ABSTRACT

Title of dissertation: SECURITY AND ENERGY EFFICIENCY IN RESOURCE-CONSTRAINED WIRELESS MULTI-HOP NETWORKS

Evripidis Paraskevas,
Doctor of Philosophy, 2016

Dissertation directed by: Professor John S. Baras
Department of Electrical & Computer Engineering,
Institute for Systems Research

In recent decades, there has been a huge improvement and interest from the research community in wireless multi-hop networks. Such networks have widespread applications in civil, commercial and military applications. Paradigms of this type of networks that are critical for many aspects of human lives are mobile ad-hoc networks, sensor networks, which are used for monitoring buildings and large agricultural areas, and vehicular networks with applications in traffic monitoring and regulation. Internet of Things (IoT) is also envisioned as a multi-hop network consisting of small interconnected devices, called "things", such as smart meters, smart traffic lights, thermostats etc.

Wireless multi-hop networks suffer from resource constraints, because all the devices have limited battery, computational power and memory. Battery level of these devices should be preserved in order to ensure reliability and communication across the network. In addition, these devices are not a priori designed to defend

against sophisticated adversaries, which may be deployed across the network in order to disrupt network operation. In addition, the distributed nature of this type of networks introduces another limitation to protocol performance in the presence of adversaries. Hence, the inherit nature of this type of networks poses severe limitations on designing and optimizing protocols and network operations. In this dissertation, we focus on proposing novel techniques for designing more resilient protocols to attackers and more energy efficient protocols.

In the first part of the dissertation, we investigate the scenario of multiple adversaries deployed across the network, which reduce significantly the network performance. We adopt a component-based and a cross-layer view of network protocols to make protocols secure and resilient to attacks and to utilize our techniques across existing network protocols. We use the notion of trust between network entities to propose lightweight defense mechanisms, which also satisfy performance requirements. Using cryptographic primitives in our network scenario can introduce significant computational overhead. In addition, behavioral aspects of entities are not captured by cryptographic primitives. Hence, trust metrics provide an efficient security metric in these scenarios, which can be utilized to introduce lightweight defense mechanisms applicable to deployed network protocols.

In the second part of the dissertation, we focus on energy efficiency considerations in this type of networks. Our motivation for this work is to extend network lifetime, but at the same time maintain critical performance requirements. We propose a distributed sleep management framework for heterogeneous machine-to-machine networks and two novel energy efficient metrics. This framework and the

routing metrics are integrated into existing routing protocols for machine-to-machine networks. We demonstrate the efficiency of our approach in terms of increasing network lifetime and maintaining packet delivery ratio. Furthermore, we propose a novel multi-metric energy efficient routing protocol for dynamic networks (i.e. mobile ad-hoc networks) and illustrate its performance in terms of network lifetime. Finally, we investigate the energy-aware sensor coverage problem and we propose a novel game theoretic approach to capture the tradeoff between sensor coverage efficiency and energy consumption.

SECURITY AND ENERGY EFFICIENCY IN RESOURCE-CONSTRAINED WIRELESS MULTI-HOP NETWORKS

by

Evripidis Paraskevas

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2016

Advisory Committee:
Professor John S. Baras, Chair/Advisor
Professor Gang Qu
Professor Charalampos (Babis) Papamanthou
Professor Tudor Dumitras
Professor Michael C. Fu, Dean's Representative

Η παιδεία, καθάπερ ευδαίμων χώρα, πάντα τ' αγαθά φέρει.

(Education, just like a fertile land, brings all good)

Socrates, 469-399 BC, Philosopher

Dedication

To my mother Eleni, my father Christos and my brother Andreas

for their invaluable love, support and inspiration

throughout my PhD journey.

Acknowledgments

I am grateful to my academic advisor, Prof. John S. Baras, for his continuous support, generosity and encouragement during my graduate studies at the University of Maryland and for giving me an opportunity to work on challenging and extremely interesting research topics. His energy, enthusiasm, persistence, deep mathematical insight and expertise in a broad range of topics have always been a constant source of motivation for me. The fact that he fosters a research environment free of constraints, where a student is encouraged to explore different research areas, allowed me to work on various problems that gave me a solid mathematical background, a lot of diversity and research experience. I would also like to thank Prof. Baras for giving me the opportunity to study abroad for a semester at KTH in Sweden, which was one of the most fruitful periods of my PhD life.

I am also grateful to Professors Gang Qu, Charalampos (Babis) Papamanthou, Tudor Dumitras and Michael C. Fu, for serving on my committee and for providing me useful feedback for the improvement and completion of this work. Prof. Qu provided to me insightful comments for most part of my dissertation, as well as helpful advices for my future endeavors. I am especially grateful to Prof. Papamanthou for his helpful advice and encouragement in several occasions of my PhD journey. Prof. Papamanthou introduced me to cloud computing security and I had the pleasure to collaborate with him in a research paper related to accountable cloud storage. His deep knowledge in security related problems motivated me to learn a lot on his research area. In addition, Prof. Dumitras introduced me to data-driven security

topic and I had the opportunity to work with him on a research project, where he provided us with invaluable insight and knowledge on the connection of machine learning and security. Finally, Prof. Fu enhanced my knowledge in stochastic modeling and queueing theory with his class.

I would also like to express my deepest gratitude to my mentors in Applied Communication Research Labs and Mitsubishi Electric Research Laboratories. Without their support and mentorship big parts of the dissertation would not have been completed. I would like to thank Dr. Kyriakos Manousakis, Dr. Mariusz Fecko and Dr. Ken Young from Applied Communication Sciences and Dr. Jianlin Guo and Dr. Philip Orlik from Mitsubishi Electric Research Laboratories. Dr. Manousakis' continuous support, deep knowledge in networking and enthusiasm drove me during my PhD degree. Finally, I would also like to specially thank Dr. Guo for his invaluable support and feedback throughout our collaboration.

At this point, I would like to thank the fellow PhD students of my research group. They made my graduate studies both intellectually stimulating and incredibly enjoyable. I would like to thank my office-mates over the years Anup Menon, Ladan Rabieekenari, Christoforos Somarakis and Yuchen Zhou, who made the office an enjoyable place to work. Special thanks to my fellow PhD comrades Lida Apergi, Dimitrios Dimitriadis, Peixin Gao, Shalabh Jain, Iakovos Katsipis, Xiangyang Liu, Wentao Luan, Dipankar Maity, Ren Mao, Chrysoula Papagianni, Tuan Ta, Eirini-Eleni Tsiropoulou and Xiangnan Weng. I am also greatly thankful to Mrs. Kim Edwards, for her generous and timely assistance with the administrative aspects of my work. Her care and patience have been a great resource for me throughout my

stay at UMD.

My time in graduate school has been especially fun thanks to my friends at UMD Adi Hajj Ahmad, Dimitrios Antonopoulos, Ioannis Demertzis, Berk Gurakan, Maya Kabkab, Nikolaos Kofinas, Vassilios Lekakis, Iason Papanikolaou, Moschoula Pternea, Theodoros Rekatsinas, Jason Thomopoulos, Zois Tsinas, Vassilios Tsirigotakis, Kleoniki Vlachou, Antonios Xenakis, Konstantinos Xirogiannopoulos, Konstantinos Zampogiannis and Vassilios Zikas. I would also like to thank my friends in the USA Arianna, Chrysoula, Ellie, Evangelos, Maria, Mike, Nasos, Theodora, Sofia and Spyridon and my friends in Greece Damianos, Elena, Fotis, Ioanna, Maria, Nikolaos, Panagiotis, Theodoros, Stefania and Vassilios. I also want to thank someone very close to my heart, Mina, for her patience and love through the toughest part of this journey.

My research was supported by the Air Force Research Laboratory (AFRL) and the Defense Advanced Research Projects Agency (DARPA) under the Wireless Network Defense (WND) program, by the Air Force Office of Scientific Research (AFOSR) MURI, by the National Science Foundation (NSF) and by the National Institute of Standards and Technology (NIST).

Finally, I would like to express my gratitude and love to my parents Eleni and Christos. Their unconditional love and support has been the source of strength for all my endeavors.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AODV	Ad-hoc On-demand Distance Vector routing protocol
AP	Active Period
BATMAN	Better Approach To Mobile Adhoc Networking routing protocol
BPN	Battery Powered nodes
B-RPL	Battery IPv6 Routing Protocol for Low-Power and Lossy Networks
CORE	Common Open Research Emulator
CRL	Collaborative Reinforcement Learning
DoS	Denial of Service
DSR	Dynamic Source Routing
DSDV	Destination Sequenced Distance Vector routing protocol
ETX	Excepted Transmission Times
EW	Energy Waste metric
MAC	Medium Access Control (Layer 2 - OSI network model)
MANET	Mobile Ad-hoc Network
MPN	Mains Powered Nodes
MSN	Mobile Sensor Network
M2M	Machine-to-machine Network
ND	Neighborhood Discovery
NE	Nash Equilibrium
NS2	Network Simulator v2
NS3	Network Simulator v3
NUM	Network Utility Maximization
OLSR	Optimized Link State Routing
PDR	Packet Delivery Ratio
QoS	Quality of Service
RC	Relay Cost metric
RC	Reliability Cost
RE	Residual Energy metric
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
RREQ	Route Request packet
SB	Selfish Behavior Attack
SN	Selfish Node Attack
SR	Soft Reliability estimates
TC	Topology Control packets

CHAPTER 1

Introduction

Wireless multi-hop networks have attracted a lot of interest from the research community, because of their wide range of applications from military networks to emergency response networks, sensor networks and healthcare systems. There have been several efforts to make these type of networks more efficient and resilient to attacks and misconfigurations. Some paradigms of wireless multi-hop networks [1] are: *Mobile Ad-hoc Networks (MANET)*, which are self-configured and dynamic networks with changing topology, *Sensor Networks*, which consist of low-power nodes and are used for monitoring large buildings and agricultural areas and *Vehicular Ad-hoc Networks (VANET)*, which consist of vehicles moving rapidly in and out of the network area. Another emerging paradigm of a wireless multi-hop network is *Machine-to-Machine (M2M)* networks, which consists mainly from sensor nodes and its operation is very similar to a sensor network. Most of these examples of networks are parts of the *Internet of Things (IoT)*, which is also considered as a wireless multi-hop network with small connected devices ("things").

