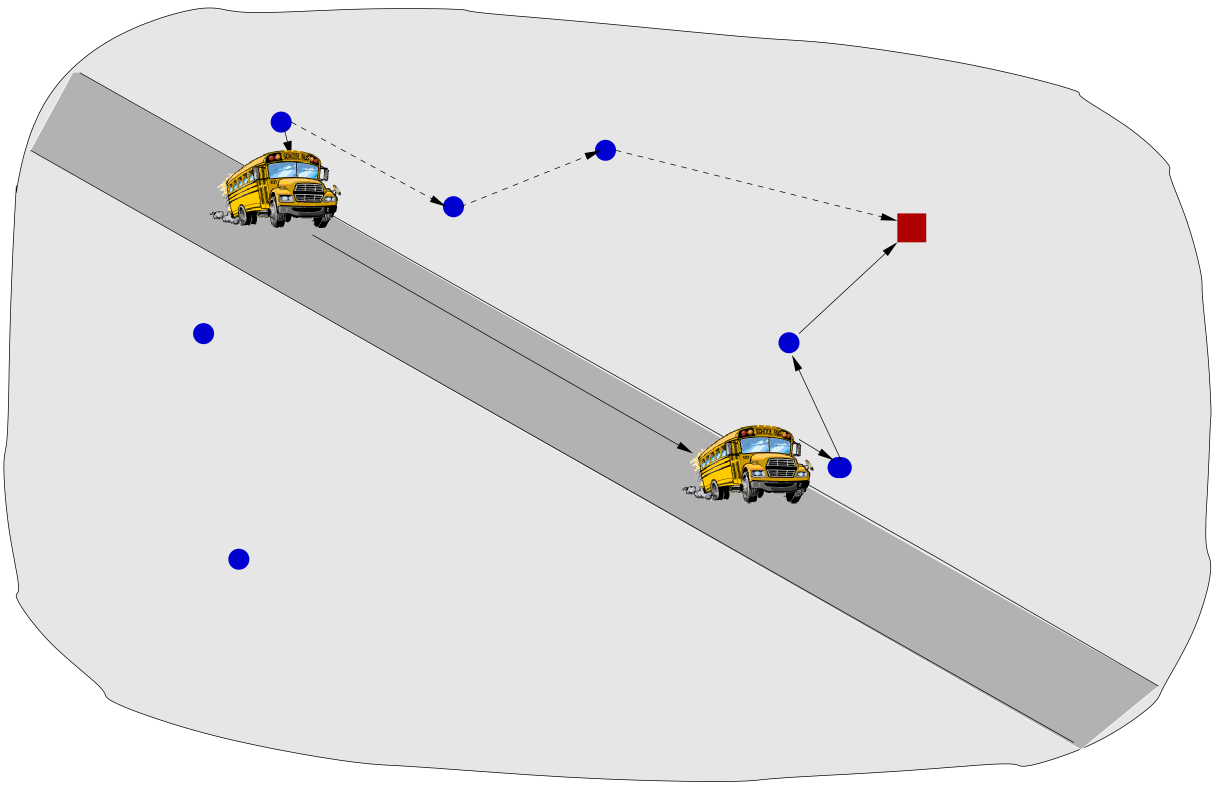


Figure 4.1: Wireless mesh network with a co-existent vehicular network path. The vehicular path is shown as the dark gray path. The mesh and base station nodes are shown as circles and squares, respectively.

multihop forwarding path and the other, shown with a solid line is a combination of a vehicular path and the conventional multihop forwarding scheme.

The OVAF approach is able to provide more options for a flow to route from its source to the destination. In many cases, it can also substitute a shorter *path* by using the vehicular ferrying approach. In this thesis, the word *path* refers to those links which directly connect wireless mesh nodes to each other. The length of a path for a given flow route is defined as the aggregated length of such links which are contributing to form the route. As an example, according to this definition, the solid line path in Fig.4.1 is shorter than the dotted path.

Fig. 4.2 is another illustration of the WMN with a co-existing vehicular network path. $R1$ is the communication range of the vehicles passing through the vehicular path. Those nodes which are in communication range of vehicles are *gateways* to the vehicular networks. $R2$ is the communication range of the wireless mesh nodes.

Defining this approach for data delivery, the main issue will be finding an appropriate path to route the packets. Two different models are introduced for this problem. The first one is a flow based model in which data packets are merged into the flows and then the best path for each flow is investigated. The other model is packet based in which data packets are considered individually and the best path for each is explored afterward.

For both of these models a lower bound on the performance is achieved by formulating the routing problem in the form of an integer linear program. Two objective functions which minimize the energy usage of the networks in different aspects have been explored for the integer linear optimization problem. Different heuristics are presented for both models which are lower in complexity. Simulation results show large improvements in network performance when OVAF is used compared to conventional single-network multi-hop forwarding.

4.1 Problem Description

Our problem deals with delay tolerant wireless mesh networks which have access to vehicles moving through their deployment region. Figure 4.2 shows an example of this type of situation where a vehicular path is given by the dark curve. We assume that the mesh nodes are all battery powered and have a maximum transmit power which imposes a limit on each node's transmission range. As illustrated in the Fig.4.2, two

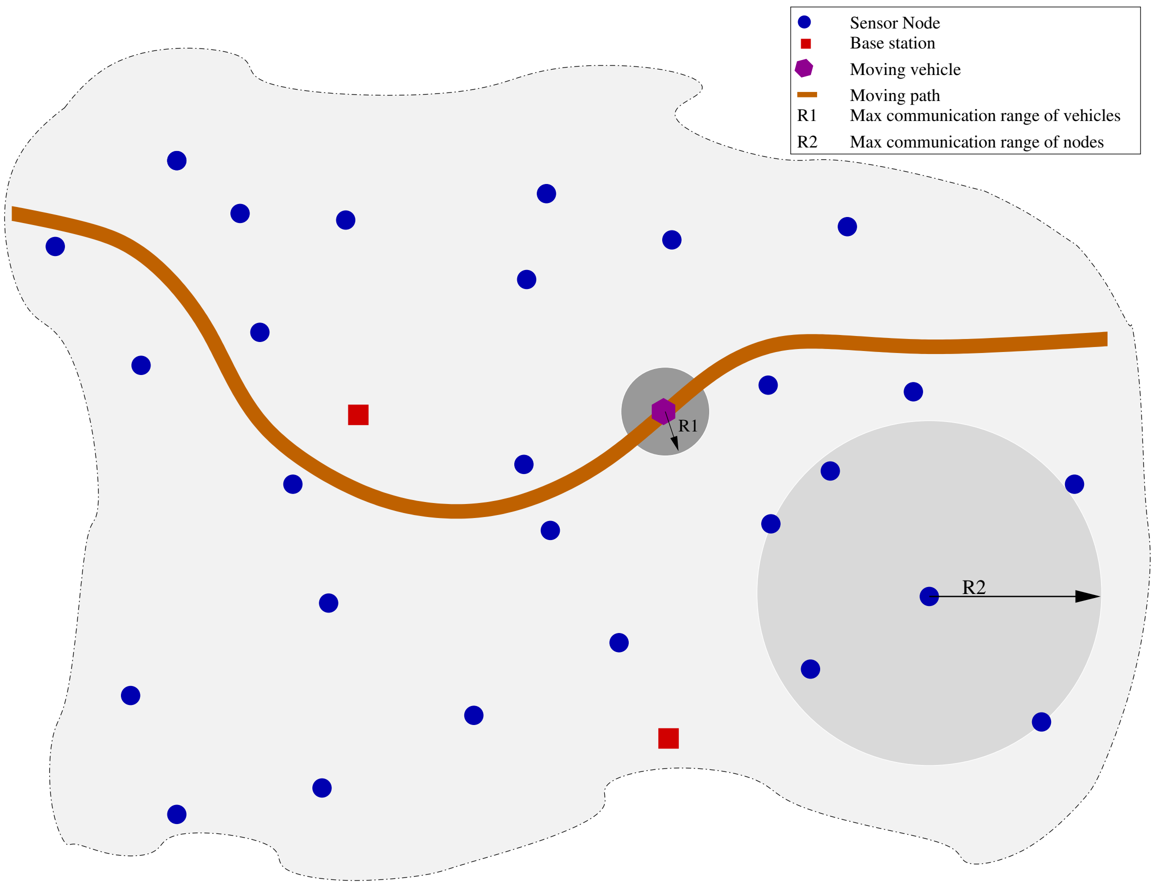


Figure 4.2: Wireless mesh network with a co-existent vehicular network path. The vehicular path is shown as a curve. The mesh and base station nodes are shown as circles and squares, respectively.

nodes are assumed to be connected directly when $P_{T,Max} > P_{T,ij}$, where $P_{T,Max}$ is the maximum transmission power and $P_{T,ij}$ is the required power for two nodes, i and j, to communicate directly. It is assumed that $P_{T,ij}$ is given by:

$$P_{T,ij} = b_1 + b_2 d_{ij}^{\alpha},$$

where b_1 is a constant which depends on the minimum power required for signal reception and d_{ij} is the distance between Nodes i and j. b_2 and α model the path loss for the transmitted signal [40] and $2 \leq \alpha \leq 4$ is the exponential path loss coefficient. In our results we assume that $\alpha = 2$.

It is assumed that the mesh nodes and base stations are aware of the network topology and can adjust their transmit power to use the minimum needed for communicating on each link. We also assume that the mesh nodes can communicate with passing vehicles while the vehicles are within radio coverage range. There are also one or more base stations which collect information and communicate to nodes in the mesh network. These are shown by squares in Figure 4.2.

The flowing subsections introduce and formulate the two different models used for the problem.

4.2 Flow Based Model and Formulation

In this section we describe the flow based model and formulate an optimization that can be used to compute a lower bound on the energy efficiency of the mesh network. This bound is compared to a proposed heuristic.

In the flow based model, network flows are assumed to be unsplittable to reduce undesired behavior such as jitter. We assume that there are delay constraints associated with the packets generated by each flow. The problem is to find the optimum routing as a combination of multi-hop forwarding and vehicular packet forwarding, such that the delivery time of the packets does not exceed their delay constraint, and such that mesh network energy efficiency is increased. We will assume that flows are differentiated based on coarse grained delay tolerance, such that the delay constraints of all the packets in a given flow are satisfied by the chosen route. A route for a particular flow may consist of several different multi-hop and vehicular forwarded segments.

The mesh nodes and base stations are identified by integers in the sets $\mathcal{N}$: $\{1, 2, .., N\}$ and $\mathcal{B}$: $\{1, 2, .., B\}$, respectively. Each road position where vehicles can communicate with mesh nodes is defined and indexed by $\mathcal{V} : \{1, 2, ..V\}$.

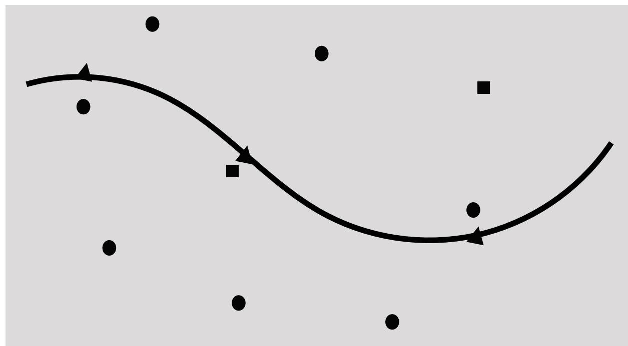
The mesh network and vehicular carriers are modeled as a graph $G(\mathcal{S}, \mathcal{L})$ in which $\mathcal{S}$ is the set of nodes, each is mapped to each of the members of $\mathcal{N}, \mathcal{B}$, and $\mathcal{V}$ and $\mathcal{L}$ is the set of all pairs (i, j), where $i, j \in \mathcal{S}$, and is defined as

$$\mathcal{L} = \{(i, j)|i, j \in \mathcal{S}, P_{T,ij} \leq \min(P_{T,Max,i}, P_{T,Max,j}),\text{ or } i, j \in \mathcal{V}\} \tag{4.1}$$

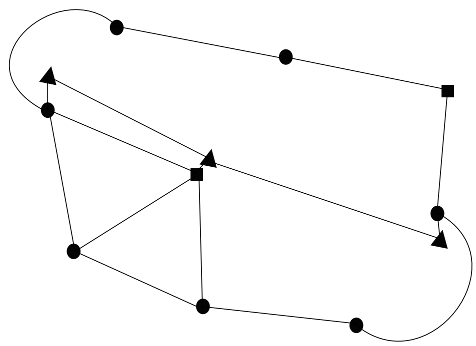
where $\mathcal{L}$ indicates the edges of the graph G. Figure 4.2 is an example of the way graph G forms. Figure 4.3(a) is an example of small mesh network with 7 mesh nodes and 2 base stations and Figure 4.3(b) is the modeled graph for the shown network. The solid lines are the edges of the graph. The set $\mathcal{N}$ includes the circles and sets $\mathcal{B}$ and $\mathcal{V}$ include the squares and triangular respectively.

$P_{T,Max,i}$ is the maximum transmission power of Node i, for $i \in \mathcal{N} \cup \mathcal{B}$, and when $i \in \mathcal{V}$, it is the maximum vehicular transmit power. By this definition all the nodes which are able to communicate directly have an edge in the graph. It is obvious that all the nodes in $\mathcal{V}$ are able to communicate with each other directly through vehicular message carrying.

For each link in the graph, a delay stamp, $D_{ij}(t)$, is defined which gives the expected maximum delay for delivering data directly from Node i to Node j in each



(a) An example of a wireless mesh network. The triangles are the virtual nodes and the circles and squares are mesh nodes and basestations respectively.



(b) The modeled graph for the example mesh network. The solid lines are the edges of the graph and the circles and square and triangles are the vertices of the graph.

Figure 4.3: An example of modeling a mesh network with a graph of nodes and edges.

time slot t. When $i \in \mathcal{N} \cup \mathcal{B}$ and $j \in \mathcal{V}$, this includes the waiting time for finding a vehicle for forwarding, when $i, j \in \mathcal{V}$ this is the maximum time for a vehicle to travel from Position i to Position j, and in the other cases it is the comparatively short time to directly relay the packet from one node to another.

4.2.1 The Bound Formulation

To formulate the bound we define the set of nodes which can communicate directly to node i as:

$$\mathcal{S}'_i = \{j | i, j \in \mathcal{S}, (i, j) \in \mathcal{L}\} \tag{4.2}$$

To incorporate the flow continuity, available bandwidth of each link and power consumption of each node into the formulation. For each possible mesh network flow, we assign an integer in the set $\mathcal{F} : \{1, 2, .., F\}$ while $Source(k)$ and $Dest(k)$ are functions giving the source and destination of Flow k, respectively, and $Flw(k, t)$ is the corresponding flow volume at time slot t. The flow continuity for each Node $i \in \mathcal{N} \cup \mathcal{V}$ can be written as

$$\sum_{j \in \mathcal{S}'_i} x^k_{ij}(t) - \sum_{j \in \mathcal{S}'_i} x^k_{ji}(t) = \begin{cases} Flw(k,t) & i = Source(k) \\ -Flw(k,t) & i = Dest(k) \\ 0 & \text{otherwise.} \end{cases} \tag{4.3}$$

where $x^k_{ij}(t)$ is the portion of Flow k which is routed from Node i to Node j in time slot t. The first term on the left hand side of Equation (4.3) is the total outgoing volume of Flow k from Node i at time slot t, and the second term on the left hand side is the total incoming volume of Flow k at time slot t. The right hand side is the volume of Flow k, which is generated in Node i and is $-Flw(k)$ if it is the flow destination and zero otherwise. Since all of the base stations are modeled as a single node, the flows destined for, or originating from, the base stations can be sent out from or delivered to each. The flow continuity constraint for the base stations can be written as

$$\sum_{i\in\mathcal{B}}(\sum_{j\in\mathcal{S}'_i} x^k_{ij}(t) - \sum_{j\in\mathcal{S}'_i} x^k_{ji}(t)) = \begin{cases} Flw(k,t) & \text{if} \quad Source(k) \in \mathcal{B} \\ -Flw(k,t) & \text{if} \quad Dest(k) \in \mathcal{B} \\ 0 & \quad \text{otherwise.} \end{cases} \tag{4.4}$$

To account for link capacity constraints, the total amount of traffic which is routed through each link should not exceed the capacity of the channel, i.e.,

$$0 \leq \sum_{k\in\mathcal{F}} x^k_{ij}(t) \leq C_{ij}, \quad (i,j) \in \mathcal{L} \tag{4.5}$$

where C_{ij} is the normalized available bandwidth for Link (i,j). The summation gives us the total amount of the traffic which is routed through Node i to Node j. As stated, the flows are assumed to be inseparable. Thus we define a decision variable u^k_{ij} to force this assumption, i.e., u^k_{ij} is a binary variable defined as

$$u^k_{ij}(t) = \begin{cases} 1 & \text{if} \quad x^k_{ij}(t) > 0, (i,j) \in \mathcal{L} \\ 0 & \text{if} \quad x^k_{ij}(t) = 0, (i,j) \in \mathcal{L} \end{cases} \tag{4.6}$$

The inseparability of flows for each Flow k and Node i is modeled by

$$\sum_{j\in\mathcal{S}'_i} u^k_{ij}(t) \leq 1 \tag{4.7}$$

$$\sum_{i \in \mathcal{B}} \Big(\sum_{j \in \mathcal{S}'_i} u^k_{ij}(t) \Big) \leq 1 \tag{4.8}$$

The first inequality indicates that flows cannot split while traveling from each Node $i \in \mathcal{N}, \mathcal{V}$ and the Inequality (4.8) guarantees flow inseparability coming from the set of basestations. The load on each Node $i \in \mathcal{N}$ at time t can be computed by using the modeling we introduced so far, i.e.,

$$\begin{aligned} L_i(t) = & \sum_{k \in \mathcal{F}} \sum_{j \in \mathcal{S}'_i} (x^k_{ij}(t) P_{t,ij} + x^k_{ji}(t) P_{r,ij}) \\ & + P_{idle} \sum_{j \in \mathcal{S}'_i} (I_{ij}(t) + I_{ji}(t)) \end{aligned} \tag{4.9}$$

$$I_{ij}(t) = \frac{C_{ij} - \sum_{k \in \mathcal{F}} x^k_{ij}(t)}{C_{ij}} \tag{4.10}$$

The first term on the right hand side of Equation (4.9) is the total energy which Node i uses at time t to transmit and receive, and the second term is the amount of energy during idle periods, for time t. The idle period is obtained by computing the utilization factor of a link in Equation (4.10).

We assume that the total maximum delay of a flow from source to its destination is the sum of the maximum delay stamps of the links that the flow is routed through. This total expected delay must be less than the expected delivery time of the flow which is primarily determined when it is sent out from the source based on its priority. This constraint for each Flow $k \in \mathcal{F}$ can be mathematically stated as follows

$$\sum_{(i,j) \in \mathcal{L}} u^k_{ij}(t) D_{ij}(t) \leq y^k(t) \quad \forall k \in \mathcal{F}, \tag{4.11}$$

in which $y^k(t)$ is the delay constraint for Flow k in time slot t. The left hand side of Inequality (4.11) calculates the summation of all the delay stamps of the links which are participating in the routing of Flow k at time slot t. The mesh nodes are assumed to be battery powered and their battery usage is modeled in discrete time. The network lifetime is divided into time slots and the amount of remaining battery for each node, $B_i(t)$, is updated at the end of each time epoch, according to the energy recursion:

$$B_i(t) = \max(B_i(t-1) - L_i(t), B_{outage}), \quad i \in \mathcal{N}. \tag{4.12}$$

It can be seen that Equation (4.6) is not linear, but can be linearized as follows.

$$u_{ij}^k(t) \in \{0,1\}, \quad x_{ij}^k(t) \le u_{ij}^k(t) < 1 + x_{ij}^k(t) \tag{4.13}$$

The above model gives a linear mixed-integer optimization. Two different objectives are used. The first minimizes the total energy usage of the network and is formulated as follows.

$$\text{minimize} \quad \sum_{i \in \mathcal{N}} \sum_t L_i(t) \tag{4.14}$$

subject to:

$$u_{ij}(t) \in \{0,1\}$$

$$0 \le \sum_{k \in \mathcal{F}} x_{ij}^k(t) \le C_{ij}, \quad (i,j) \in \mathcal{L}$$

$$x_{ij}^k(t) \leq u_{ij}^k(t) < 1 + x_{ij}^k(t)$$

$$\sum_{j \in \mathcal{S}_i'} u_{ij}^k(t) \leq 1$$

$$\sum_{i \in \mathcal{B}} (\sum_{j \in \mathcal{S}_i'} u_{ij}^k(t)) \leq 1$$

$$\sum_{j \in \mathcal{S}_i'} x_{ij}^k(t) - \sum_{j \in \mathcal{S}_i'} x_{ji}^k(t) = \begin{cases} Flw(k,t) & \text{if} \quad i = Source(k) \\ -Flw(k,t) & \text{if} \quad i = Dest(k) \\ 0 & \quad \text{otherwise.} \end{cases}$$

$$\sum_{i \in \mathcal{B}} (\sum_{j \in \mathcal{S}_i'} x_{ij}^k(t) - \sum_{j \in \mathcal{S}_i'} x_{ji}^k(t)) = \begin{cases} Flw(k,t) & \text{if} \quad Source(k) \in \mathcal{B} \\ -Flw(k,t) & \text{if} \quad Dest(k) \in \mathcal{B} \\ 0 & \quad \text{otherwise.} \end{cases}$$

$$\sum_{(i,j) \in \mathcal{L}} u_{ij}^k(t) D_{ij}(t) \leq y^k(t) \quad \forall k \in \mathcal{F}$$

$$L_i(t) = \sum_{k\in\mathcal{F}}\sum_{j\in\mathcal{S}_i'}(x_{ij}^k(t)P_{t,ij} + x_{ji}^k(t)P_{r,ij}) \\ +P_{\text{idle}}\sum_{j\in\mathcal{S}_i'}(I_{ij}(t) + I_{ji}(t))$$

$$I_{ij}(t) = \frac{C_{ij} - \sum_{k\in\mathcal{F}} x_{ij}^k(t)}{C_{ij}}$$

$$B_i(t) = B_i(t-1) - L_i(t), \quad i \in \mathcal{N}$$

$$B_{outage} \leq B_i(t)$$

The routing with this first objective is equivalent to shortest *path* routing if we define the length of a *path* as the distance that flows traverse in the mesh network, i.e., omitting the distance that they are carried by the vehicles. If we define the energy efficiency from the aspect of lengthening the lifetime of the network, the second objective can be used. Considering network lifetime as the time when the first node depletes its battery reserves, the second objective can be formulated as

$$\text{minimize} \quad \max_{i\in\mathcal{N}}\sum_t L_i(t), \tag{4.15}$$

to maximize lifetime, and which replaces the objective function in Equation (4.14). This minimizes the maximum battery usage of the nodes and extends the lifetime of the network.

The above formulation results in an optimum bound which can be used as a basis for comparison with practical forwarding algorithms. An algorithm for this is proposed in the next section.

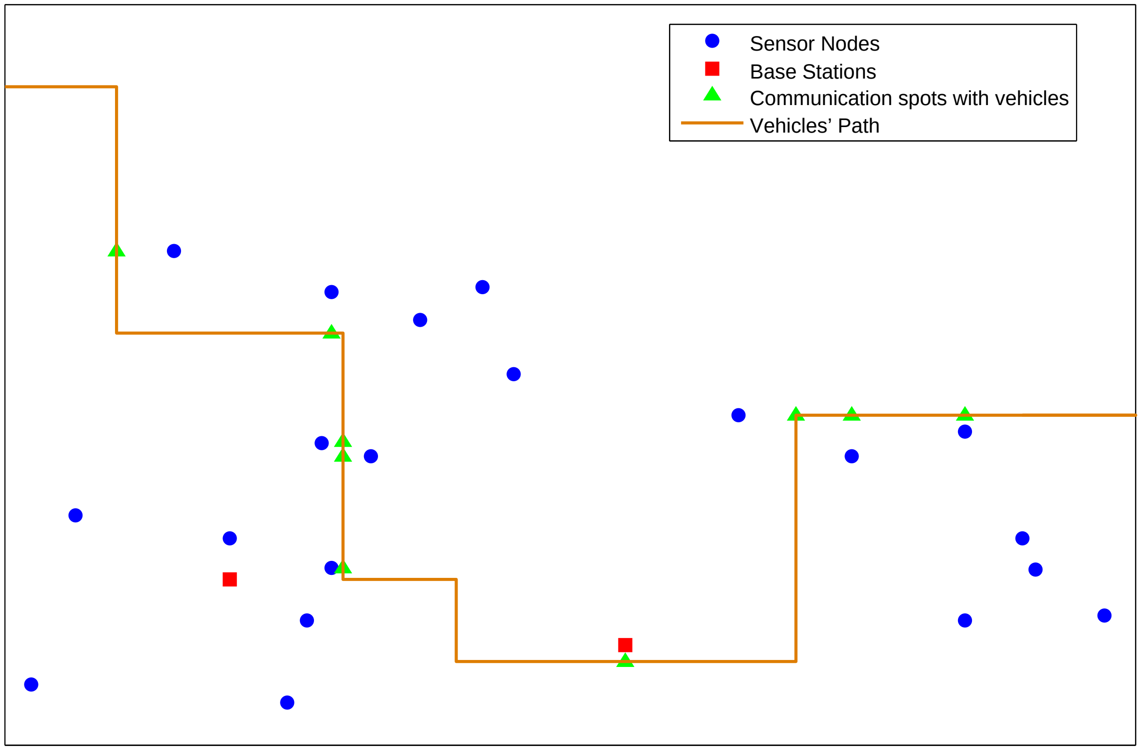


Figure 4.4: A 20 node network example with 2 base stations and a predefined vehicular path.

4.2.2 Heuristic Forwarding Algorithm

The optimization problem in the previous section is a linear integer program involving a large number of integer decision variables, making it infeasible for large problem instances. In addition, the optimization is in general a non-realizable bound since the node traffic flows are considered in a non-causal manner. To address this issue, we propose a heuristic which can be used in practical networks. In this greedy algorithm, the optimal route is designed based on the current and previous data flows for the current time slot. An optimization is used in each time slot, and therefore the size of the problem is equal to the number of time slots, T. For the first objective, i.e., shortest path, the algorithm solves the same optimization as in the previous section

Algorithm 1 Optimum OVAF routing

```
TBU(n) = 0, ∀n ∈ 𝒱
for t = 1 to T do
    min(max(TBU(n) + BU(n, t))
    TBU(n) ← TBU(n) + BU(n, t), ∀n ∈ 𝒱
end for
```

with the deployment lifetime of 1 time slot and using the traffic flow and flow delay constraints during that time slot. For the second objective we consider the algorithm shown as Algorithm 1, where $TBU(n)$ is the total battery usage of Node n in the previous time slots and $BU(n,t)$ is an array of variables indicating the battery usage of each node in the time epoch t. This algorithm contains a for loop which solves an optimization problem for each time slot. So in each iteration a set of the variables corresponding to battery usage of the nodes in the current time slot is determined by solving the optimization problem. The objective which is minimized is the maximum battery usage of the nodes by the end of the current time slot given the battery usage of the nodes in previous time slots.

4.3 The Packet Based Model

In this model the information units earmarked for routing in the WMN are packets. In contrast to the flow based model where the packets corresponding to each source-destination network routes are grouped and treated as flows, in this model each packet is assigned its own delay constraint and is routed separately through the network. Although this model is more complex and has many more variables to solve, it nevertheless provides more freedom to explore OVAF in detail. The most important feature of this model is its ability to incorporate the exact flow of the vehicles and helps measure the accuracy of the model under different vehicles' arrival assumptions. It also enables us to consider the effect of queuing packets in highly congested networks.

In the remainder of the section, all the notations which are used in the description of the model are listed and subsequently the bound for the model is formulated. Then, four different heuristics are introduced in order to approximate the bound with an acceptable precision and sufficiently low complexity.

4.3.1 Notation

$n \in \mathcal{N}$	Set of mesh nodes
$b \in \mathcal{B}$	Set base stations
$\mathcal{V}_g$	Set of virtual nodes on the vehicular path which connect gateways to the vehicular path
$\mathcal{V}_l$	Set of virtual nodes in the graph used for breaking the link
$\mathcal{S}$	set of all nodes in the graph
$p \in \mathcal{P}$	Set of packets during the deployment
$(i,j) \in \mathcal{L}$	Set of all links in the modeled graph G
T_a	Total number of time slots (clocks) during the network deployment
N_a	Number of all nodes in the modeled graph G
N_n	Number of mesh nodes
N_b	Number of base stations
N_v	Number of virtual nodes in the constructed modeling graph

$R_p(t)$	Remaining delivery allowance time of packet p in the time slot t, i.e., the maximum number of the time slots the packet is allowed to spend on the way to the destination.
G_p	Number of the time slot in which packet p is generated
DD_p	Number of time slots which is the delivery deadline of packet p, i.e., packet p should be delivered by that time slot
$C_{ij}(t)$	Capacity of link (i,j) at time t
Pr_p	Priority ratio of packet p
$u_{ij}^p(t) \in \{0,1\}$	the binary decision variable which determines whether packet p uses link (i,j) at time t
$BU_i(t)$	Battery usage of node i up to time t
D_{ij}	The expected delay stamp of link (i,j)
$y_p \in \mathcal{Y}$	the k^{th} delay category class corresponding to packet p
S_t	The time clock step for calculating the Energy Efficient Path Table (EEPT)
$M_{ij}(y)$	The best next move from node i to destination j corresponding to the delay category $y \in Y$
$P_{sd}(y)$	The set of sorted nodes forming a path from node s to node d such that the expected delivery delay of the path lies in the delay category range of $y \in Y$
DT_p	The maximum number of time slots that packet p can tolerate for delivery: $DT_p = G_p - DD_p$

q_j^0	Queue of the unscheduled packets in node j. It contains the generated packets at the current time slot in j and also the packets which have entered the node in previous time slot
S_r	the time step by which the queues are being resorted and the packets rechecked to see if they are behind the schedule and need rescheduling.

Table 4.1: Notations

A list of all the notation which is used in this section is shown in Table 4.1.

4.3.2 Modeling and Lower Bound Formulation

In this section the model that is used for packet routing is introduced and then a linear integer program problem for this model is formulated.

In the packet based model, the time clock of the network is considered as the time unit in which decisions are made. Each mesh node generates packets destined for the base stations and there are also controlling packets sent from the base stations to the mesh nodes. Each data packet p, has four basic characteristics which are namely: source(Sr_p), destination(Dt_p), generation time (GT_p) and delivery deadline (DD_p). The tolerable delivery delay of each packet is DT_p and is obtained by subtracting the generation time from the delivery deadline: $DT_p = DD_p - GT_p$.

The flow of passing vehicles in the vehicular path can be modeled as a random process. Clearly, depending on the problem conditions, different traffic models may apply for different vehicular traffic. A Poisson random process is one which applies to vehicular traffic if the traffic flows freely and no congestion exists. Accordingly, by assuming such conditions in the proposed model, the Poisson process is used for the vehicular traffic. This is accomplished by assuming approaching vehicles on both sides of the road, such that their inter-arrival time gaps are distributed according to the exponential distribution.