This type of networks suffer from resource constraints, such as computation

power, battery power and memory limits. These resource limitations should be taken into account when designing new protocols or proposing novel optimization schemes. These networks are also vulnerable to different type of attacks, which attempt to deplete the network resources and degrade the network performance. In addition, energy preservation is of utmost importance for this type of networks. In this dissertation, we propose novel schemes and methodologies from a practical and a theoretical point of view in order to handle with problems under these two different optimization criteria, security and energy efficiency.

1.1 Security Considerations in Wireless Multi-hop Networks

Over the past few years, several critical security issues have emerged, both in commercial and civil infrastructure domain. Wireless multi-hop networks are being used to deliver sensitive and critical data and are prone to attackers, because the network nodes are designed to defend against different attacks. Hence, attackers are capable of taking control of the network and releasing different type of attacks from denial-of-service (DoS) attacks to data integrity attacks, by dropping, modifying, injecting or delaying various messages. These attacks lead to significant performance degradation, because of modifications imposed by the adversaries in the benign network operation, or leakage of sensitive private data.

In the first part of this dissertation, we propose lightweight, in terms of computation overhead, modifications on routing protocols and other network operations, in particular cross-layer optimization, in order to mitigate the effect of different type

of network attacks. We investigate the attacker landscape and the notion of trust, which will be incorporated as a security and resiliency metric in this dissertation.

1.1.1 Attacker Landscape

Systems with distributed architecture, such as wireless multi-hop networks, are prone to attackers. There are several types of attacks that are deployed in wireless multi-hop networks with different objectives. The adversaries deploy their attacks in several parts of the network and aim at degrading the *availability* and the *integrity* of the network. The adversaries are intelligent and it is in general more difficult to detect their misbehavior in comparison with traditional systems.

The different attackers can be classified as external or internal. External attackers are not authenticated and do not possess the corresponding credentials (e.g., digital certificates) to be considered part of the network. Misbehavior of external adversaries can be identified via traditional cryptographic techniques, such as public-key based techniques (confidentiality) or message authentication codes (message integrity) and can be excluded from the network.

On the other hand, internal attackers possess the required credentials to be part of the network, but they are not following the protocol. These adversaries release more sophisticated attacks, which are more difficult to be detected. They manipulate control and data packets in order to cause disruption and degradation of network performance. In this dissertation, we focus on mitigating the effect of these internal network adversaries, or so called Byzantine adversaries, which behave

as normal entities, but they modify the protocol's operation. Some examples of these internal Byzantine adversaries, which are presented in [2] and [3], are black-hole/grayhole attacks, wormhole and selective forwarding attacks.

1.1.2 Notion and Benefits of Trust

Wireless multi-hop networks consist of devices with low computational power, limited energy and memory. Hence, traditional cryptographic techniques are not suitable for these paradigms of networks, because they create heavy computational overhead. In addition, traditional cryptographic techniques cannot be used to detect sophisticated internal adversaries, who possess the credentials to be part of the network. To address the misbehavior of compromised nodes, we use the notion of trust between the network entities. Trust represents a quantified relationship between the network entities, based on some observations of neighbor entities' behavior or previous interactions. Trust values constitute a lightweight security metric, which can be exploited to increase security and resiliency in the network. In addition, trust values can also represent misconfigurations of failures in the network.

Several definitions for trust have been proposed in the literature. In [4] trust denotes the entity's opinion about the trustworthiness of a digital certificate is defined as a continuous value in $[0, 1]$. A general theory of trust in networks that consist of devices and humans is introduced in [5]. The authors distinguish the trust in behavioral and computational and attempt to establish new trust relations, in the case that humans participate actively in the network. One other definition of

trust introduced in [6] incorporates the notion of confidence for our estimated trust value of a network entity. Confidence indicates the accuracy of the trust estimation for a particular entity.

Estimation of the trust values in different update periods is a crucial operation and it can be performed using different approaches. One way is to deploy various detectors as proposed in [7] to observe misbehavior of network entities. Other approaches for trust evaluation have been proposed in [8] and [9]. Trust values should be periodically propagated in the network, in order to be used by the distant network nodes. Theodorakopoulos and Baras in [6] proposed a semiring based framework, which provides a robust mechanism for distribution of trust.

Finally, an advantage of using trust estimates is that they can be easily integrated into system components to make them more resilient and secure compared to traditional cryptographic methods. Hence, in our work, we utilize trust estimates to enhance security in different protocol components.

1.1.3 Component-based Protocol Design

Component-based protocol design is a systems engineering approach [10] for modeling and designing network protocols for wireless multi-hop networks. Network protocols can be viewed as *systems of systems* and can be separated into components based on their different functionalities. The distinct protocol components interact with each other and exchange useful information related to the protocol procedure. The methodology was first proposed by Baras et. al in [11] and [12]. A cross

layer analysis of MAC and routing protocols, based on the idea of component-based modeling, was also introduced in [13]. The method provides a systematic approach that can be used in the design, performance analysis and optimization of network protocols. The main objective of the approach is the separation of concerns between the different components, which overlap as little as possible in functionality.

The component-based approach provides two major contributions in protocol design and modeling. First, it allows *modularity* in protocol design. Routing protocols are usually implemented as large monolithic software, which are very difficult to adapt to varying environmental conditions. By using the component-based approach, we abstract the functionalities of the protocol into fundamental building blocks and we can easily design and model each of these blocks to adapt to the environmental changes. Furthermore, our approach allows *reusability* of existing components across current and future protocols of the same class. The objective is to create a library of components that can be easily plug into each protocol and configure its functionality according to the environmental conditions to increase the performance. The novel components should be designed in a way that allow reusability with minor modifications. Minor modifications are needed based on the implementation details of each network protocol.

In the second chapter of the dissertation, we take advantage of the component-based approach to design novel modified protocol components, which mitigate the effect of different types of attacks that cause significant network performance degradation. We observed that different type of attacks affect significantly the performance of specific protocol components. Hence, this approach provides localization

of attacks to protocol components. As part of this dissertation, we utilize updated trust estimates to design a set of lightweight mitigation techniques, which are incorporated into protocol components and lead to high performance recovery.

1.2 Energy Efficiency Considerations in Wireless Multi-hop Networks

Wireless multi-hop networks suffer from energy limits, because most of the nodes in this type of networks are battery-powered and are not rechargeable. Hence, a crucial limitation that we should take into account when designing new protocols is the battery capabilities of the network.

In the second part of this dissertation, we propose novel energy efficient routing schemes for different types of wireless multi-hop networks. We investigate the different sources of energy depletion in a network and propose efficient solutions that incorporate sleep management and the introduction of novel routing metrics. Sleep management enables nodes to be inactive in specific time periods that they do not need to forward data packets. Routing metrics modify the route selection in a multi-hop network. Our proposed solutions can be easily integrated to standard routing protocols.

Finally, we deal with the energy-aware coverage problem in a sensor network. Coverage is a crucial operation of wireless sensor networks, because sensor nodes should configure appropriately their coverage area in order to optimize data collection. Data packets are then forwarded to the sink nodes of the sensor multi-hop network. However, battery limitations of the deployed sensor nodes should be taken

into account in the coverage problem. Therefore, we propose novel game-theoretic coverage approach to capture the tradeoff between coverage and energy consumption.

1.2.1 Energy Efficient Routing

Energy efficiency considerations should be taken into account when designing and implementing new routing schemes for wireless multi-hop networks. There are several causes of energy depletion during the routing procedure, such as transmitting, receiving data packets, idle listening and overhearing. The proposed routing schemes, which we investigate as part of this dissertation, aim at reducing energy depletion through idle listening and overhearing and selecting more energy efficient paths by introducing appropriate routing metrics. Hence, we focus on sleep management schemes and new energy efficient schemes that prolong network lifetime. We define as network lifetime, the time until one node of the network is completely depleted from energy. There are several other definitions for network lifetime that we have not taken into account as part of this dissertation, such as the time until a specific number of nodes is depleted from energy or the time until network partitioning, because several nodes are depleted.

Sleep management schemes have been proposed in [14], [15], [16] and [17]. These schemes enable scheduling of the nodes in order to be active in specific periods of time and hence save energy and extend network lifetime. Energy efficient routing metrics have been proposed in [18], [19], [20], [21] and [22]. We will describe these

metrics in more details in the following chapters of this dissertation. These metrics modify the routing procedure and enable more energy-aware selection of forwarding paths.

1.2.2 Energy-Aware Sensor Coverage

Coverage is a crucial aspect of wireless sensor networks operation and is used for efficiently monitoring large areas and collecting measurements. The sensor nodes, which are used for collecting data, are battery-limited and consume significant energy in order to increase the efficiency of coverage. Therefore, sensor nodes aim at maximizing a common utility function through efficient coverage, but at the same time they should reduce the energy consumption.

Different approaches have been proposed to optimize sensor coverage problems As part of this dissertation, we mainly focus on game-theoretic approaches introduced in [23], [24] and [25]. Moreover, there is a lot of interest to investigate energy limitations for the sensor coverage problem. Some solutions that capture the trade-off between efficient coverage and energy efficiency, which is the objective of Chapter 6 of this dissertation, are presented in [26] and [27].