Similar to the flow based model, in this case the mesh nodes and base stations are identified by integers in the sets $\mathcal{N} : \{1, 2, .., N\}$ and $\mathcal{B} : \{1, 2, .., B\}$. Each road position where vehicles can communicate with mesh nodes is defined as a virtual node and indexed by integers in the set $\mathcal{V}_g : \{1, 2, ..V_g\}$. $\mathcal{V}_g$ is the set of virtual gateways to the WMN in the vehicular path. A set of pairs is also defined and indexed by $\mathcal{L}' : \{1, 2, ..L'\}$, corresponding to the links between those nodes which are able to communicate with each other directly. This set also includes self directed links, i.e., a node can communicate with itself and can maintain the packets in its memory without sending it to other adjacent nodes.

Since the time slot in this model is much smaller compared to the flow based model, the packets may be in the middle of a link at the beginning of a time slot. Considering that the formulation of the model requires the packets to be in nodes at such instances, another set of virtual nodes are defined for this model: $(\mathcal{V}_0)$. They are obtained by breaking existing links into smaller pieces such that all the links in the graph become small enough and allow the packets to traverse them in one time slot. The new broken links are indexed and shown by the set $\mathcal{L} : \{1, 2, ..L\}$.

Accordingly, the network is modeled as a graph with nodes and edges: $G(\mathcal{S}, \mathcal{L})$. $\mathcal{S}$ is the set of the nodes and is easily formed by the union of the sets $\mathcal{N}$, $\mathcal{B}$, $\mathcal{V}_g$ and $\mathcal{V}_0$ and the edges of the graph correspond to the set $\mathcal{L}$ which is defined in the previous paragraph. The set $\mathcal{S}'_i$ is also defined with the same definition of the Equation 4.2.

Having this graph as the model for the network structure, the packet continuity in node $i \in \mathcal{N} \cup \mathcal{V}$ is formulated by:

$$\sum_{j \in \mathcal{S}'_i} u^p_{ij}(t) - \sum_{j \in \mathcal{S}'_i} u^p_{ji}(t+1) = \begin{cases} 1 & \text{if} \quad i = Sr_p \quad \& \quad t = GT_p \\ -1 & \text{if} \quad i = Dt_p \quad \& \quad t = DD_p \\ 0 & \quad \text{otherwise.} \end{cases} \tag{4.16}$$

in which $u^p_{ij}(t) \in \{0,1\}$ is an integer decision variable indicating whether packet p is using link ij at time slot t. And for packet continuity in the base stations we have:

$$\sum_{i \in \mathcal{B}} \Big(\sum_{j \in \mathcal{S}'_i} u^p_{ij}(t) - \sum_{j \in \mathcal{S}'_i} u^p_{ji}(t+1) \Big) = \begin{cases} 1 & \text{if} \quad Sr_p \in \mathcal{B} \quad \& \quad t = GT_p \\ -1 & \text{if} \quad Dt_p \in \mathcal{B} \quad \& \quad t = DD_p \\ 0 & \quad \text{otherwise.} \end{cases} \tag{4.17}$$

To account for the inseparability of the packets which are being routed we have:

$$\sum_{j \in \mathcal{S}'_i} u^p_{ij}(t) \leq 1 \tag{4.18}$$

$$\sum_{i \in \mathcal{B}} \Big(\sum_{j \in \mathcal{S}'_i} u^p_{ij}(t) \Big) \leq 1 \tag{4.19}$$

where inequality 4.18 is for inseparability of packets in the nodes and inequality 4.19 is the term for the base stations.

To incorporate the capacity limit of the links, the number of packets which are routed in a specific time slot t should be limited to the capacity of that link. This condition is expressed as:

$$0 \leq \sum_{k \in \mathcal{P}} u_{ij}^{p}(t) \leq C_{ij}(t), \quad (i,j) \in \mathcal{L} \tag{4.20}$$

In the above Inequality, $C_{ij}(t)$ is the capacity of link (i, j) at the time t. This capacity depends on the type of link. When $i \in \mathcal{N} \cup \mathcal{B}$ and $j \in \mathcal{V}_g$ the $C_{ij}(t)$ will be zero when no vehicle is in the communication range of node i.

The load on each node is easily calculated by incorporating some small changes in the one we had in flow based model.

$$\begin{aligned} L_i(t) = \sum_{p \in \mathcal{P}} \sum_{j \in \mathcal{S}_i'} \Big(& u_{ij}^{p}(t) P_{t,ij} w_p + u_{ji}^{p}(t) P_{r,ij}) w_p \\ & + P_{idle} \sum_{j \in \mathcal{S}_i'} (u_{ij} I_{ij}(t) + u_{ji} I_{ji}(t) \Big) \end{aligned} \tag{4.21}$$

$$I_{ij}(t) = \frac{C_{ij}(t) - \sum_{k \in \mathcal{F}} u_{ij}^{k}(t)}{C_{ij}} \tag{4.22}$$

We will have the same model for the batteries of the mesh nodes as we had for the flow based model:

$$B_i(t) = \max(B_i(t-1) - L_i(t), B_{outage}), \quad i \in \mathcal{N}. \tag{4.23}$$

In packet based modeling, we focus on the lengthening of the lifetime of the mesh network. Therefore the second objective that is used in the previous section is considered. This objective minimizes the maximum battery usage of the nodes. Consequently, defining the lifetime as the time by which all nodes are alive and participating in the network, the maximum lifetime of the network will be achieved.

The linear integer decision variable is formulated as follows:

$$\text{minimize} \quad \max_{i \in \mathcal{N}} \sum_{t} L_i(t), \tag{4.24}$$

subject to:

$$u_{ij}^p(t) \in \{0, 1\}$$

$$\sum_{j \in \mathcal{S}'_i} u_{ij}^p(t) - \sum_{j \in \mathcal{S}'_i} u_{ji}^p(t+1) = \begin{cases} 1 & \text{if} \quad i = Sr_p \quad \& \quad t = GT_p \\ -1 & \text{if} \quad i = Dt_p \quad \& \quad t = DD_p \\ 0 & \qquad \text{otherwise.} \end{cases}$$

$$\sum_{i \in \mathcal{B}} \Big(\sum_{j \in \mathcal{S}'_i} u_{ij}^p(t) - \sum_{j \in \mathcal{S}'_i} u_{ji}^p(t+1) \Big) = \begin{cases} 1 & \text{if} \quad Sr_p \in \mathcal{B} \quad \& \quad t = GT_p \\ -1 & \text{if} \quad Dt_p \in \mathcal{B} \quad \& \quad t = DD_p \\ 0 & \qquad \text{otherwise.} \end{cases}$$

$$\sum_{j \in \mathcal{S}'_i} u_{ij}^p(t) \leq 1$$

$$\sum_{i \in \mathcal{B}} \Big(\sum_{j \in \mathcal{S}'_i} u_{ij}^p(t) \Big) \leq 1$$

$$0 \leq \sum_{p \in \mathcal{P}} u_{ij}^p(t) \leq C_{ij}(t), \quad (i, j) \in \mathcal{L}$$

$$L_i(t) = \sum_{p\in\mathcal{P}}\sum_{j\in\mathcal{S}'_i}\Big(u^p_{ij}(t)P_{t,ij}w_p + u^p_{ji}(t)P_{r,ij})w_p$$
$$+ P_{idle}\sum_{j\in\mathcal{S}'_i}(u_{ij}I_{ij}(t) + u_{ji}I_{ji}(t)\Big)$$

$$I_{ij}(t) = \frac{C_{ij}(t) - \sum_{p\in\mathcal{P}} u^p_{ij}(t)}{C_{ij}}$$

$$B_i(t) = max(B_i(t-1) - L_i(t), B_{outage}), \quad i \in \mathcal{N}.$$

This linear integer program gives us the lower bound for the problem. However the number of integer decision variables in this program will become very large for an actual network. This is because the number of variables are proportional to the product of the number of packets, the number of links and the number of time slots. Hence, to reduce the complexity and maintain a causal algorithm, heuristics are proposed in the next section.

4.3.3 Heuristics

Four different heuristics are introduced in this section to approximate the bound presented in the previous section. All of these heuristics have a common structure which decides on the next move of the packets at the current time slot based on their current status, including their priority ratio, their current position and their source and destination. The decision for the next move is made from a table by giving the status of the packet. The table is called the Energy Efficient Path Table (EEPT) which is calculated at the beginning of the deployment and is renewed at each S_t time slots.

The four heuristics are different with each other in two aspects:

1. *The energy efficient path table:* There are two methods by which the EEPT is calculated. The first gives us the optimum EEPT based on its definition and the second is a heuristic method incorporating Dijkstra's Algorithm and approximates the optimum EEPT with lower complexity.

2. *The Next Move Algorithm (NMA):* This is the method which is used to extract the best next move for each packet from the EEPT. We use two different algorithms for obtaining the best next move. These two will be explained in detail in the rest of this chapter.

Combining the options for each of these two will result in four different heuristics with different complexities. Their performance is compared and with the lower bound in the Results chapter.

The EEPT

The EEPT is a table comprising cells corresponding to each possible source, destination and delivery delay range. Each cell contains a set of ordered nodes which in turn form an optimum energy efficient path for the corresponding source and destination, meeting a specific range of delay criteria. In other words, given any two nodes (including both actual and virtual nodes) as the source and destination and a delay range, the EEPT gives a path from the given source to the destination which satisfies the delay constraint and also minimizes the maximum battery usage of the wireless mesh nodes.

The optimum energy efficient path can be calculated by the flowing linear integer program:

$$\text{minimize} \quad \max_{i \in \mathcal{N}}(BU(i)) \tag{4.25}$$

Subject to:

1. *Flow continuity:*

$$\sum_{j \in N} u_{ij} - \sum_{j \in N} u_{ij} = \begin{cases} 1 & \text{if} \quad i = n \\ -1 & \text{if} \quad i = m \\ 0 & \text{if} \quad O.W \end{cases}$$

2. *Inseparability of Packets:*

$$\sum_{j \in N} u_{ij} \leq 1$$

3. *Battery Usage:*

$$BU(i) = BU_p(i) + \sum_{j \in \mathcal{S}'_i} u_{ij} PT_{ij}$$

4. *Delay Constraint*

$$\sum_{(i,j) \in \mathcal{L}} u_{ij} D_{(i,j)} \leq y_k$$

By defining five different delay categories which cover the possible delivery delay range of the packets, the above linear integer programing will be used to obtain the optimum EEPT using Algorithm 2. In which $P_{sd}^{y_k}$ is the optimum path from source to destination obtained by the Optimization 4.25 satisfying the k^{th} delay constraint category. The table is renewed with the updated list of nodes' battery usage with the

Algorithm 2 Obtaining the optimum EEPT

```
for s ∈ 𝒩 do
  for d ∈ 𝒩 do
    for k ∈ 1..5 do
      Source ← s , Dest ← d
      Solve the optimization problem (4.25) from Source to Dest
      Get the P_sd^k from the solved optimization problem.
    end for
  end for
end for
```

time step of S_t.

In order to reduce the above algorithm complexity, we can replace the linear integer optimization problem used in obtaining the EEPT, with a shortest path Dijkstra Algorithm [20] problem. The steps of the EEPT construction in this case is shown in Algorithm 3.

In the Dijkstra' Algorithm version the weight of each link ij is a function of the delay stamp of the link, delay category of the path and the battery usage of the node i.

Algorithm 3 Obtaining EEPT via the Dijkstra algorithm

```
for s ∈ 𝒩 do
  for d ∈ 𝒩 do
    for k ∈ 1..5 do
      Source ← s , Dest ← d
      Run Dijkstra's Algorithm from Source to Dest
      Obtain P_sd^k from the solved problem.
    end for
  end for
end for
```

The NMA

The two different NMAs that are presented in this thesis can be categorized under two major sets of routing schemes:

- Non-source routing (NSR)
- Source routing (SR)

The NSR-NMA is used as an acronym for the first algorithm and SR-NMA is used for the second.

1. *NSR-NMA:* This algorithm goes through the network and for each packet it checks for packet state and its allowed remaining delivery time. The corresponding delay category of the packet is determined based on the range of the remaining time to its delivery deadline. The current node in which the packet is queued is also considered as the source of the packet. Accordingly, in the next step, the cell in the EETP which is associated with the given delay category, source and destination of the packet is chosen. The first node in the optimum path given in the corresponding cell in the EEPT is then considered as the next move of the packet.

   To avoid forming closed loops, the chosen next move is checked with all nodes that the packet has been into so far. If there is a match, the algorithm will run again by omitting the resultant next move from the network nodes and search for another next move from the other allowed delay category. If no node is found, the packet will be sent back to the previous node.

2. *SR-NMA:* In this algorithm the path of each packet is determined at its generation time from the corresponding cell in the EEPT. The path of the packet

is not renewed at each node unless the packet falls behind its own schedule. A threshold value is introduced to determine whether the packet is on time or behind. This threshold is defined on the *latency time* of the packet k given by:

$$Lt_k = R_k(t) - ExR_k \tag{4.26}$$

in which the $R_k(t)$ is the remainder of the time to the delivery deadline of the packet in time t and ExR_k is the remaining expected delivery time if the packet continues on its current path. If $Lt_k < -0.1 \times R_k(t)$, the packet is considered behind its schedule. At this time, the source of the packet is changed to the current node in which it is now queued and based on the range of $R_k(t)$ its delay category is determined. Consequently, given the new delay category and source the corresponding path is extracted from the associated cell in the EEPT.

In general the SR-NMA has a better performance compared to the NSR-NMA, as it does not create closed loops in the network and thus has less overhead. The four heuristics resulting from a combination of the above methods, are shown in Algorithms 4 to 7.

4.3.4 Complexity Analysis

The linear integer program which yields the lower bound is an NP-hard problem i.e. the complexity of the problem grows exponentially with the size of the variables. This is mainly a result of large number of integer decision variables in actual networks. In fact, the number of integer variables in this problem are proportional to the product of the number of packets, the number of links and the number of time slots.

To cope with the complexity problem of the lower bound the proposed heuristics are devised. Their main computational cost comes from the EEPT forming stage. According to the heuristic algorithms to form the optimum EEPT for a network an optimization problem should be solved for each cell of the EEPT. The number of the integer decision variable in this optimization problem is proportional to the squared number of mesh nodes. This number is much less than the lower bound optimization integer decision variables since it does not include the number of packets and number of time clocks. Consequently, the complexity of achieving optimum EEPT grows slower with size of the network compared to the lower bound. However, it is still an NP hard problem which is a result of integer decision variables involvement.

The Dijkstra driven EEPT is a way to eliminate the integer decision variables problem and get away from NP-hardness of the problem. The worst-case running time for the Dijkstra algorithm on a graph with n nodes and l edges is $O(n^2)$. Since the EEPT has n^2 cells and each cell needs to run the Dijkstra algorithm once, the complexity of achieving the EEPT will be $O(n^4)$. Where n is the number of mesh nodes.