1.3 Contributions of the dissertation

In this dissertation, we introduce practical and theoretical schemes for security and energy efficiency in resource-constrained wireless multi-hop networks. In the beginning of the dissertation, we introduce a novel approach for secure and

resilient protocol design inspired by the component-based protocol design methodology. This methodology allows us to separate network protocols into components and introduces the notion of component-based security and resiliency by hardening the individual components rather than the protocol itself. This approach is proved to be more efficient than the related work, where protocols are created from the beginning or modified in order to mitigate the effect of specific network attacks. We have introduced a framework for mitigating the effect of different network attacks. The framework contains a set of lightweight mitigation techniques that are associated with protocol components. These mitigation techniques utilize trust estimates to change the functionality of each component and are highly reusable across protocols of the same class. We introduce and develop three mitigation techniques for our framework, which we have integrated into standard proactive routing protocols, and we illustrate through experimentation that they can achieve fast and high performance recovery under different deployed attacks and environmental conditions.

In the following chapter, we introduce the notion of security into cross-layer optimization problems (NUM problem) for wireless multi-hop networks. Our optimization framework utilizes trust estimates as the security metric, in order to prevent allocating data rate through untrusted paths consisting of compromised intermediate nodes. Trust estimates are introduced as soft constraints in the optimization problem and affect the optimization process. We have also introduced delay constraints in the optimization problem, in order to address QoS requirements. Our performance evaluation indicates that source rate is mainly allocated in the trusted paths, but in the scenario that there is higher traffic demand the untrusted paths

should be utilized for data forwarding.

In the second part of the dissertation, we propose novel distributed sleep management techniques for heterogeneous wireless M2M networks to efficiently manage battery lifetime. Our network is heterogeneous in the sense that it consists of battery-powered nodes (BPNs) and a low percent of mains-powered nodes (MPNs). BPNs schedule variable length sleep intervals that start with an active period (AP), followed by a sleep period (SP). BPNs decide the length of sleep intervals in a distributed manner based on their local data traffic, which consists of buffered packets, self-generated packets, and prediction of incoming relay data traffic. For this purpose, we introduce a model to estimate the number of incoming relay packets, in which we take into account buffer overflow, channel uncertainty and probability of the BPN being active. In addition, we have proposed an AP extension scheme, which gives BPNs the flexibility to transmit or receive more data packets. We also introduce two novel battery energy aware metrics: *Battery Node Energy Waste* (EW) and *Battery Node Relay Cost* (RC). These two metrics are node related metrics and take into account sleep management and different causes of energy depletion. They aim to discover the best route that satisfies some battery related properties. We have integrated our proposed scheme and routing metrics into a standard protocol for M2M networks to show their applicability. Finally, we have illustrated through extensive performance evaluation that our scheme achieves better performance than the related work.

Furthermore, we propose a novel multiple metric routing scheme to be applied in dynamic wireless multi-hop networks, such as mobile ad-hoc networks (MANET),

and integrate it into a standard routing protocol for MANETs, called OLSR, to examine its effectiveness. We use a combination of routing metrics from multiple layers and network topology to create a unified routing metric, which represents the cost of a network node. These metrics are MAC queue utilization, residual energy and node degree. For our experimentation, we create the *Modified OLSR* protocol, which utilizes the novel multi-metric energy efficient scheme, and we examine its energy behavior and its performance (in terms of network lifetime and PDR) compared to the standard OLSR protocol. Finally, in the last part of the dissertation, we address the energy-aware sensor coverage problem using a game-theoretic control approach. We design a utility function that captures the trade-off between coverage efficiency and energy consumption and we propose a distributed learning rule that enables bit-valued communication between the different nodes (agents). In addition, we analyze and prove the convergence of our approach to a Nash Equilibrium (NE).

1.4 Organization of the dissertation

This dissertation is organized into five chapters. The first two chapters address security considerations in wireless multi-hop networks. We propose efficient and lightweight solutions, which utilize trust estimates, to modify appropriately network protocols and network optimization procedures and make them more resilient to network layer attacks. The last three chapters address energy preservation considerations in this type of networks. We introduce novel techniques for energy efficient routing and energy-aware sensor coverage.

Chapter 2 contains our work on component-based reusable adaptive mitigation techniques for network layer attacks. Inspired by the component-based protocol design, we have proposed a set of mitigation techniques that use trust estimates and could be reused across different protocols. We illustrate the efficiency of the techniques in terms of performance recovery with experimental evaluation. This work is presented in [28] and [29].

In Chapter 3, we describe our work on trust-aware network utility maximization, which appeared in [30]. We enhance the cross-layer optimization technique with trust metrics in order to mitigate the effect of malicious nodes across the different paths. We illustrate the effectiveness of our approach via simulations.

In Chapter 4, we describe our work on distributed sleep management for heterogeneous wireless machine-to-machine network, which appeared in [31]. We have integrated the proposed solutions into a standard sensor network routing protocol (i.e. RPL [32]) and conducted network simulations to show the efficiency of our proposed scheme.

Chapter 5 presents our work on multi-metric energy efficient routing for dynamic ad-hoc networks. We have integrated the novel introduced routing metrics into a standard ad-hoc routing protocol (i.e. OLSR [33]) and we have conducted extensive simulations under static and various mobility scenarios to indicate the effectiveness of our approach. This work appeared in [34].

In Chapter 6, we present our work on distributed energy-aware mobile sensor coverage, which appeared in [35]. We formulate energy-aware sensor coverage as a game theoretic control problem and we propose an algorithm that is proved to reach

a Nash Equilibrium for the sensor coverage problem.

Finally, we conclude this dissertation and we present several directions for future work in Chapter 7.

CHAPTER 2

Component-based Mitigation of Attacks in Mobile Ad-hoc Networks

2.1 Overview

Mobile ad-hoc networks are exposed to control layer attacks, which decrease significantly their performance. Most of these attacks have tremendous impact on the packet delivery ratio and the end-to-end latency, which are crucial performance metrics for a wide range of applications. The motivation of our work is to design efficient and lightweight schemes to mitigate the effect of the attacks and achieve reasonable performance. The mitigation schemes should be easily applicable to existing MAC and routing protocols in order to be used in a wide variety of networks. The innovative idea in order to make the adaptive mitigation applicable to many existing and future protocols is to identify some reusable protocol components, i.e. fundamental building blocks of communication protocols. These reusable protocol components are defined by the separate functionalities of the protocols and they need to be general enough to include the instantiation of various protocols.

The different types of attacks do not affect in the same way the different

components of the protocol, i.e. an attacker that selectively drops packets in the data plane have a significant effect in the data forwarding component. Our objective is to detect and modify the "weak" components with a set of mitigation techniques. Mitigation techniques use reliability estimates as input, which are combinations of trust and confidence (for the trust estimate) values. These estimates indicate which nodes are compromised and what type of attacks have been deployed in the network. The next incentive for the component-based approach is that we are capable of embedding the mitigation techniques into specific protocol components, which are activated appropriately when we detect a specific attack to adapt to the environmental conditions. If we take it one step further we can claim that the mitigated components can be used as fundamental building blocks in many communication protocols without significant changes. There would only be some protocol specific changes that should be done and it is a very small percentage of the main protocol's component. This increases maximum reusability across multiple protocols and contributes to the design of new and more robust communication protocols.

2.1.1 Related Work

In this subsection, we investigate the state of the art work on proposed mitigation techniques for wireless network protocols, as well as, the related work on component-based protocol design methodologies.

2.1.1.1 Mitigation techniques for network protocols

There are several mitigation techniques proposed for wireless network protocol to resist against various types of attacks. In [3] the authors propose a new secure on-demand routing protocol for ad-hoc networks, called ODSBR, which is proved to be resilient against several Byzantine attacks. It uses an adaptive probing technique to identify the malicious links and then it modifies the route discovery process based on a metric that captures this adversarial behavior. A new secure routing protocol with quality of service guarantees was proposed in [36]. This new protocol incorporates a secure route discovery process and the addition of some trustworthiness-based QoS routing metrics, which combine trust and QoS guarantees (packet delay and link quality). Secure route discovery schemes that modify existing protocols have been proposed in [37] for AODV [38], which is a reactive routing protocol, and in [39] for OLSR routing protocol, which is a proactive routing protocol. In addition, in [40] the authors introduce novel cross-layer routing metrics to mitigate the effect of selfish behavior (packet dropping attack). They illustrate the efficiency of the proposed routing metrics by extensive performance evaluation using the ODBSR routing protocol. Secure message transmission (SMT) was introduced by Papadimitratos and Hass in [41] to secure the data transmission process, by detecting and avoiding non-operational or compromised routes. Finally, a new scheme for topology control that creates k-connected networks was introduced in [42]. This scheme can adapt the network topology in the case that our network is completely partitioned due to a set of compromised nodes.

Most of the prior art, described above, introduced new secure protocols or protocol-specific mitigation techniques to defend against common network layer attacks. However, the applicability of these techniques across a variety of existing and future network-layer protocols is limited because they are not designed in a reusable manner. A general methodology is therefore needed for modifications of the behavior of different existing protocols based on the current network state and detected threats.

2.1.1.2 Component-based design of network protocols

Component-based design of wireless routing protocols was first proposed by Baras and He in [12]. In this work, the authors proposed a general decomposition of reactive routing protocols, such as AODV [38] and DSR [43] into four main components, based on the different operations of this type of protocols: path discovery, route maintenance, topology database maintenance and data packet forwarding component. They also defined component related performance metrics and examined the effect of them in the overall performance metrics of this type of protocols. In addition, they proposed a methodology to detect and replace the weak component, i.e. the component that leads to significant performance degradation.