Algorithm 4 Heuristic with optimum EEPT and NSR-NMA method

for $t = 0$ to T_a **do**
  if $t \bmod S_t = 0$ **then**
    for $s \in \mathcal{N}$ **do**
      for $d \in \mathcal{N}$ **do**
        for $k \in 1..5$ **do**
          $Source \leftarrow s$, $Dest \leftarrow d$
          Solve the optimization problem from $Source$ to $Dest$
          Get the $M_{sd}(y_k)$ from the solved optimization problem.
        end for
      end for
    end for
  end if
  for $j = 1$ to N_a **do**
    $\forall P_k$ in Node:j:
    Calculate the R_k for the P_k
    Calculate the corresponding delay category of the packet: $y_k^{P_k} = f(R_k)$
    Calculate the $M_{j,Des}(y_k^{P_k})$, Des: Destination of P_k
    Put P_k in the queue corresponding the $M_{j,Des}(y_k^{P_k})$
  end for
  for $j = 1$ to N_a **do**
    Calculate the priority ratio for each packet: $Pr_k \ \forall P_k \in j : P_k \notin nq_j$
    Sort the packets in the queue based on their priority ratio $\forall P_k \in j : P_k \notin nq_j$
    Take the first $C_{ji}(t)$ packets in each queue in j except the nq_j and change the state of the packets from j to i
  end for
end for

Algorithm 5 Heuristic with EEPT driven via Dijkstra's Algorithm and NSR-NMA

for $t = 0$ to T_a **do**
  if $t \bmod S_t = 0$ **then**
    for $k \in 1..5$ **do**
      Calculate the cost matrix corresponding to y_k: $\mathcal{C}_k$ given the $BU_p(i)$, $i \in \mathcal{N}$
      for $s \in \mathcal{N}$ **do**
        for $d \in \mathcal{N}$ **do**
          $Source \leftarrow s$, $Dest \leftarrow d$
          Calculate the Dijkstra algorithm
          Get the $M_{sd}(y_k)$ from the solved Dijkstra
        end for
      end for
    end for
  end if
  for $j = 1$ to N_a **do**
    $\forall P_k$ in Node:j:
    Calculate the R_k for the P_k
    Calculate the corresponding delay category of the the packet: $y_k^{P_k} = f(R_k)$
    Calculate the $M_{j,Des}(y_k^{P_k})$, Des: Destination of P_k
    Put P_k in the queue corresponding the $M_{j,Des}(y_k^{P_k})$
  end for
  for $j = 1$ to N_a **do**
    Calculate the priority ratio for each packet: Pr_k $\forall P_k \in j : P_k \notin nq_j$
    Sort the packets in the queue based on their priority ratio $\forall P_k \in j : P_k \notin nq_j$
    take the fist $C_{ji}(t)$ packets in each queue in j except the nq_j and change the state of the packets from j to i
  end for
end for

Algorithm 6 Heuristic with optimum EEPT and SR-NMA

for $t = 0$ to T_a **do**
  if $t \bmod S_t = 0$ **then**
    for $s \in \mathcal{N}$ **do**
      for $d \in \mathcal{N}$ **do**
        for $k \in 1..5$ **do**
          $Source \leftarrow s$, $Dest \leftarrow d$
          Solve the optimization problem** from $Source$ to $Dest$
          Get the $\mathcal{P}_{sd}(y_k)$ from the solved optimization problem.
        end for
      end for
    end for
  end if
  for $j = 1$ to N_a **do**
    $\forall P_k$ in Node:j:
    if $P_k \in nq_j$ **then**
      Calculate the Delay tolerance: DT_k for the P_k
      Calculate the corresponding delay category of the the packet: $y_k^{P_k} = f(DD_k)$
      $s \leftarrow Source(P_k)$, $d \leftarrow Destination(P_k)$
      Calculate the $M_{s,d}(y_k^{P_k})$ from the $\mathcal{P}_{sd}(y_k)$
      Put P_k in the queue corresponding the $M_{s,d}(y_k^{P_k})$
    end if
  end for
  for $j = 1$ to N_a **do**
    if $t \bmod S_r = 0$ **then**
      Calculate the priority ratio: $Pr_k \; \forall P_k \in j : P_k \notin nq_j$
      Get the Expected Remaining Delivery Time $\forall P_k \in j : P_k \notin nq_j$: ExR_k
      Get the Remaining time to delivery deadline $R_k(t) \forall P_k \in j : P_k \notin nq_j$
      if $R_k(t) - ExR_k < -0.1 * R_k(t)$ **then**
        $Source(P_k) \leftarrow j$
        $G(P_k) \leftarrow t$
        Move P_k from the current queue to nq_j
      end if
      Sort the packets in each queue based on their priority ratio
    end if
    take the first $C_{ji}(t)$ packets in each queue and change the state of the packets from j to i
  end for
end for

Algorithm 7 Heuristic with EEPT driven via Dijkstra algorithm and SR-NMA method

for $t = 0$ to T_a **do**
  if t mod $S_t = 0$ **then**
    for $k \in 1..5$ **do**
      Calculate the cost matrix corresponding to y_k: $\mathcal{C}_k$ given the $BU_p(i)$, $i \in \mathcal{N}$
      for $s \in \mathcal{N}$ **do**
        for $d \in \mathcal{N}$ **do**
          $Source \leftarrow s$, $Dest \leftarrow d$
          Calculate the path from Source to Destination with Dijkstra algorithm
          Get the $\mathcal{P}_{sd}(y_k)$ from the Dijkstra algorithm result.
        end for
      end for
    end for
  end if
  for $j = 1$ to N_a **do**
    $\forall P_k$ in Node:j:
    if $P_k \in nq_j$ **then**
      Calculate the Delay tolerance: DT_k for the P_k
      Calculate the corresponding delay category of the the packet: $y_k^{P_k} = f(DD_k)$
      $s \leftarrow Source(P_k)$, $d \leftarrow Destination(P_k)$
      Calculate the $M_{s,d}(y_k^{P_k})$ from the $\mathcal{P}_{sd}(y_k)$
      Put P_k in the queue corresponding the $M_{s,d}(y_k^{P_k})$
    end if
  end for
  for $j = 1$ to N_a **do**
    if t mod $S_r = 0$ **then**
      Calculate the priority ratio: $Pr_k \ \forall P_k \in j : P_k \notin nq_j$
      Get the Expected Remaining Delivery Time $\forall P_k \in j : P_k \notin nq_j$: ExR_k
      Get the Remaining time to delivery deadline $R_k(t) \forall P_k \in j : P_k \notin nq_j$
      if $R_k(t) - ExR_k < -0.1 * R_k(t)$ **then**
        $Source(P_k) \leftarrow j$
        $G(P_k) \leftarrow t$
        Move P_k from the current queue to nq_j
      end if
      Sort the packets in each queue based on their priority ratio
    end if
    take the fist $C_{ji}(t)$ packets in each queue and change the state of the packets from j to i
  end for
end for

Chapter 5

Results

5.1 Introduction

This section presents simulation results for the two different modeling methods which are introduced in the previous chapter, namely,

- The flow based model
- The packet based model

Simulations are conducted for networks with two varying parameters:

- Size of the WMN
- Flow rate of vehicular traffic

The performance of the OVAF approach is investigated for different networks and the effect of each parameter on the network energy efficiency is explored. They are also compared to the results of the traditional multihop forwarding technique.

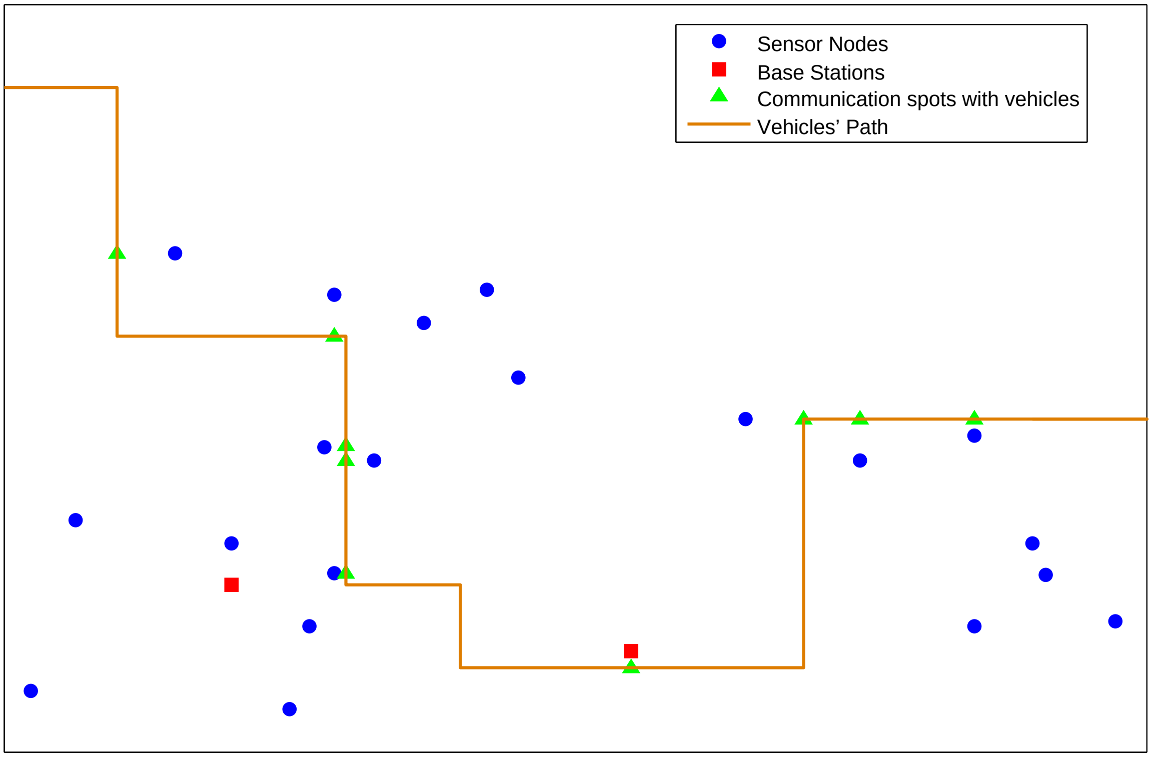


Figure 5.1: A 20 node network example with 2 base stations and a predefined vehicular path. The size of the deployment area is 100×90 meters.

The comparison shows a large increase in efficiency when using OVAF. The obtained results have also confirmed that the performance of OVAF decreases for sparse traffic flows.

5.2 The Flow Based Model

This section presents simulation results which evaluate the performance of the proposed algorithms under the flow based modeling.

The simulations are performed in mesh networks with different sizes where data flows originate at mesh nodes and terminate at base stations. Each flow, in each time slot, has a normalized value which is uniformly distributed between 0 and 0.5. The delay constraint which is associated with each flow randomly chosen from a uniform distribution between 0 and D_{max} where D_{max} is 100 unit of times. A path of vehicles also exits which is passing through the network. This path and the network deployment area are the same for all the simulated networks and are identical to Figure 5.1. Network deployment area is a rectangular of 100×90 meters. Mesh nodes which are in the communication range of this vehicular path are gateways to the vehicular network. The expected waiting time for each gateway node to find an available vehicle in each time slot varies from 10 unit of time to 80 unit of time depends on the time of the day. This amount determins the delay stamp associated with links between mesh gateway nodes and virtual nodes. The variation pattern of this delay stamp with the time in 24 hours of a day is shown in Figure 5.2. Also the expected carrying time for the vehicles to carry data from one gateway to another are calculated based on their expected speed which is 1.2 meter per unit of time.

It is assumed that vehicles do not stop or change their path in the middle of their way. To account for these variations an Automatic Repeat-Request (ARQ) protocol should be designed. Moreover, other concerns such as security issues are not discussed here. They can be considered as an extension to this thesis work since we are only focusing on the routing problem in the OVAF approach.

Figure 5.1 shows an example of a simulated network with 20 mesh nodes and 2 basestations. The nodes are randomly placed using a uniform distribution within the network area. The solid line indicates the vehicular path and the positions marked by triangles on the path indicate the corresponding nodes in $\mathcal{V}$. The simulations are performed for mesh networks with 10, 15, 20, 30 and 40 nodes and the two different

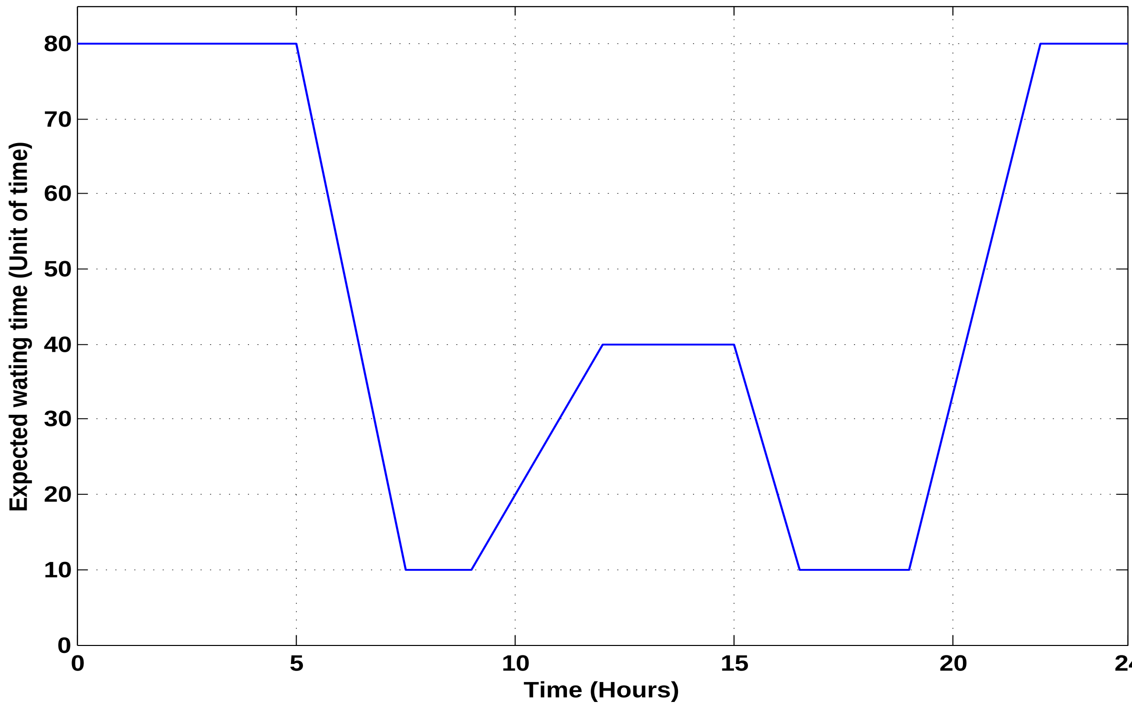


Figure 5.2: Expected waiting time of a mesh gateway node for an available vehicle with regards to the time in 24 hours of a day.

objectives, i.e., Shortest Path (SP), Maximum Lifetime (ML), and the heuristic for both of these objectives as previously discussed. The achieved results are the battery usage of the nodes and they are normalized to the unit of energy (J) in the figures.

The simulation results are compared with the results of conventional mesh networks which do not use opportunistic vehicular forwarding. We refer to this routing scheme as Multihop Mesh Forwarding (MMF) in the remainder of this chapter.