Baras et. al introduced a decomposition of proactive routing protocols in [11] and used OLSR [33] as a case study. The authors proposed a decomposition of this type of routing protocols into components and analyzed the operation of three of the fundamental building blocks. In addition, they focused on the Neighborhood

Discovery Component (NDC) and provided a methodology for design and modification of this component that leads to a routing protocol with reliable performance. The authors conducted performance analysis among the modified version of OLSR with the proposed NDC component and the standard OLSR protocol.

A software framework called CONFab for component based optimization of wireless sensor networks protocol stacks is proposed in [44]. The authors treated the protocol stacks as a collection of interdependent configurable components. Based on the scenario and the desired performance metrics the framework suggested suitable protocol stacks and selection of parameters. It also took advantage of a deployment feedback mechanism that uses knowledge of previous deployments of protocol stacks (combined routing and MAC layer protocols) in order to select the protocol stack to meet the performance requirements.

Furthermore, a component-based architecture for power-efficient MAC protocol development in wireless sensor networks, named MAC Layer Architecture (MLA), is presented in [45]. The authors defined and implemented a set of fundamental components for MAC layer protocols in wireless sensor networks. These components are optimized and reusable across different protocols as they implement a set of common features shared by existing MAC protocols. The authors examined the flexibility of the architecture by implementing five well-known MAC layer protocols using the defined reusable components. Performance evaluation showed that these implementations have comparative performance with the monolithic implementations of the same protocols. Finally, a declarative perspective on adaptable extensible MANET protocols is presented in [46]. The authors proposed the con-

struction of composite protocols using two mechanisms: policy-driven hybrid protocols and component-based routing. In component-based routing they presented some initial thoughts of specifying declaratively common functionalities of routing protocols as components that will be used across multiple protocols and will be activated upon occurrence of certain events in the network.

2.1.2 Summary of Contributions

In this chapter, we present a novel methodology for secure and resilient protocol design using the component-based approach, which enables decomposition of protocols of the same class into different components. The decomposition of the network protocol allows us to analyze the individual components within protocols and develop mitigation techniques associated with specific components rather than with the entire protocol. Hence, we are capable of swapping components (without built-in mitigation techniques) with the ones that have mitigation techniques built into them, which we call mitigated components, to defend against different network attacks. In addition, maximum *reusability* of these techniques can be achieved among different protocols of the same class by creating a library of well-known components of network-layer protocols. Thus, our objective is to develop a framework that incorporates a set of reusable lightweight mitigation techniques, which are applied to existing and future protocols and not to design completely new secure protocols. Finally, our methodology enables localization of different attacks to different components, because each attack aims at degrading the performance of a

specific protocol component. After localizing the weakest component, we activate a mitigation technique associated with this component in order to recover network performance.

The main contributions of this chapter are:

- The design of component-specific mitigation techniques that harden the selected network layer (e.g., routing) protocols against a wide range of attacks and are reusable across protocols of the same class. These novel mitigation techniques are lightweight, because they utilize trust estimates, and modify the functionality of particular protocol component.

- Experimental validation of the effectiveness of the applied mitigation techniques in terms of performance recovery.

We introduce three network layer mitigation techniques, which are presented in [28] and [29]: *trust-aware link weight adjustment (TALWA)*, *redundant packet forwarding (RPF)* and *neighbor discovery gatekeeper (NDG)*. These mitigation techniques are incorporated into specific components of the network-layer protocol and modify their behavior dynamically in response to severe attacks against the control plane. The techniques are also tested to be highly reusable across different protocols since the percentage of protocol specific code is very low. Our performance evaluation shows: (i) *fast recovery* against various attacks in the case where the most efficient mitigation technique is selected and (ii) *high recovery* of throughput.

2.1.3 Chapter Organization

The chapter is organized as follows. In Section 2.2 we describe the soft reliability (SR) estimates, which are utilized for the proposed techniques. We describe our proposed proactive routing protocol decomposition in Section 2.3. We used proactive routing protocol as the use case for our proposed system. In Section 2.4 we present the adversary models used in our analysis and experiments. We describe the newly designed component-based mitigation techniques in Section 2.5. We examine the performance of the mitigation techniques in Section 2.6. Finally, we conclude our work in Section 2.7.

2.2 Soft Reliability (SR) estimates

The proposed mitigation techniques utilize trust estimates to modify the functionality and increase the resiliency of protocol components. By using trust estimates we avoid the heavy cryptographic operations and we are also capable of detecting misbehavior of authenticated malicious entities in the network. In addition, by using the trust estimates we can capture additional misconfigurations or failures in the network. Finally, trust estimates compared to cryptographic techniques can be easily integrated in modular system design and used as an input to protocol components.

In this chapter, we adopt the definition for trust introduced by Theodorakopoulos and Baras in [6], where we have multiple tuples of *trust* and *confidence* values (t, c) for each network node. Trust values indicate the trustworthiness of an

entity derived by the past observations and interactions of this entity with each of its neighbors nodes. Confidence values indicate the accuracy of the trust value that we have received for a particular entity. Both trust and confidence take numerical values in the range $[0, 1]$ and are updated and propagated to the network periodically. Higher trust value indicates that the corresponding entity is benign and higher confidence value shows that the entity has passed a large number of tests or it had a lot of interactions with its neighbor nodes in order to get an accurate trust value estimate.

In this work, we have used a set of detectors, described in [7], which observe behavior of network entities, in order to derive the trust values t of the neighbor nodes in different update periods. The confidence values are associated with the different detectors and indicate their accuracy (how "good" are the trust estimates). Hence, in the case that we have high confidence, we should assign the nodes the corresponding trust value close to the estimated trust value t, whereas in the scenario that we obtain low confidence value, we should assign a lower trust value. To depict the effect of the confidence value derived from the detectors to the assigned trust values, we propose the combination of the trust and confidence values into one a unified single metric, called *soft reliability (SR)* estimate. This metric takes value in $[0, 1]$ and it is close to the trust values only in the case that confidence value is high (close to 1). Otherwise, soft reliability estimate is proportionally lower than the derived trust values, based on the confidence values. Fig. 2.1 indicates how the soft reliability (SR) estimates are derived from the tuples (t, c). These estimates are used as an input for our proposed mitigation techniques and indicate whether a

node is observed to be malicious or benign.

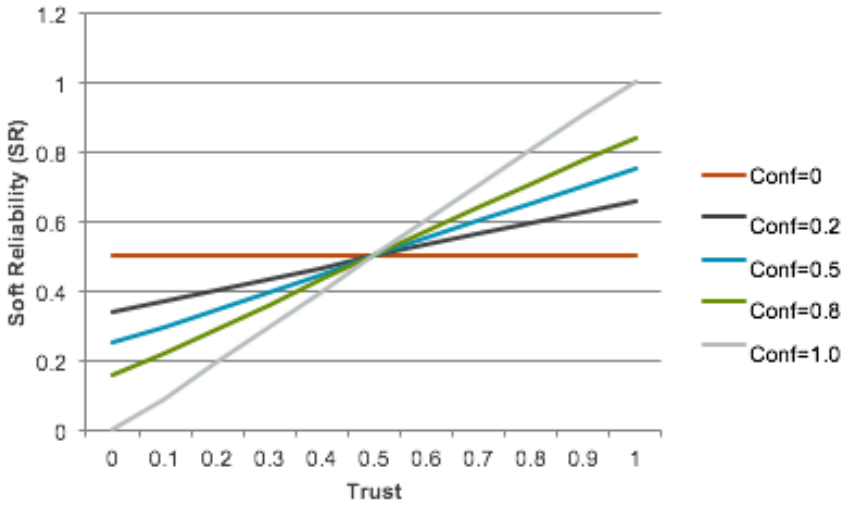


Figure 2.1: Soft Reliability (SR) estimates

2.3 Component-based Protocol Design Approach

Component-based protocol design is the centerpiece of our proposed solution. As we discussed in the introduction of this dissertation this approach defines a collection of elementary modules that can be combined to synthesize protocols with various capabilities. Components are fundamental abstractions of the protocols based on their distinct common functionalities. Hence, we are capable of designing and implementing lightweight mitigation techniques, which can be integrated into specific protocol components to modify appropriately their behavior against different type of attacks. These techniques utilize the updated trust estimates and they do not require heavy cryptographic operations. In addition, they can be easily deployed and reused among different current and future routing protocols with minor modifications.

2.3.1 Decomposition of Proactive Routing Protocols

In this subsection, we propose a decomposition of proactive routing protocols into components according to the protocol's operations. Proactive routing protocols is a class of routing protocols that constructs a priori the paths and therefore the routing tables, based on topology information that is being disseminated periodically across the network. Examples of proactive routing protocols are OLSR [33], DSDV [47] and BATMAN [48]. We decompose the protocol based on the different functionalities of this class of routing protocols and formalize and define the interactions between the separate components. This decomposition of proactive routing protocol into components is presented in [49], [50] and is illustrated in Fig 2.2.