The accumulated load for each node under four different scenarios for a 10 node mesh network is shown in Figures 5.3 and 5.4, which gives results for the first hundred hours of network deployment time. The heuristic results using SP are shown in Figure 5.3 and the results for ML in Figure 5.4. Most of the nodes have to forward much less load in vehicle assisted forwarding and it can be seen that the loads are

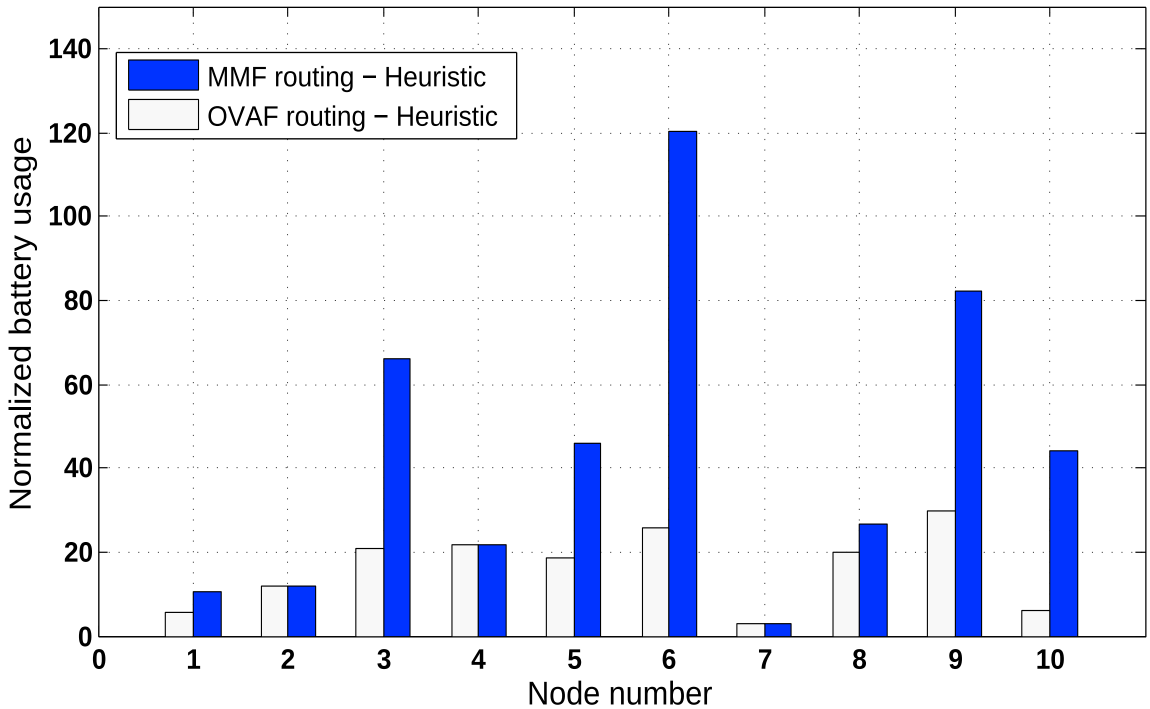


Figure 5.3: Battery usage in a 10 mesh node network after 100 hours of deployment time using the SP objective. The battery usage is normalized to the unit of energy (J).

more evenly distributed (See Table 5.2). The node, indexed seven, has a very low load burden in both since it does not participate in multihop mesh forwarding.

With the ML objective, the mesh network has to support a higher average load at the expense of longer network lifetime. Moreover, higher numbers of nodes deplete at the end of the network lifetime compared to the SP objective. It can also be seen that loading is more evenly distributed with the ML objective than the SP objective for both MMF and OVAF routing (See Table 5.2).

The average load can be compared in Figures 5.5 and 5.6 for both the optimization results and the heuristic with two objectives and higher numbers of nodes after 100 hours of deployment. For the SP objective, vehicular forwarding saves more than

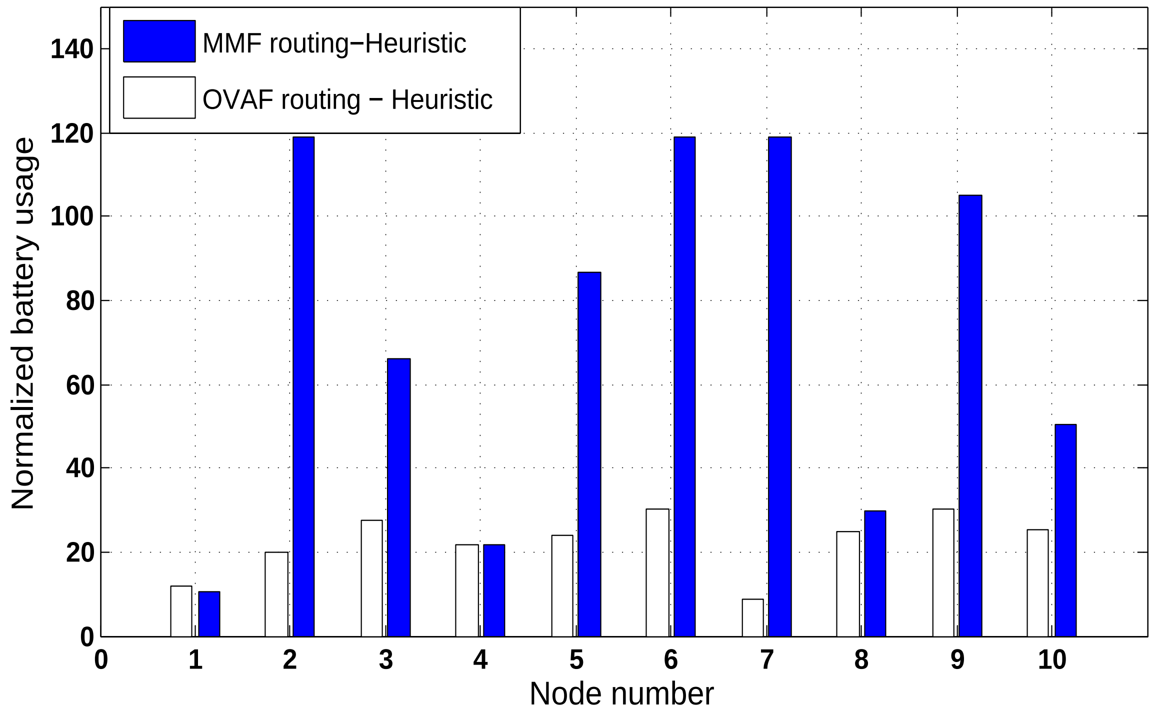


Figure 5.4: Battery usage in a 10 mesh node network after 100 hours of deployment time using the ML objective. The battery usage is normalized to the unit of energy (J).

60% of the energy compared to MMF operation. ML uses more resources for both the vehicular and non-vehicular cases, since its objective is to minimize the maximum use of resources. The fair load distribution in vehicular forwarding is also notable in the ML objective. Figure 5.7 shows the maximum battery usage of the heuristic and the optimum results for the ML objective and it can be seen how close the heuristic results are to the lower bound. An interesting observation by comparing the two figures is how much more average energy the LB has to use in MMF routing to slightly improve the ML objective while in OVAF routing the average energy usage of both optimum results and the heuristic are almost the same. It is also notable that the heuristic generates the exact same result as the optimum solution for the SP

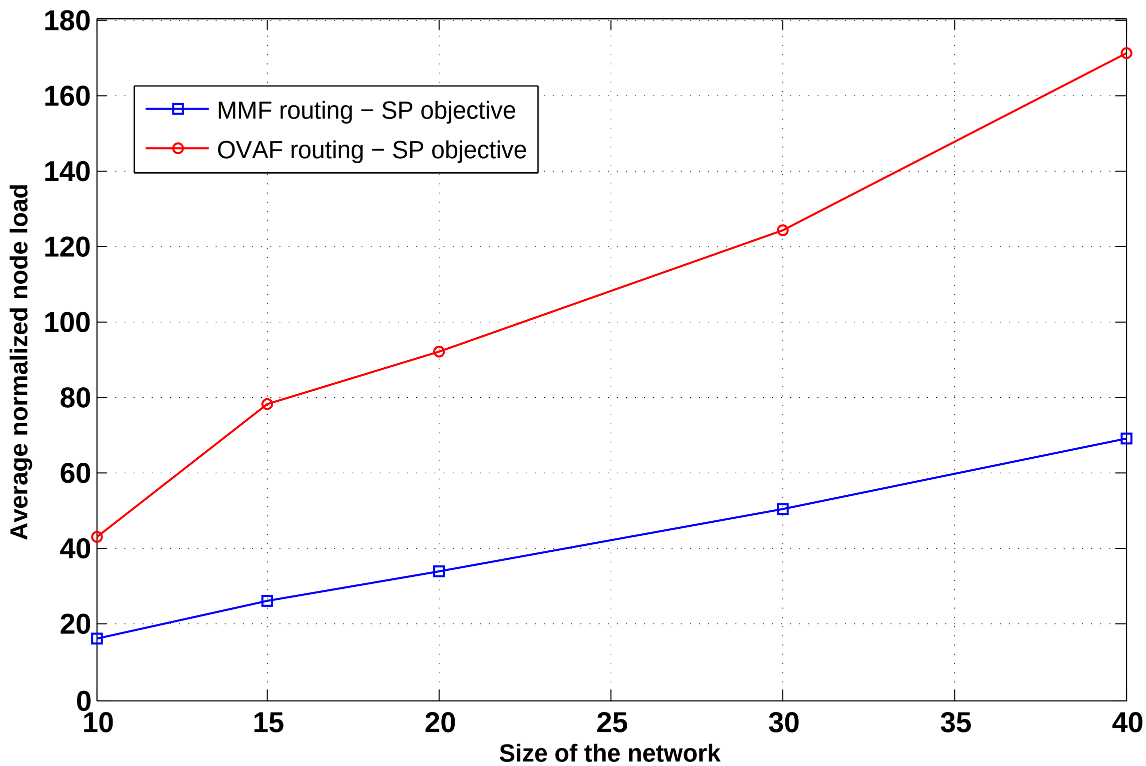


Figure 5.5: Average battery usage with the SP objective for different network sizes after 100 hours of deployment. The average battery usage is normalized to the unit of energy (J).

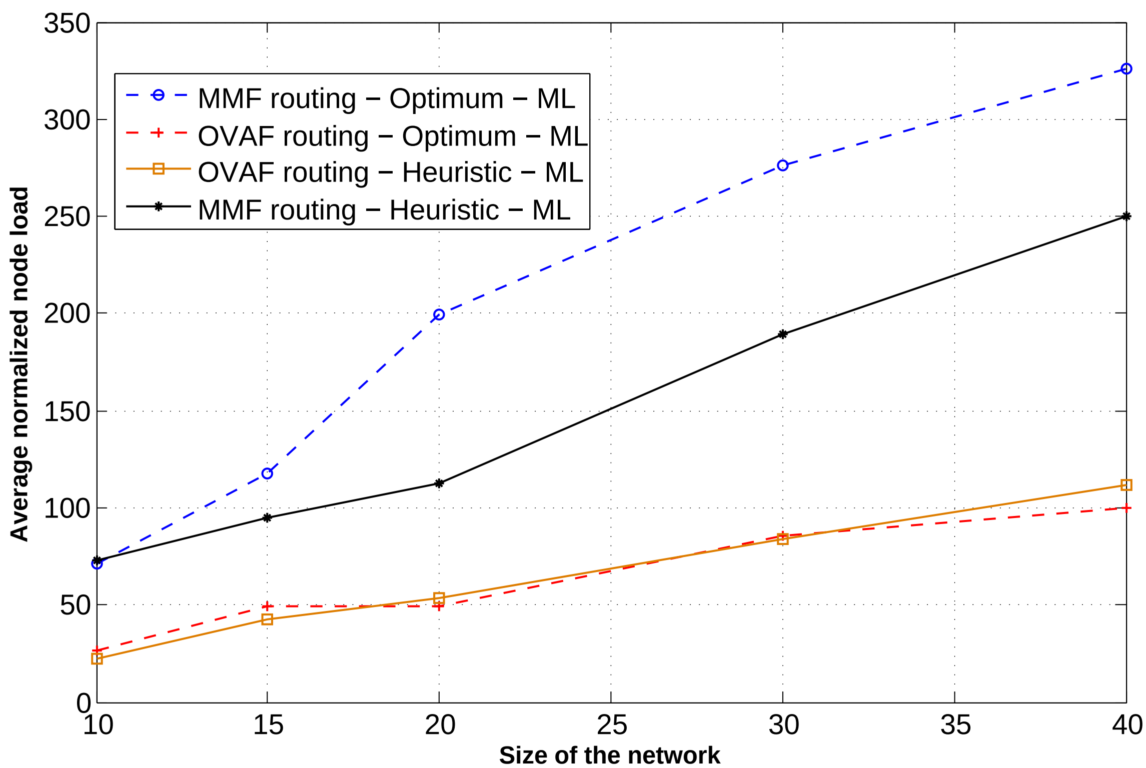


Figure 5.6: Average battery usage with the ML objective for different network sizes after 100 hours of deployment. The average battery usage is normalized to the unit of energy (J).

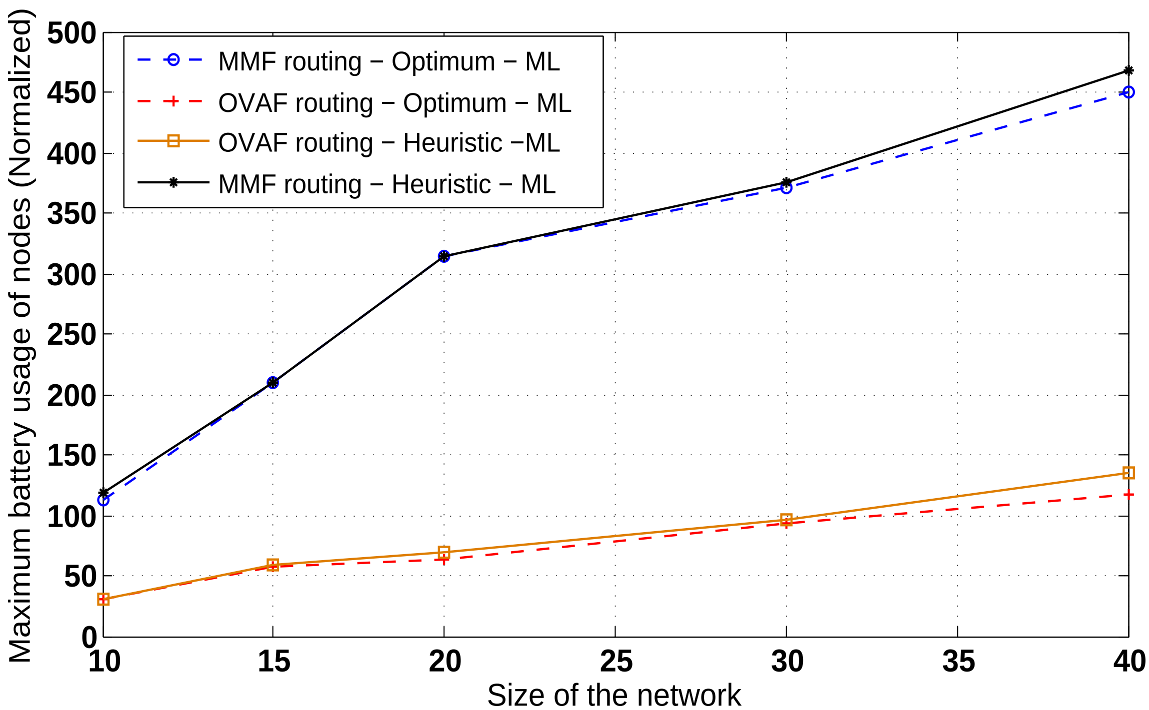


Figure 5.7: Maximum normalized battery usage of simulated networks with different sized under the objective of ML. The maximum battery usage is normalized to the unit of energy (J).

objective. This is a result of the objective which does not need to use the other time slot flow information to find the shortest path in each time epoch.

Results for the network lifetime are shown in Table 5.1. As can be seen, the vehicular forwarding case has much better performance than in the MMF case and the ML optimization performs better than the SP case as would be expected.

The optimum routing with the ML objective uses all the possible nodes no matter how long the route, to alleviate maximum load burdens. Consequently, it has much better average battery usage which leads to better network lifetime.

	Mechanism					
Network Size	SP-VA	SP-MMF	ML-VA-Opt.	ML-MMF-Opt.	ML-VA-H	ML-MMF-H
10	101	27	112	30	99	28
15	48	15	59	22	55	17
20	34	12	47	18	45	12
30	20	8	40	16	36	10
40	14	5	29	9	26	8

Table 5.1: Network lifetime with battery capacity of 30 units. Each unit of battery capacity is the amount of available energy in the battery normalized to the unit of energy (J).

To measure how fair the load is distributed through the nodes we calculate the normalized variance of the accumulated mesh node loading. These are compared in Table 5.2. The normalized variance also called coefficient of variance is calculated by:

$$c(X) = \frac{\sigma_X^2}{\overline{X}^2},$$

in which σ_X^2 is the variance of X and $\overline{X}$ is the average of the variable X. It can be seen that using the OVAF approach helps the network to distribute the load more evenly over the mesh nodes.

	Mechanism					
Network Size	SP-VA	SP-MMF	ML-VA-Opt.	ML-MMF-Opt.	ML-VA-H	ML-MMF-H
10	0.32	0.74	0.02	0.3	0.11	0.34
15	0.50	1.13	0.06	0.38	0.17	0.76
20	0.58	1.35	0.11	0.29	0.16	0.95
30	0.58	1.32	0.04	0.18	0.06	0.53
40	0.81	1.93	0.09	0.41	0.11	0.56

Table 5.2: Coefficient of variance of load after 100 deployment hours.

5.3 The Packet Based Model

In this model the basic data units are packets and the nodes are assumed to be uniformly distributed over the network field. The deployment time of the network is broken into the time slots which are equal to the system's time clocks. The data packets are generated in the mesh nodes, destined for the base stations and there are also controlling packets from the base stations to the mesh nodes. The packet generation pattern over time and the DT_p which is the maximum tolerable delivery delay of packets may be modelled with different statistical distributions. These distributions are dependent on the WMN usage specification and the area in which the nodes are placed. Similar to the flow based model nodes are randomly placed using a uniform distribution within the network area. The position of the vehicular path within the network area is the same as the solid line in Figure 5.1 and the positions marked by triangles on the path indicate the corresponding nodes in $\mathcal{V}_g$.

In the conducted simulations, a uniform distribution is considered for packets' distribution over time and the nodes. 10% of the packets are considered as instantaneous packets, i.e., the DT_p is less than a few time slots which don't allow the packets to use a vehicular path. However, it gives them enough freedom to choose an appropriate path through the WMN. The maximum tolerable delay (DT_p) for the rest of the packets is uniformly distributed from a few time slots to a maximum range which depends on the network's traffic type. In the conducted simulations this amount is set to 100 unit of times.

The vehicles are assumed to flow from opposite sides of the road and their inter-arrival gaps are randomly distributed in accordance with an exponential process. The rate of the achieved Poisson process is an indication of the sparsity of the vehicular traffic.

Heuristic 1	NSR, Optimum EEPT
Heuristic 2	NSR, EEPT via Dijkstra's Algorithm
Heuristic 3	SR, Optimum EEPT
Heuristic 4	SR, EEPT via Dijkstra's Algorithm

Table 5.3: The index number of each heuristic.

The optimization results under MinMax objective are achieved for networks with maximum size of 20 mesh nodes and over maximum deployment time of 100 time slots. The optimum solution for larger network size over a longer period of deployment time cannot be achieved in a reasonable time due to increasing complexity. The attained optimum results in small network sizes are compared with heuristics over the same networks. The results obtained from solving the optimization problem with MinMax objective are called the *optimum MinMax* results in this chapter. There are also index numbers considered for heuristics which are shown in Table 5.3.

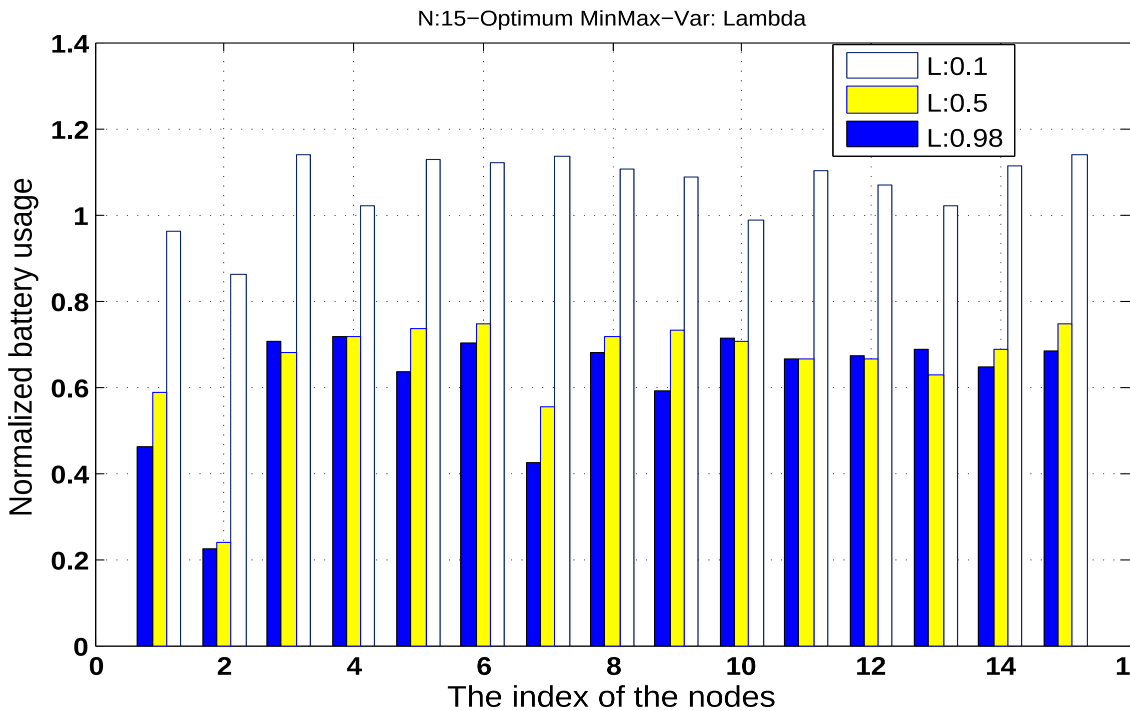


Figure 5.8: The optimum MinMax solution: Accumulated load burden of a simulated WMN with 15 nodes after 100 time slots of deployment for different traffic flow rates. The loads are normalized to the unit of energy (J).

Figure 5.8 shows the normalized battery usage of a WMN with 15 nodes after 100 time slots of deployment under different rates of vehicular traffic flows achieved by solving the optimization problem. It is a notable observation that increasing the rate of passing vehicles form 0.1 to 0.5 has a great impact on improving the energy efficiency and lengthening the lifetime of the system. On the other hand, the same amount of increase from 0.5 to 0.98 has less impact on energy efficiency. As it can be seen increasing the flow rate does not result in less battery usage in all the nodes. For example nodes indexed 3, 10 and 13 has less battery usage in traffic flow rate of 0.5 compared to traffic flow rate of 0.98 or the battery usage for both traffic flow rates of 0.5 and 0.98 are the same for nodes indexed as 11 and 4.

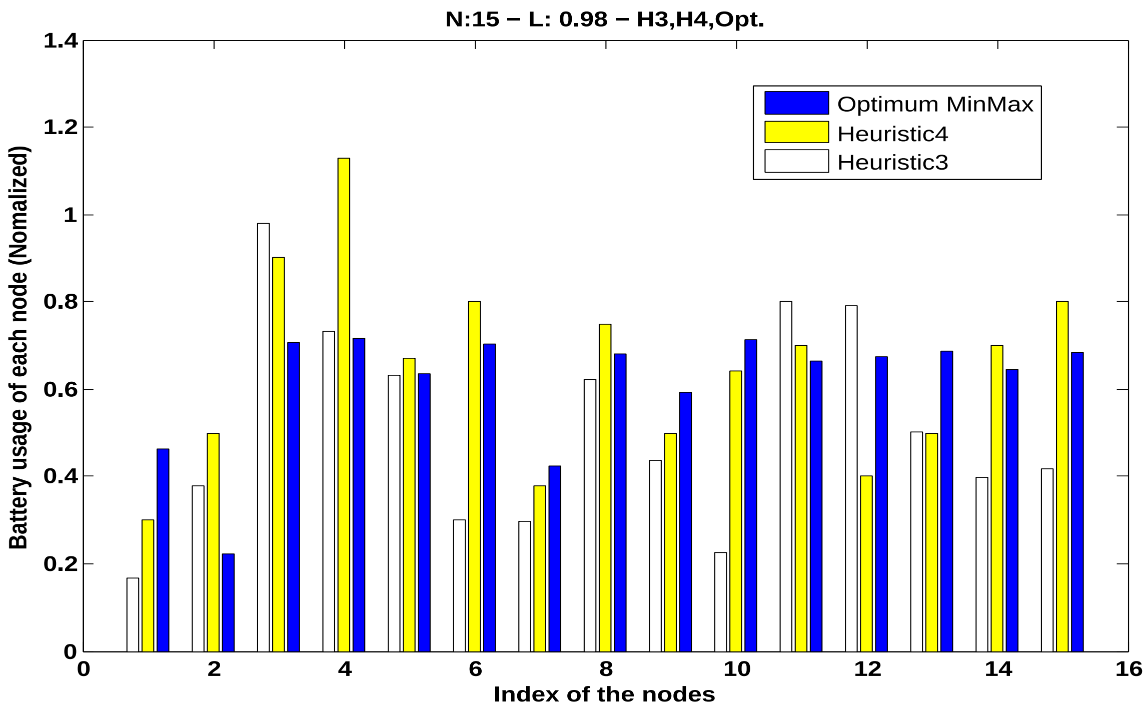


Figure 5.9: Comparison of optimum results for MinMax objective and heuristics 3 and 4 for a network with 15 nodes and traffic flow rate of 0.98 after 100 time slots of deployment. The loads are normalized to the unit of energy (J).

Figure 5.9 shows the obtained results for the network with 15 mesh nodes after 100 time slots of deployment for a vehicular traffic rate of 0.98. The normalized battery usage of the network nodes are compared separately for the optimum MinMax solution and heuristic 3 and heuristic 4. As it can be seen the heuristics can not be as efficient as the optimum solution. Heuristic 3 puts heavier load on nodes number 3 while the optimum MinMax solution has used the other nodes like node number 1 more often to rout the traffic and reduce the load of node 3. The same observation is true for heuristic 4 and the optimum MinMax solution. Heuristic 3 has better performance than heuristic 4 since the maximum battery usage of the nodes in heuristic 3 is less. This better performance is a result of using optimum EEPT rather than obtaining the EEPT from the Dijkstra algorithm.

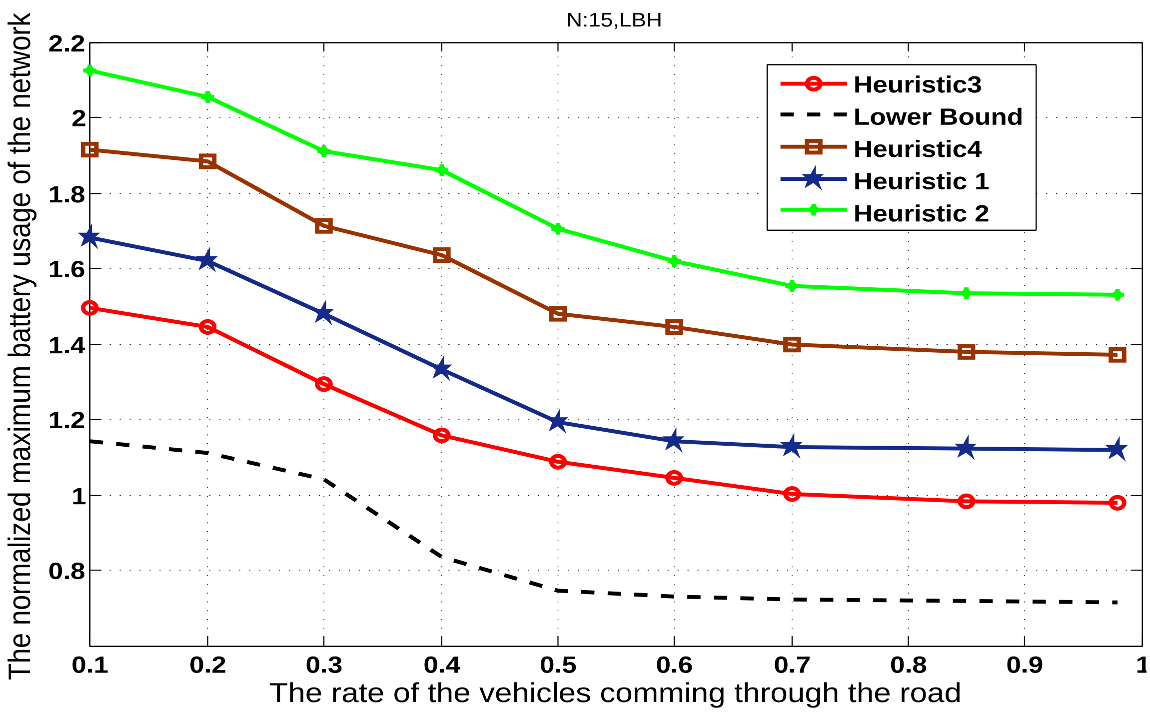


Figure 5.10: Comparison of the objective of lower bound and heuristics in a simulated WMN with 15 nodes and different traffic flows after 100 time slots of deployment. The maximum loads are normalized to the unit of energy (J).

The lower bound and the four different heuristics are compared with each other under different assumptions for vehicle flow rates in Figure 5.10. The number of mesh nodes are 15 and the deployment time is 100 time slots. Overall, a very dense traffic (rate= 0.98) helps the network to increase its energy efficiency over 30% compared to a very sparse traffic flow (rate= 0.1). Furthermore, on average, the NSR with the optimum EEPT has the closest performance to the lower bound. The SR method usually has a weaker performance. This is mainly a result of the closed loops, formed undesirably in the mesh network.

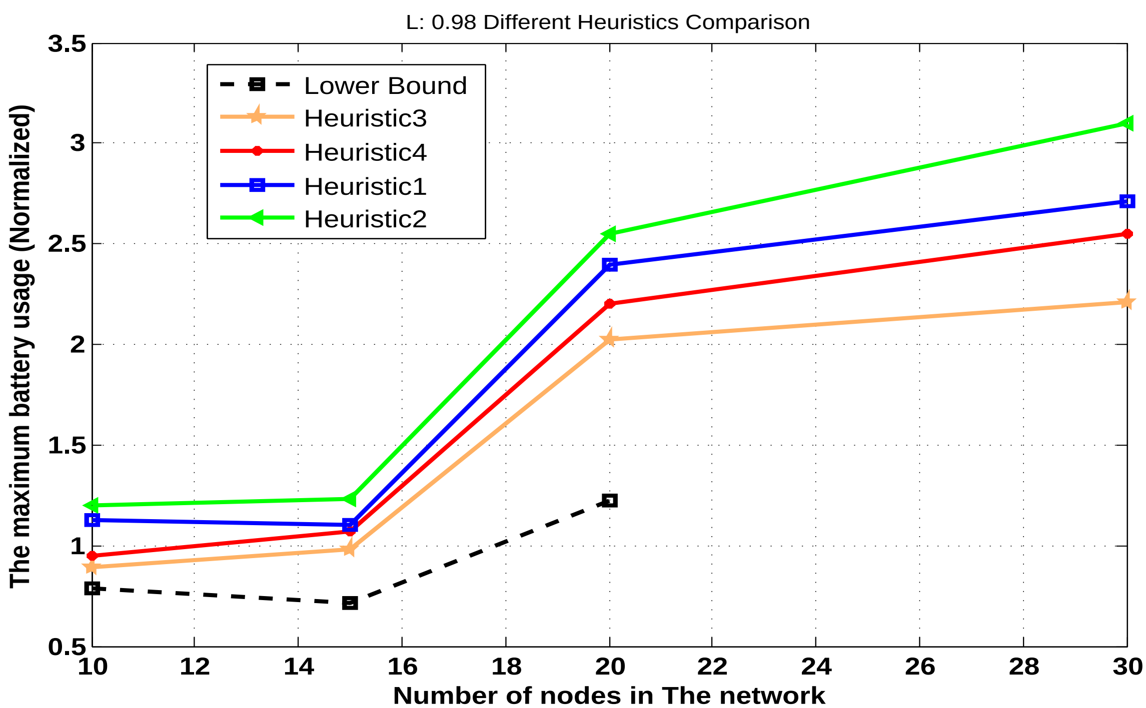


Figure 5.11: Comparison of the objective function (Maximum battery usage) for lower bound and heuristics in different network size and traffic flow rate of 0.98 after 100 time slots of deployment. The maximum loads are normalized to the unit of energy (J).