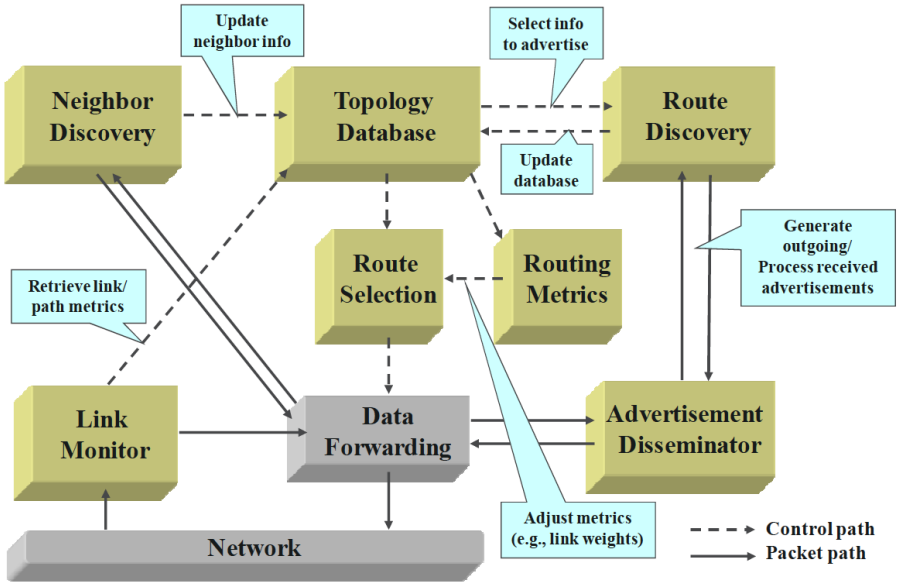


Figure 2.2: Decomposition of Proactive Routing Protocols

We describe the different components, their functionalities and the dependencies between them:

- **Neighbor Discovery**: Describes the operation of the routing protocol to discover its immediate neighbors, which have stable links. A performance model for the neighbor discovery component (NDC) in proactive routing protocols is described in [11] and [51]. The neighbor discovery, as we see in Fig 2.2, has an immediate association with the topology database component, in order to update the set of links stored in the topology database component.

- **Topology Database**: This component is responsible for storing the updated topology information that will be used in the route selection. It has a direct association with neighbor discovery component, route discovery and link monitor. It takes from the neighbor discovery component the immediate neighbors at each time and from the link monitor the information regarding the link weights. It receives from the route discovery component the relay set that it decides to advertise to the rest of the network and the links advertised from the rest of the network to update appropriately the database. It stores all the topology information for the mobile ad-hoc network at each time and each node can create its routing tables in a proactive way, i.e. know the optimal paths before a routing request is instantiated by the source node.

- **Route Discovery**: This component (RDC) is responsible for the selection of topology to disseminate to the network. It uses information through topology database component to find the set of links that the node is going to advertise to the rest of the network through the advertisement disseminator component. Thus, this component executes the pruning algorithms [52] in order to select

the relay set. In the case of OLSR the route discovery component selects the multi-point relays (MPR) of the node [53]. In addition, the route discovery class of components receives the advertisements from the other nodes of the network, process them and sends the updated information to the topology database component. Therefore, at each time in a proactive routing protocol that follows these functionalities, we have an updated topology database that has the information about the advertised links and the link weights. This information can been taken as an input to the route selection component to construct the routing tables of the mobile node.

- **Advertisement Disseminator**: This component sends the topology control packets to the rest of the nodes of the network. Topology control packets contain the set of links that the node wants to advertise to the network to be used for routing selection. It gets the information as an input from route discovery component and it uses the data forwarding component to flood the packets to the rest of the mobile network (through the relay set of the originator node). It also receives the advertisements from the rest of the nodes and it sends it to the route discovery component to process them and send updated information to the topology database component.

- **Route Selection**: This component is responsible for selecting the route for the specific source and destination pair. It takes as an input the updated topology information and the adjusted routing metrics in order to decide the route that it will choose. It uses some optimization framework or some online

learning techniques in order to make the decision. The simplest example is a minimization algorithm on the sum of weights among the set of possibles paths.

- **Routing Metrics**: This component is adjusting the routing metrics for the links according to the routing requirements and the environmental conditions.

- **Data Forwarding**: This component is crucial to forwarding packets to the rest of the network. It stores the routing tables and therefore the least-cost paths information. It is associated with four components: the route selection, the advertisement disseminator, the link monitor and the neighbor discovery component.

2.4 Adversary Models

In this section, we describe the classes of attacks that we attempt to mitigate with our proposed techniques. These attacks influence the functionality of the protocol and affect its performance. Each of these attacks affects different components of the protocol and we want to investigate which mitigation techniques are more efficient against different attacks and environmental conditions (e.g. number of compromised nodes).

There are two main classes of attacks: a) control plane and b) data plane attacks. Control plane attacks aim at manipulating the control messages to disrupt the network operation. Data plane attacks are related to packet forwarding. We investigate both classes of attacks and their impact on the routing protocol operation.

The network-layer attacks that we have examined fall into three main categories of control plane attacks: blackhole/greyhole attack, wormhole attack, and selfish behavior in the control plane. We also implemented one data plane attack, i.e., the selfish behavior in the data plane, which is a type of selective data forwarding attack. These attacks and some of their variations are described in [2] and [3].

2.4.1 Blackhole/Greyhole Attack

In this type of Byzantine attack, the malicious node manipulates the control plane packets by advertising false topology to attract data traffic. The adversary falsely presents himself as the node with the closest proximity to the rest of the nodes by modifying the corresponding control packets (data integrity attack). The attracted traffic can be entirely (blackhole (BH)) or partially (greyhole) eavesdropped, dropped (availability attack), or manipulated. This leads to significant decrease of packet delivery ratio (PDR) and throughput.

The application and impact of the attack differs across diverse routing types. For reactive routing protocols (e.g., AODV, DSR), blackhole/greyhole type of attacks are easier to deploy. Their effectiveness depends on the location of the node with respect to the RREQ messages. For proactive protocols (e.g., OLSR, OSPF, DSDV, BATMAN), this type of attacks is more challenging to implement due to the use of the topology control (TC) or the link state messages. The effectiveness of the attack in proactive protocols depends on the number of malicious nodes and not on the node location.

2.4.2 Wormhole Attack (Fake out-of-band communication links)

This attack requires colluding adversaries who use an out-of-band communication channel. The objective of the colluding adversaries is to advertise that they are one-hop neighbors and influence the shortest path selection algorithm in order to select the links between them as part of the shortest path between the source and destination. This attack impacts the long distance traffic, which will attempt to use this link, and causes packet drop, long end-to-end latency, and eavesdropping.

2.4.3 Packet Relays Attack (Man-in-the-Middle attack)

This attack is a wormhole type of attack, but does not require colluding adversaries. A single malicious node operates in stealth mode and forwards appropriately modified relay control packets between a selected set of one-hop neighbors. When the malicious node receives a HELLO message from any of the neighbors, it adds the IP addresses of the rest of the neighbors and relays this HELLO message, so that the 2-hop neighbors appear to be one-hop away. The behavior of the malicious node will impact the 2-hop neighbors: although they are not directly connected, due to the modified HELLO message received they think that they can communicate with each other.

2.4.4 Selfish Node Behavior (Control Plane)

Selfish behavior (SB) attack, or otherwise defined as packet dropping or selective forwarding attack, on the control plane is an opportunistic attack that affects

the control traffic that passes through the malicious nodes. In selfish behavior attack, the adversary does not try to influence the path of the traffic flow, but just affects the forwarding traffic. When a malicious node has to relay or send control traffic, it drops the traffic deterministically or statistically depending on its objective. In the deterministic drop case, the attacker prunes existing and legitimate links making them unusable. Depending on the location of the malicious node, the network may be partitioned. During the statistical drop of control traffic, legitimate links appear to be unstable, whereby causing multiple transient effects and influencing the convergence and correct operation of the routing protocol. This attack leads to an increase in end-to-end delay, because it modifies the routing procedure.

2.4.5 Selfish Node Behavior (Data Plane)

Selfish behavior (SB) attack, or otherwise defined as packet dropping or selective forwarding attack, can also be launched in the data plane. In this case the malicious node manipulates the data traffic by dropping the packets that it is supposed to forward. This attack leads to the decrease of packet delivery ratio (PDR) and throughput. The adversary can deploy deterministic or statistical packets drops based on her objective. Specifically, in the deterministic case the adversary drops all the traffic so that it can cause high packet losses. However, if there is some monitoring mechanism observing the adversary's behavior, then is straightforward to determine that such an attack is being deployed. On the other hand, the objective of a statistical selfish behavior attack is to cause high packet loss but remain unde-

tected. Hence, in this type of attack a compromised node drops the packets to be forwarded in a probabilistic manner, making the detection much more challenging.

2.5 Network Layer Mitigation Techniques

In our framework, we consider multiple control-plane attacks against various network and link layer protocols, which degrade the network performance. Our objective is to develop a set of component-based mitigation techniques to thwart these attacks. In this section, we propose three mitigation techniques: *trust-aware link weight adjustment (TALWA)*, *redundant packet forwarding (RPF)* and *neighbor discovery gatekeeper (NDG)*. Table 2.1 illustrates the association of our proposed mitigation techniques and other possible mitigation techniques, inspired from prior work, that could be used from our system.

2.5.1 Trust-Aware Link Weight Adjustment

The trust-aware link weight adjustment (TALWA) is associated with the routing metrics components and adjusts the link weights for link-state routing protocols such that the most trusted paths are utilized when the network is under attack and the standard shortest paths are used in the benign state. The routing metrics component obtains trust and confidence values, which are then converted into a single soft reliability value. This mitigation technique is designed and implemented without modifying the shortest path mechanism of routing and can be transparently switched on and off without restarting a routing protocol instance. A similar idea

Table 2.1: Mitigation techniques for protocol components

Protocol Component	**Mitigation Techniques**
Neighbor Discovery	*Neighbor Discovery Gatekeeper (NDG)*
Route Discovery	Randomized route request forwarding [37] k-robust relay set selection [39]
Route Selection	Trusted Routing [54]
Routing Metrics	*Trust-Aware Link Weight Adjustment (TALWA)* Cross layer routing metrics [40]
Topology Database	Reputation-Based Games [42]
Data Forwarding	Secure Message transmission [41] *Redundant Packet Forwarding (RPF)*

for trusted routing was presented in [54].

As shown in Fig. 2.3, TALWA mitigation technique adjusts the weights on the links based on the soft reliability (SR) metric of neighbors in order to route around compromised nodes while ensuring that the weight assignment results in loop-free routes. To prevent significant variation of the soft reliability metric SR_i of node i and to include memory of the evaluation, we suggest using an exponential weighted moving average (EWMA) [55] to update the link weights as a function of the previous and the most updated reliability metric (RC) estimate. The update formula of the weight $w_{i,j}$ for a node i and its link its neighbor j is defined as follows

$$w_{i,j} = (1-\alpha) * \frac{1}{SR_{j,prev}} + \alpha * \frac{1}{SR_{j,new}},$$

where $\alpha \in [0, 1]$ is a constant weight indicating the relative preference between the updated and historic samples of soft reliability metric of the neighbor node j.

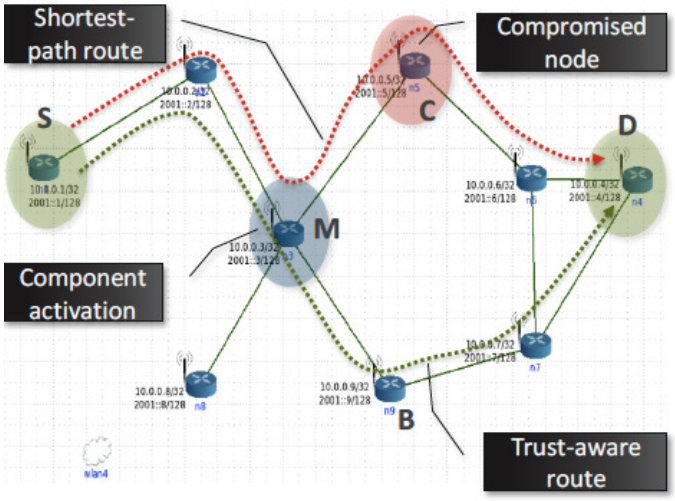


Figure 2.3: TALWA Example

The loop-free property is achieved because the above weight adjustment is *isotonic* [56]. If we add a third path C to two paths (A, B), the relative cost of the two new paths (A + C, B + C) maintains the relative cost of the initial (A, B) paths. Currently, TALWA mitigation technique uses the soft reliability (SR) metric of neighbors to find an alternative path.

TALWA is designed and implemented in a component-based framework, which ensures high *reusability* of the mitigation among different routing protocols. TALWA passes the adjusted links weights directly to the routing metrics component of a routing protocol. These metrics are then used to select routes based on the standard shortest-path calculation. We have implemented and integrated TALWA mitigation in OLSR and BATMAN routing protocols with high reusability.

2.5.2 Redundant Packet Forwarding (RPF)

Redundant packet forwarding (RPF) is associated with the data forwarding component and is only activated for low-throughput and high-priority traffic, which allows differentiating a response to attacks depending on data priorities. The RPF mitigation technique is designed to work with single-path routing protocols. By taking advantage of path diversity, the RPF clones a packet and sends the copy via a secondary neighbor if the primary neighbor is estimated as unreliable. To select the secondary neighbor the current algorithm computes the end-to-end reliability cost of the path of each secondary neighbor to the destination and combines it with the original cost. The end-to-end reliability cost (RC) and the total cost of the path $p_j \in P$ are computed by the two following equations

$$RC_j = 1 - \prod_{i \in p_j} SR_i, \ \forall p_j \in P \tag{2.1}$$

$$Total\_Cost_j = Original\_Cost_j * (1 + 10 * RC_j), \ \forall p_j \in P \tag{2.2}$$

After computing the costs, RPF selects as secondary neighbor the one with the lowest total path cost to the destination. The cloning process is activated along the end-to-end routing path at the first node with observed unreliable primary next hop. However, a packet may be cloned at multiple intermediate nodes on its way to destination in the case where the intermediate next-hops along the data path are unreliable.

The RPF mitigation technique is designed and implemented in a component-

based framework, which ensures high reusability of the mitigation among different routing protocols. It runs as part of the data forwarding component. It applies lightweight control actions that do not involve packet encapsulation or modifications to the routing protocol. However, RPF mitigation introduces significant overhead due to the packet cloning process. Hence, in order to reduce overhead, the RPF also contains the redundancy elimination (RE) module. Since multiple copies of a packet may be in transit, a cloned packet is dropped by RE if the original is also observed at the same node.

2.5.3 Neighbor Discovery Gatekeeper

The goal with this mitigation is the implementation of an external Neighbor Discovery Gatekeeper that interacts with the neighbor discovery component in order to exclude from the neighbor table, the compromised neighbor nodes. It uses the soft reliability metrics (SR) of the neighbor nodes, which as we described are a combination of their (t, c) values, in order to decide which neighbors are compromised and drop the HELLO messages originating from them. Thus, it executes a filtering operation on the standard neighbor discovery component. The mitigated component also utilizes hysteresis in order to cope with noisy reliability estimates. In this way, it takes into account the transition phase in the reliability metrics of the node between the *Benign* and the *Malicious* state, which are defined by some threshold of the SR estimates. These states are used in order to exclude neighbor nodes from the neighbor discovery procedure. This mitigation technique interacts

with the neighbor discovery component of the routing protocol to excludes nodes that are characterized as malicious from the topology database.

By applying the mitigated component for neighbor discovery we observe in the performance evaluation section (Section 2.6) that we achieve significant performance improvement in the case of a severe attack, such as Blackhole attack, where the compromised node advertises itself as being the immediate neighbor of all nodes of the network. Finally, this mitigated component is highly reusable and can be integrated into most existing neighbor discovery protocols with minor modifications.

2.6 Performance Evaluation

We used the Common Open Research Emulator (CORE) [57] as our evaluation environment for network layer mitigations techniques. CORE is a tool for emulating networks on one or more machines. It consists of a GUI for drawing topologies of lightweight virtual machines, and python modules for creating scripts for network emulation. It also allows running unmodified applications and network protocol software since each virtual machine provides a full network stack. Hence, we are capable of using real protocols rather than simulation models to examine the performance of our solutions. We also used MGEN [58], which is a tool that generates real-time traffic patterns for the network, in order to conduct experiments. The tool supports both TCP and UDP test traffic scenarios, but for the purpose of our experimentation we have mainly used UDP traffic.

We have used real routing protocols for experimentation and evaluation. The

different mitigated techniques are initially implemented using OLSR as a case study. We have implemented and evaluated some of them to other proactive routing protocols, such as BATMAN, to indicate the high reusability of our solutions.

2.6.1 Performance evaluation of mitigation techniques

In this subsection, we examine the performance of the individual mitigation technique introduced in this chapter. We show the effectiveness of these techniques under different scenarios and under different types of attacks.

2.6.1.1 Trust-Aware Link Weight Adjustment (TALWA)

The trust-aware link weight adjustment (TALWA) mitigation is shown to be effective when there are alternative trusted (reliable) paths with enough capacity to carry all the traffic from the original (unreliable) path. In this case, the routing metrics adjustment with the TALWA mitigation leads to the total recovery of packet delivery ratio (PDR) and throughput in our scenario. We consider the 9-node topology shown in Fig. 2.4, where there are a lot of compromised nodes deploying selfish behavior attack in data plane (indicated by selfish nodes (SN)). In this scenario the source node with IP address `10.0.0.1` sends UDP traffic to the destination node with IP address `10.0.0.4` with rate 41 Kbps, but we observe significant decrease of the throughput due to the selfish nodes (SN) attacks deployed by the intermediate nodes. After activating TALWA in the source node and one intermediate node, we notice in Fig. 2.5 that the traffic is redirected to the trusted path.

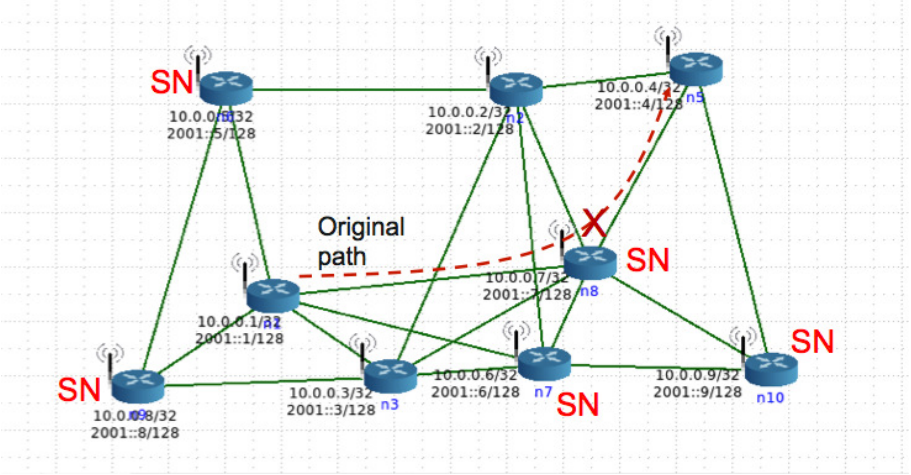


Figure 2.4: Original Scenario for TALWA

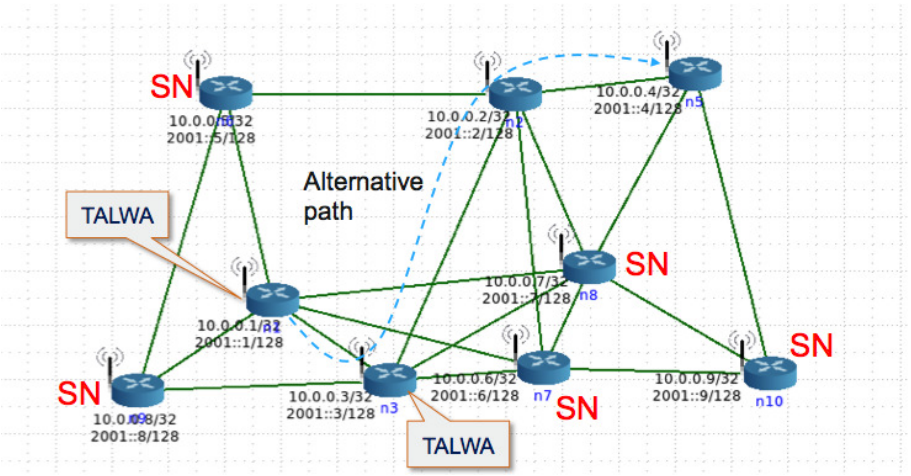


Figure 2.5: Scenario after Mitigation

We also observe in Fig. 2.6 that our system achieves almost total recovery of the throughput after activating the TALWA mitigation technique, which indicates the effectiveness of the mitigated component for scenarios that there is diversity of paths in the network and there are alternative trusted paths.