Figure 5.11 depicts the performance of the lower bound and the heuristics for different network sizes for a 100 time slot deployment time. The lower bound could not be obtained for networks over 20 nodes because of the increasing complexity. Heuristics are achieved for all sizes of mesh networks and their performance remain satisfactory for larger network sizes. By increasing the size of the network the maximum battery usage increases as there are more nodes generating packets over time. It can be seen that the priority of the performance of the heuristics remains the same for all the network sizes.

The average load of the simulated networks are also compared for the optimum MinMax solution and heuristics in Figure 5.12 to Figure 5.16. Note that the average that is depicted in these figures is not the objective of the optimization problem and the optimum MinMax solution does not give the optimum value for that. Nevertheless, the performance of different algorithms for the average load of the network have almost the same priority that they have for the objective function (maximum battery usage). In other words, the optimum solution has the minimum load burden in most of the cased and the heuristics are close to it in performance. In Figure 5.12, there is a gap between the performance of those heuristics using the optimum EEPT and those using Dijkstra algorithm driven one. Figure 5.13 and 5.14 show the average battery usage of the networks with different sizes for vehicle flow rate of 0.5 and 0.98 respectively. According to these figures optimum MinMax solution puts heavier average load on the network with the size of 20 nodes compared to the heuristic 3. Heuristics 4 and 2 have a close performance for different network sizes. It is also notable that in heuristics with optimum EEPT for a network size of 30 nodes, NSR-NMA has a slightly better performance than the SR-NMA. It is an indication of small number of formed closed loops in these networks with SR-NMA.

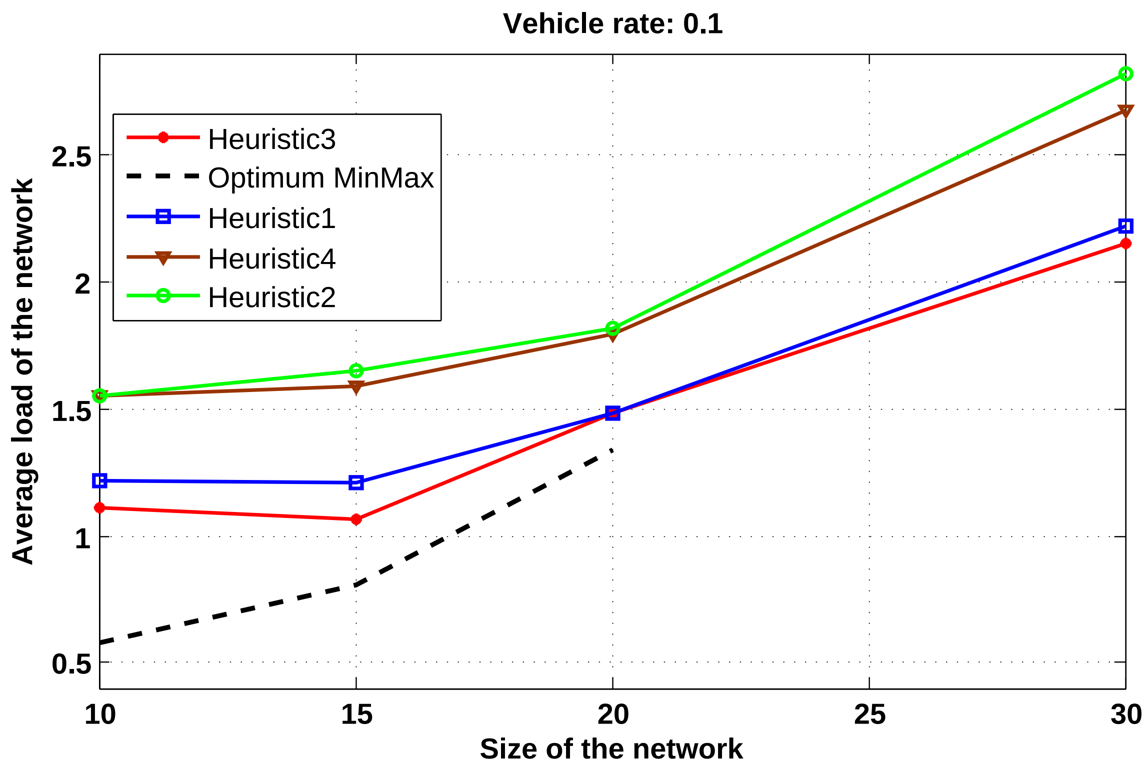


Figure 5.12: Average load comparison of the optimum solution with MinMax objective and heuristics for different network sizes and with a vehicle flow rate of 0.1 after 100 time slots of deployment. The average loads are normalized to the unit of energy (J).

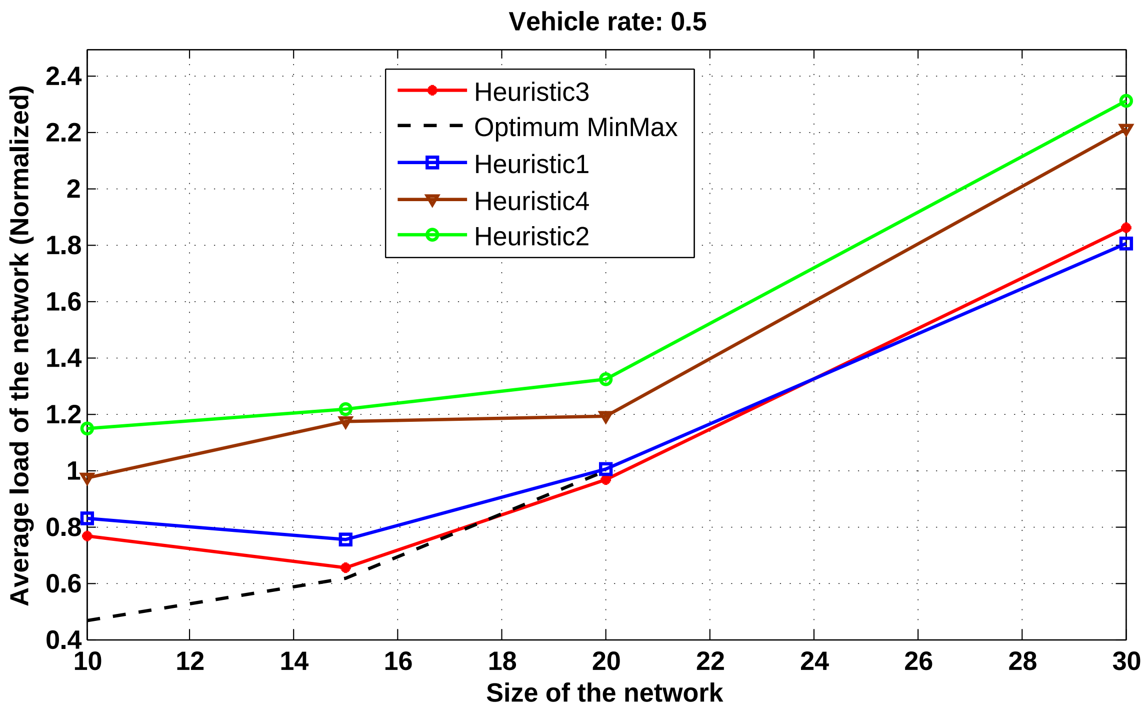


Figure 5.13: Average load comparison of the optimum solution with MinMax objective and heuristics for different network sizes and with vehicle flow rate of 0.5 after 100 time slots of deployment. The average loads are normalized to the unit of energy (J).

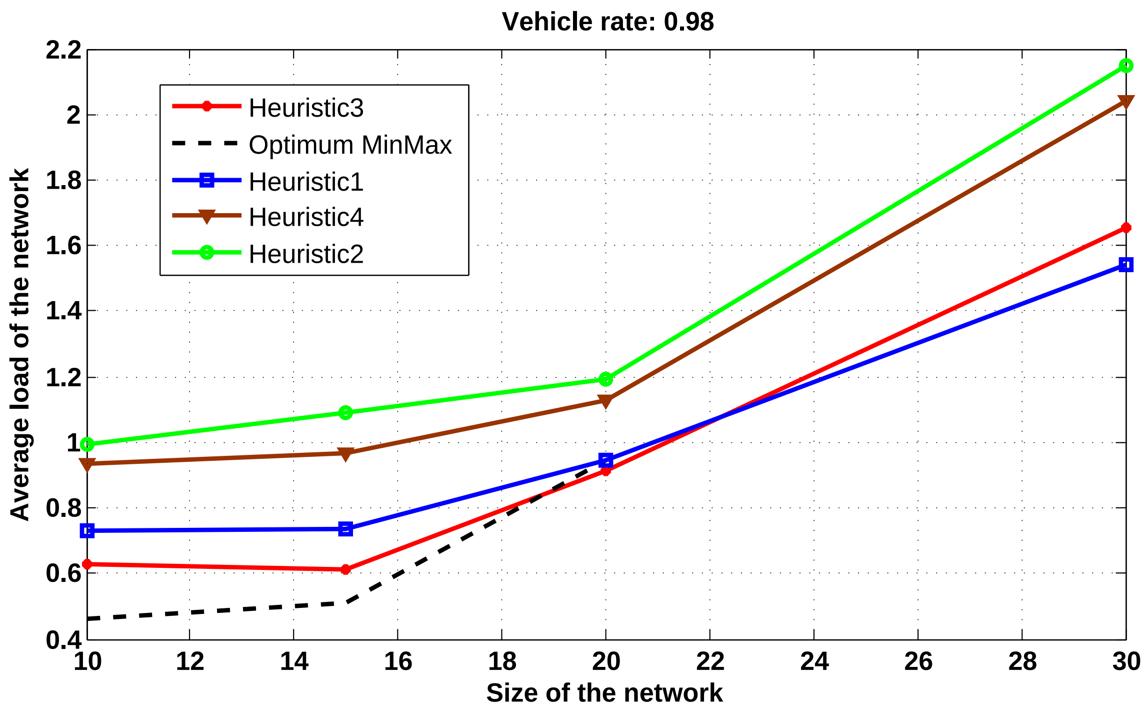


Figure 5.14: Average load comparison of the optimum solution with MinMax objective and heuristics for different network sizes and with vehicle flow rate of 0.98 after 100 time slots of deployment. The average loads are normalized to the unit of energy (J).

Figure 5.15 and 5.16 respectively shows average battery usage of a 10 and 15 nodes network with different traffic flow rates for the optimum MinMax solution and heuristics. Similar to maximum battery usage graph, the average load of the networks decreases rapidly for the traffic flow rate increases from 0.35 to 0.7. The SR-NMA and NSR-NMA has close performances in both network sizes while there is still a gap between the performance of heuristics using optimum EEPT and those using Dijkstra algorithm to drive EEPT.

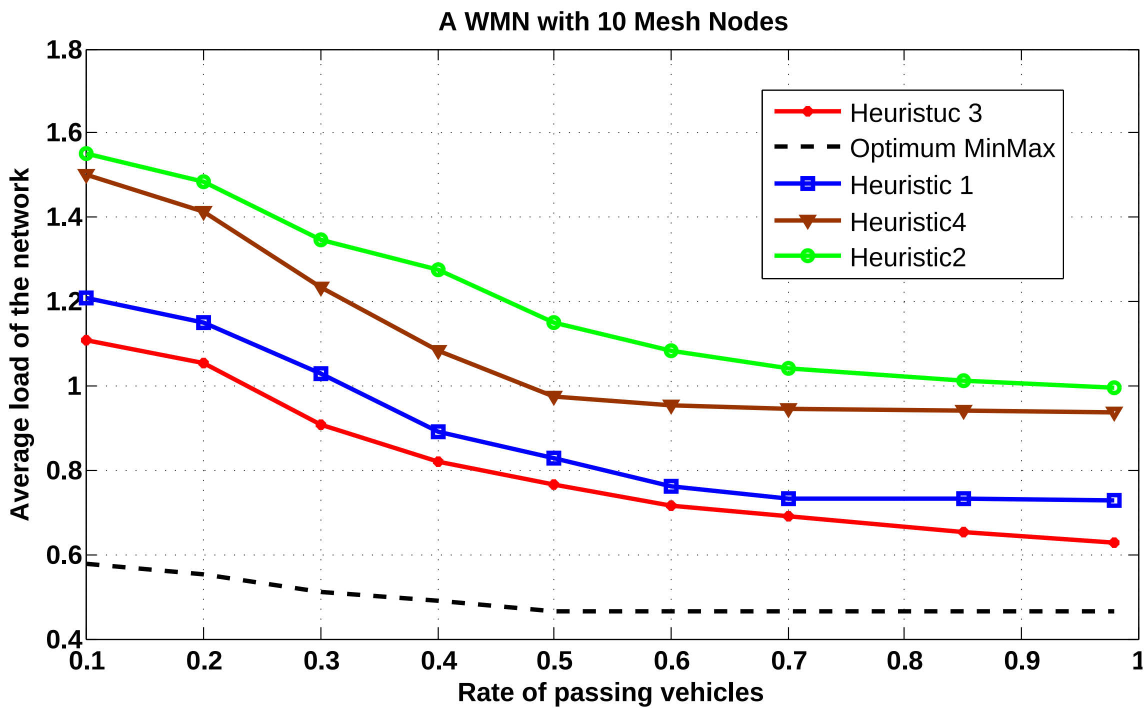


Figure 5.15: Average load comparison of the optimum solution with MinMax objective and heuristics for different vehicle flow rates in a network with the size of 10 after 100 time slots of deployment. The average loads are normalized to the unit of energy (J).