We have also integrated TALWA into BATMAN routing protocol with minor modifications, which indicates the reusability of our proposed technique. It also achieved high performance recovery in a few seconds in the presence of multiple selfish behavior node attacks in the network.

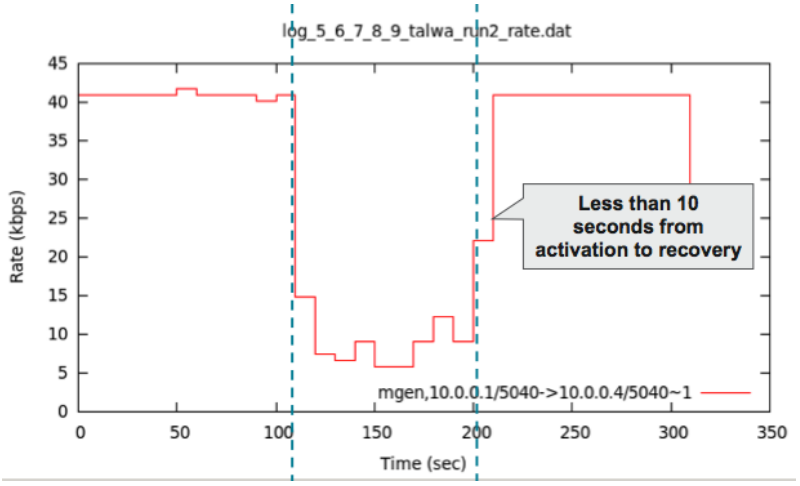


Figure 2.6: Throughput Recovery using TALWA

2.6.1.2 Redundant Packet Forwarding (RPF)

For the performance evaluation of RPF, we consider the 9-node topology shown in Fig. 2.7 that includes three compromised nodes (indicated by SN), along all the possible paths. The source node with IP address `10.0.0.1` sends UDP traffic to the destination node with IP address `10.0.0.4` with rate 1 Kbps. The compromised

intermediate nodes selfishly drop packets in the data plane (selfish behavior attack) with dropping probability equal to 0.5.

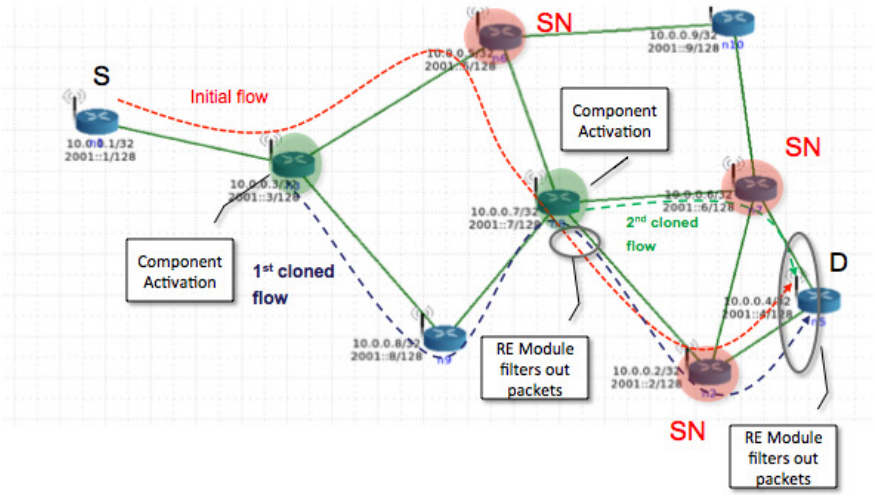


Figure 2.7: RPF Mitigation Scenario

We activate the RPF at the intermediate nodes along the data path to clone data packets on the most trusted alternative routes in order to recover throughput. In the beginning, we activate the RPF in one intermediate node and we observe partial performance recovery and then we activate RPF in a second intermediate node, creating three clones of the initial traffic flow, which leads to almost full recovery of throughput. The results are shown in Fig. 2.8.

2.6.1.3 Neighbor Discovery Gatekeeper (NDG)

To evaluate the performance of the NDG mitigation technique, we used a 9-node topology with one traffic flow. The source node with IP address `10.0.0.1` sends UDP traffic to the destination node with IP address `10.0.0.4` with rate 41

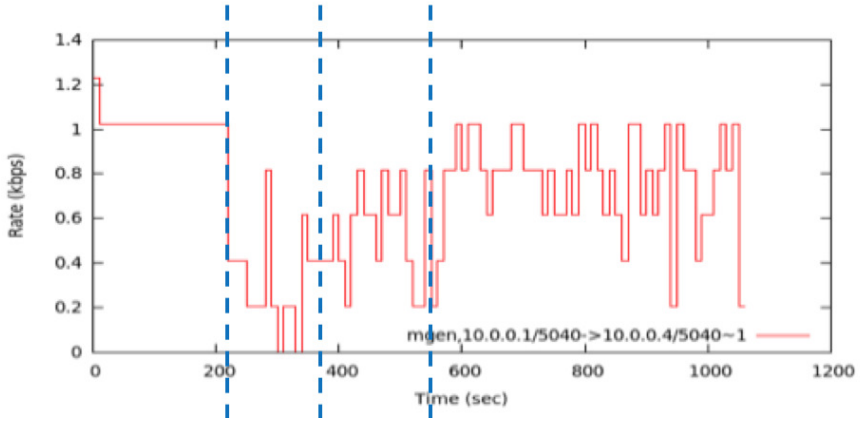


Figure 2.8: Throughput Recovery using RPF

Kbps. In the scenario shown in Fig. 2.9, there are two compromised nodes that launch the combined blackhole and selfish behavior attacks. In the first stage of the attack the malicious nodes deploy blackhole attack (BH) to attract the network traffic and then they release a selfish behavior attack (SN) to actively drop data packets and degrade network performance.

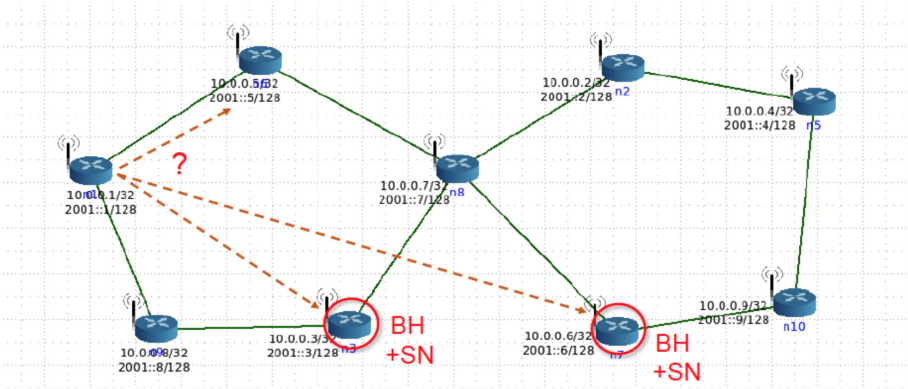


Figure 2.9: NDG Scenario-Combined Blackhole and Selfish Node Attackers

After the ND Gatekeeper mitigation is activated in some of the intermediate nodes of the network, it starts discarding HELLO messages from the malicious nodes (based on the soft reliability (SR) estimates). As a result, these nodes are prevented

from forming routing adjacencies with other nodes. We show the efficiency of this technique against severe attacks in Fig. 2.10. In the beginning there is a small decrease of throughput due to the blackhole attack (BH), then we observe a significant decrease to to the severe deployed selfish behavior attack (SN). After activating NDG mitigation technique, the throughput is quickly restored in 20 seconds.

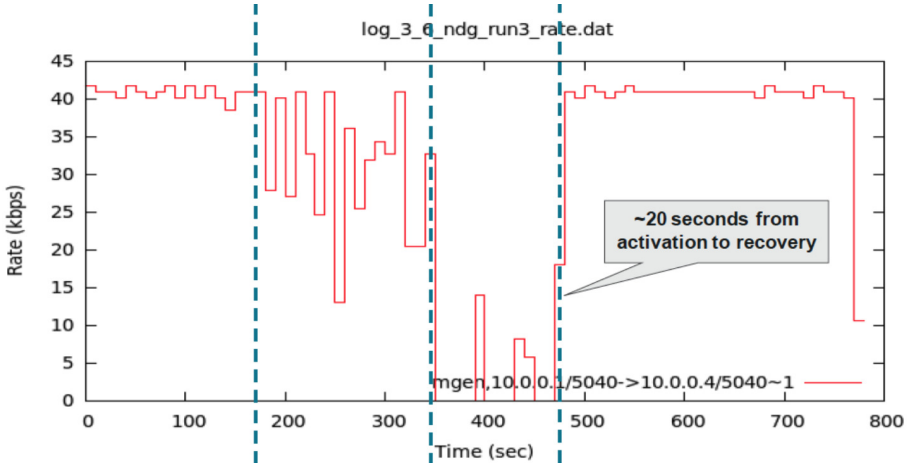


Figure 2.10: Throughput Recovery using NDG

The ND Gatekeeper is an aggressive, but efficient mitigation technique that relies on the exclusion of malicious nodes from the routing procedure. The performance evaluation of this mitigation technique shows the effectiveness against severe attacks that manipulate control traffic, such as blackhole/greyhole attacks and their combinations.