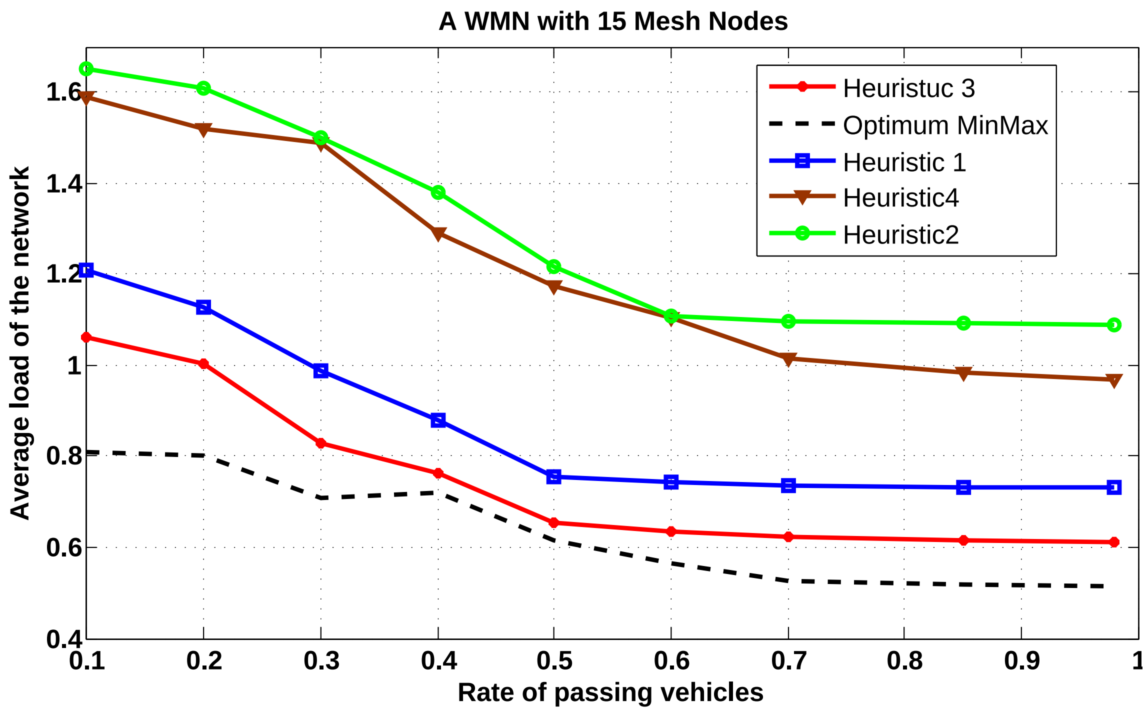


Figure 5.16: Average load comparison of the optimum solution with MinMax objective and heuristics for different vehicle flow rates in a network with the size of 15 after 100 time slots of deployment. The average loads are normalized to the unit of energy (J).

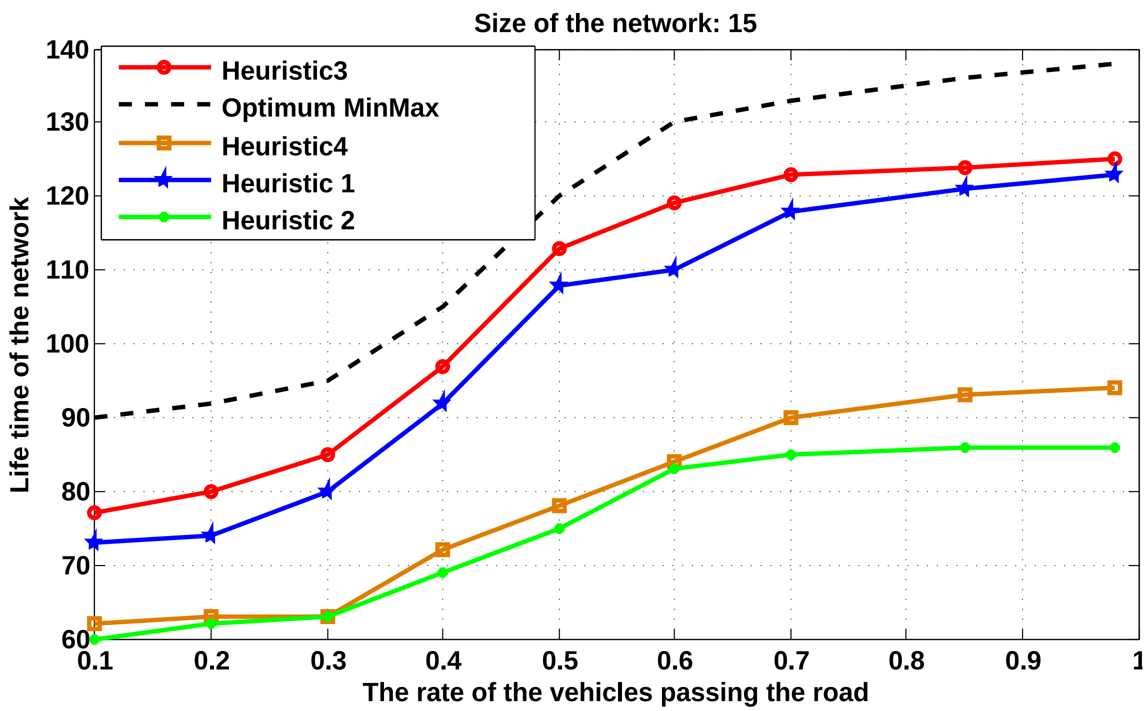


Figure 5.17: Life time of a WMN (In time slots) with 15 nodes and 0.7 units of battery capacity under different approaches and different vehicle flow rates. Each unit of battery capacity is the amount of available energy in the battery normalized to the unit of energy (J).

Figure 5.17 also shows the priority in performance, representing Life Time (LT) of a given network under different approaches. The nodes battery supply in this simulation are 0.7 unit. As can be seen, the variation of traffic flow rate has maximal influence on the network load burden when it is in the range of 0.3 to 0.6. The optimum solution has the best performance in terms of network lifetime compared to the heuristics. Heuristics 2 and 4 have the worse performance as they use the Dijkstra algorithm for driving EEPT rather than the optimum EEPT.

Chapter 6

Conclusion

In this thesis, improvements in mesh network energy efficiency are proposed by opportunistically ferrying delay tolerant traffic through co-existent vehicular networks. This is done by routing traffic through vehicular paths when this forwarding is consistent with trafffic flow quality-of-service constraints.

Two different approaches were considered to model the problem. The first one integrates the data units in flows and obtains an optimum routing bound under both Shortest Path (SP) and Maximum Lifetime (ML) objectives. This approach is called the flow based model. Simulation results for a proposed heuristic showed large improvements in network performance when this is used compared to conventional single-network multi-hop forwarding. In particular, we find that using the ML criterion leads to much improved network lifetimes compared to that using the SP criterion and that obtained without vehicular forwarding. It was also found that the proposed algorithm results in considerable improvements in fair load distribution.

The other approach, called the packet based model, considers the packets as the data units and formulates an optimization problem. This optimization achieves a bound in network lifetime by finding the best packet routes. The model accounts for the queuing effects in congested networks and hence incorporates packet queuing delays. It has also the ability to schedule packets using the vehicular traffic pattern. The vehicular traffic is modeled by a Poison random process and the simulation results are achieved for different vehicular flow rates. Simulation results for four different proposed heuristics show significant improvements in network lifetime.

The packet based model is a more advanced and complicated model compared to the flow based model. It is capable of incorporating more traffic routing detail and hence is more appropriate for future extensions. Some of the possible work in packet based modeling of the OVAF can be summarized as follows:

1. Exploring the effect of different data traffic models on the OVAF approach.
2. Considering different models for vehicular input traffic.
3. Comparing the proposed heuristics in terms of packet loss.
4. Obtaining the network throughput for each heuristic as a function of data traffic load.
5. Devising a detailed ARQ protocol for integration of WMNs and VANETs.
6. Conducting node resource provisioning for the OVAF approach.
7. Designing a security standard for the data ferrying.

Bibliography

[1] C. Ma, Z. Zhang, and Y. Yang, "Battery-Aware Router Scheduling in Wireless Mesh Networks," in *20th International Parallel and Distributed Processing Symposium, IPDPS*, April 2006.

[2] G. Badawy, A. Sayegh, and T. Todd, "Energy Provisioning in Solar-Powered Wireless Mesh Networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 8, pp. 3859–3871, Oct. 2010.

[3] R. Shah, S. Roy, S. Jain, and W. Brunette, "Data Mules: Modeling and Analysis of a Three-Tier Architecture for Sparse Sensor Networks," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 215–233, 2003.

[4] R. Sugihara and R. Gupta, "Optimal Speed Control of Mobile Node for Data Collection in Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 9, no. 1, pp. 127–139, Jan. 2010.

[5] M. Ye, X. Tang, and D. L. Lee, "Fair Delay Tolerant Mobile Data Ferrying," in *10th International Conference on Mobile Data Management: Systems, Services and Middleware*, May 2009, pp. 182–191.

[6] I. Akyildiz and X. Wang, "A Survey on Wireless Mesh Networks," *IEEE Communications Magazine*, vol. 43, no. 9, pp. S23–S30, Sept. 2005.

[7] S. Dey, H. Sarmah, S. Samantray, D. Divakar, and S. Pathak, "Energy Efficiency in Wireless Mesh Networks," in *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec. 2010, pp. 1–4.

[8] F. Aron, T. Olwal, A. Kurien, and M. Odhiambo, "Energy Efficient Topology Control Algorithm for Wireless Mesh Networks," in *International Wireless Communications and Mobile Computing Conference, IWCMC*, Aug. 2008, pp. 135–140.

[9] E. Bulut, S. Geyik, and B. Szymanski, "Efficient Routing in Delay Tolerant Networks with Correlated Node Mobility," in *IEEE 7th International Conference on Mobile Adhoc and Sensor Systems (MASS)*, Nov. 2010, pp. 79–88.

[10] A. Nayebi, H. Sarbazi-Azad, and G. Karlsson, "Routing, Data Gathering, and Neighbor Discovery in Delay-Tolerant Wireless Sensor Networks," in *IEEE International Symposium on Parallel Distributed Processing, IPDPS*, May 2009, pp. 1–6.

[11] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein, "Energy-Efficient Computing for Wildlife Tracking: Design Tradeoffs and Early Experiences with ZebraNet," in *Proceedings of the 10th International Conference on Architectural Support for Programming Languages and Operating Systems.* ACM, 2002, pp. 96–107.

[12] Y. hua Zhu, W. deng Wu, V. Leung, and L. huai Yang, “Energy-Efficient Tree-Based Message Ferrying Routing Schemes for Wireless Sensor Networks,” in *Third International Conference on Communications and Networking in China*, Aug. 2008, pp. 844–848.

[13] O. Tekdas, V. Isler, J. Lim, and A. Terzis, “Using Mobile Robots to Harvest Data from Sensor Fields,” *IEEE Wireless Communications*, vol. 16, no. 1, pp. 22 –28, February 2009.

[14] K.-C. Lan and Z. M. Wu, “On the Feasibility of Using Public Transport as Data Mules for Traffic Monitoring,” in *IEEE Intelligent Vehicles Symposium*, June 2008, pp. 979–984.

[15] A. Pentland, R. Fletcher, and A. Hasson, “DakNet: Rethinking Connectivity in Developing Nations,” *Journal of Computers*, vol. 37, no. 1, pp. 78–83, Jan. 2004.

[16] M. Dikaiakos, A. Florides, T. Nadeem, and L. Iftode, “Location-Aware Services over Vehicular Ad-Hoc Networks using Car-to-Car Communication,” *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1590 –1602, Oct. 2007.

[17] R. Bruno, M. Conti, and E. Gregori, “Mesh Networks: Commodity Multihop Ad-Hoc Networks,” *IEEE Communications Magazine*, vol. 43, no. 3, pp. 123–131, March 2005.

[18] D. Aguayo, J. Bicket, S. Biswas, D. De Couto, and R. Morris, “MIT Roofnet Implementation,” 2005. [Online]. Available: http://pdos.lcs.mit.edu/roofnet/design/

[19] D. Benyamina, A. Hafid, and M. Gendreau, "Wireless Mesh Networks Design; A Survey," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1 –12, 2011.

[20] D. Johnson, "A Note on Dijkstra's Shortest Path Algorithm," *Journal of the ACM (JACM)*, vol. 20, no. 3, pp. 385–388, 1973.

[21] D. Lee, "Proof of a Modified Dijkstra's Algorithm for Computing Shortest Bundle Delay in Networks with Deterministically Time-Varying Links," *IEEE Communications Letters*, vol. 10, no. 10, pp. 734–736, Oct. 2006.

[22] A. Farbod and T. Todd, "Resource Allocation and Outage Control for Solar-Powered WLAN Mesh Networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 8, pp. 960–970, Aug. 2007.

[23] H. Miao, Y. Jin, Q. Ge, and Y. He, "Expected Transmission Energy Routing Metric for Wireless Mesh Senor Network," in *6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, Sept. 2010, pp. 1–4.

[24] Al-Hazmi, de Meer, K. Hummel, H. Meyer, M. Meo, and D. Remondo, "Energy-Efficient Wireless Mesh Infrastructures," *IEEE Network,*, vol. 25, no. 2, pp. 32–38, March-April 2011.

[25] C. Ma, M. Ma, and Y. Yang, "A Battery Aware Scheme for Energy Efficient Coverage and Routing in Wireless MIMO Mesh Networks," in *IEEE Wireless Communications and Networking Conference*, March 2007, pp. 3603–3608.

[26] G. Anastasi, M. Conti, E. Monaldi, and A. Passarella, "An Adaptive Data-Transfer Protocol for Sensor Networks with Data Mules," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, June 2007, pp. 1–8.

[27] R. Sugihara and R. Gupta, "Optimizing Energy-Latency Trade-Off in Sensor Networks with Controlled Mobility," in *IEEE INFOCOM*, April 2009, pp. 2566–2570.

[28] S. Jain, R. Shah, W. Brunette, G. Borriello, and S. Roy, "Exploiting Mobility for Energy Efficient Data Collection in Wireless Sensor Networks," *Mobile Networks and Applications*, vol. 11, no. 3, pp. 327–339, 2006.

[29] D. Jea, A. Somasundara, and M. Srivastava, "Multiple Controlled Mobile Elements (Data MULES) for Data Collection in Sensor Networks," *Lecture Notes in Computer Science*, vol. 3560, pp. 244–257, 2005.

[30] R. Sugihara and R. Gupta, "Data Mule Scheduling in Sensor Networks: Scheduling under Location and Time Constraints," Citeseer, Tech. Rep., 2007.

[31] ——, "Improving the Data Delivery Latency in Sensor Networks with Controlled Mobility," *Lecture Notes in Computer Science*, vol. 5067, pp. 386–399, 2008.

[32] M. Ma and Y. Yang, "Sencar: An Energy Efficient Data Gathering Mechanism for Large Scale Multihop Sensor Networks," *Distributed Computing in Sensor Systems*, pp. 498–513, 2007.

[33] M. Zhao, Z. Chen, X. Deng, L. Zhang, A. Liu, and G. Huang, "MAEC: A

Movement-Assisted Energy Conserving Method in Event Driven Wireless Sensor Networks," *IEEE Region*, 2005.

[34] G. Xing, T. Wang, W. Jia, and M. Li, "Rendezvous Design Algorithms for Wireless Sensor Networks with a Mobile Base Station," in *Proceedings of the 9th ACM International Symposium on Mobile Ad-Hoc Networking and Computing.* ACM New York, NY, USA, 2008, pp. 231–240.

[35] S. Gandham, M. Dawande, R. Prakash, and S. Venkatesan, "Energy Efficient Schemes for Wireless Sensor Networks with Multiple Mobile Base Stations," in *Proceedings of IEEE Globecom*, vol. 1. Citeseer, 2003, pp. 377–381.

[36] A. Azad and A. Chockalingam, "Mobile Base Stations Placement and Energy Aware Routing in Wireless Sensor Networks," in *IEEE Wireless Communications and Networking Conference, WCNC*, vol. 1, April 2006, pp. 264–269.

[37] M. Chuah and P. Yang, "A Message Ferrying Scheme with Differentiated Services," in *MILCOM*, vol. 3, 2005, p. 1521.

[38] R. Viswanathan, J. Li, and M. C. Chuah, "Message Ferrying for Constrained Scenarios," in *6th IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks*, June 2005, pp. 487–489.

[39] J. Yang, Y. Chen, M. Ammar, and C. Lee, "Ferry Replacement Protocols in Sparse MANET Message Ferrying Systems," in *IEEE Wireless Communications and Networking Conference*, vol. 4, March 2005, pp. 2038–2044.

[40] Y. Shi and Y. Hou, "Approximation Algorithm for Base Station Placement in

Wireless Sensor Networks," in *4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2007, pp. 512–519.