2.6.2 Efficiency of individual mitigation techniques

In this subsection, we examine the efficiency of the three proposed lightweight mitigation techniques based on our performance evaluation results. The mitigation techniques have different effect in the presence of different attacks and environmental

conditions (e.g. path diversity). Table 2.2 summarizes the effect of the different proposed mitigation techniques against different types of attacks and different number of compromised nodes (network state). As we notice from the performance evaluation section TALWA mitigation is effective in less severe attacks and the Neighbor Discovery Gatekeeper mitigation is effective against more severe attacks that also intervene in control plane and cause routing instabilities.

Table 2.2: Efficiency of different mitigation techniques

Attack Type	**Network State**	**Severity**	**Effective Mitigation**
Blackhole	All neighbors Compromised	Difficult to Recover	Neighbor Discovery GateKeeper
Selfish Node Data Plane	Subset of neighbors Compromised	Easy to Recover	Trust-Aware Link Weight Adjustment
Selfish Node Data Plane	All neighbors Compromised	Difficult to Recover	Redundant Packet Forwarding

We use OLSR routing protocol as the case study to implement the mitigation techniques and examine their performance, but we have also implemented some of the mitigation techniques in BATMAN routing protocol in order to show that our mitigation techniques are reusable across different existing protocols. Based on our implementation, the amount of code that is related to a specific routing protocol and needs to be changed for a mitigated component is very low. Therefore, the mitigated components are highly reusable across multiple protocols of the same

class. In Table 2.3 we indicate the implemented mitigation techniques, the associated component that they modify, and the amount of code that can be reused across different protocols of the same class, based on the results that we got from the implementation using the OLSR routing protocol.

Table 2.3: Reusability of mitigation techniques

Mitigation technique	**Component**	**Code reuse**
TALWA	Routing Metrics	1998 / 2070 (96.5%)
RPF	Data Forwarding	3565 / 3780 (94%)
NDG	Neighbor Discovery	3200 / 3260 (98%)

We observe from Table 2.3 that our mitigated components are high reusable across different protocols. To enhance our claim, we have also integrated TALWA into BATMAN routing protocol. The mitigated component has similar effect in the throughput recovery when we use BATMAN routing protocol and the protocol specific changes are 1.6% (23/1399) of the protocol code, which indicates high reusability.

2.7 Concluding Remarks

In this chapter, we present a novel framework for mitigating control and data plane attacks against wireless network protocols. The system takes advantage of the component-based protocol design to change the protocol functionality in a modular way in response to a variety of attacks. We have proposed a set of mitigation tech-

niques, which are incorporated into protocol components, and utilize trust estimates to modify component's functionality. The implemented mitigation techniques are lightweight and highly reusable across different protocols of the same class. The performance evaluation of these techniques indicates high performance recovery in different attack scenarios. As part of the future work, we will develop new mitigation techniques in order to create a library of mitigated components that can be used from our system. In addition, we will investigate the design of a component activation engine, which will enable the automatic activation of mitigation techniques from our library of mitigated components depending on the network state. We have described the operation of the engine as part of our future work in Chapter 7.

CHAPTER 3

Trust-Aware Network Utility Optimization with Delay Constraints

In the previous chapter, we investigated the problem of utilizing trust estimates (called soft reliability metrics) to develop a set of efficient and lightweight mitigation techniques, which can be easily integrated to existing and future protocol components. In this chapter, we utilize the trust estimates to develop a unified framework for resilient cross-layer optimization that aims at secure resource allocation in wireless multi-hop networks. Our optimization framework captures the tradeoff between security, network performance and delay requirements, which are introduced via soft optimization constraints.

3.1 Overview

The problem of resource allocation in wireless networks has been a growing area of research over the past decade. Recent advances in the area of network utility maximization (NUM) driven cross-layer design [59], [60], [61] have led to efforts on top-down development of next generation wireless network architectures. By linking

decomposition of the NUM problem to different layers of the network stack, we are able to design protocols, based on the optimal NUM derived algorithms, which provide much better performance gain over the current network protocols.

Traditionally, network protocols are strictly layered. Source rate control, routing and scheduling (e.g. back-pressure scheduling [62]) are implemented independently at different layers. In order to achieve high performance and efficient resource utilization, these protocols should be jointly designed while the layered structure is preserved. However, the nature of wireless multihop networks imposes new challenges to this cross-layer design, since the wireless channel is a shared medium where the transmissions of users interfere with each other. The channel capacity is "elastic" (time-varying) and the contention over such shared and limited network resources provides a fundamental constraint for resource allocation. All these challenges cause interdependencies across users and network layers. In spite of these difficulties, there have been significant developments in optimization-based approaches that result in loosely coupled cross-layer solutions [63].

In recent years, network security has become increasingly important in the context of wireless multihop networks. Different types of network attacks can be released and affect significantly their performance. In our work, we consider that the adversary is capable of releasing some form of *denial of service* (DoS) attack. Hence, without proper security consideration, the network operation is possible to be disrupted. To capture the notion of security, we use "trust weights" [6] in the network utility optimization process. These weights indicate whether a network entity (node) is malicious or not, based on its interactions with the other network

entities. Thus, by using them, we enhance the correct operation of the network and its resilience to attacks. The trust weights are developed by our network community based on monitoring and are disseminated via efficient methods so that they are timely available to all nodes that need them [64].

End-to-end delay is a critical *quality of service* (QoS) requirement for resource-constrained wireless networks. Network applications, served from different traffic flows in the wireless network, have different delay requirements. For example, video streaming applications are time-critical and have strict delay requirements. Hence, it is crucial to take into account these delay constraints, corresponding to different classes of traffic flows, to our trust-aware NUM problem.

3.1.1 Related Work

Network utility maximization (NUM) problems have been investigated widely during recent years. Most of works [59], [60], [61], [63] focus on using NUM for cross-layer optimization. Chiang et. al [59] introduced a methodology for optimizing functional modules of the network, such as congestion control, routing and scheduling, through optimization decomposition. Chen et. al [60] proposed a sub-gradient algorithm for cross-layer optimization and its extension to time-varying channels and adaptive multi-rate devices. The proposed solution in most of these works depends on the decomposition of the dual function to different subproblems. Decomposition methods for NUM problem are proposed in [65].

Several works have introduced delay considerations for the traffic flows into

the NUM problem formulation. Trichakis et. al [66] proposed a dynamic NUM formulation with delivery contracts for the different traffic flows. Delivery contracts ensure that some quantity of a traffic flow will be delivered during a time interval. One other concept for delay, used for the NUM problem, is the *link capacity margins*, which we use in our work. These margins were introduced in [67] and [68] to control the average end-end delay. Link margins represent the estimated delay of the link, because higher link margin indicates lower link congestion and thus less delay.

As far as we are concerned, there are not a lot of works that relate security with the NUM problem [69], [70]. Tague et. al [70] proposed a jamming-aware throughput maximization approach. The authors estimate the effect of jamming on packet delivery ratio. Then, they use these jamming estimates in the NUM problem to allocate data traffic appropriately in order to achieve throughput maximization. They adopt an objective function, based on portfolio selection theory to maximize throughput for the different source nodes.

To the best of our knowledge, our work is the first to study trust-aware network utility maximization problems. Trust values affect the outcome of the NUM process and make it resilient to malicious nodes' behavior.

3.1.2 Summary of Contributions

In this chapter, we introduce the notion of security into a cross-layer optimization problem (NUM problem), by using the trust values of the network nodes. To the best of our knowledge, this is one of the first initiatives on introducing security in

optimization problems and it is the first step towards creating a unified optimization framework that ensures security and resiliency.

Our optimization framework assigns to the users higher utility, when they relay packets though trusted paths. Hence, our proposed *trust-aware NUM* process ensures that untrusted paths (with malicious entities) do not receive high traffic rate. This is achieved by using the trust estimated incorporated into soft constraints of the optimization problem and not by introducing rules (binary constraints) that most of the prior work utilize to exclude malicious nodes from the network operation. We also add end-to-end delay constraints in the NUM problem based on [67]. These delay constraints indicate the QoS requirements of the different traffic flows. The notion of *link capacity margin* [67] is used to control the end-to-end delay. Finally, we propose a distributed cross-layer optimization algorithm for the trust-aware NUM problem with delay constraints. The distributed algorithm is based on the dual decomposition into source rate control, average end-to-end delay control and scheduling subproblems.

3.1.3 Chapter Organization

The chapter is organized as follows. Section 3.2 introduces the system model that we consider in this work, including the network model, the adversary model, the trust values estimation, and the interference model. Section 3.3 outlines the optimization constraints, which include link capacity, average end-to-end delay, and scheduling constraints, as well as the primal optimization problem. The dual func-

tion and its decomposition into different subproblems is studied in Section 3.4. Section 3.4.4 discusses the distributed algorithm for solving the network utility maximization (NUM) problem. The simulation results for our proposed trust-aware NUM problem with delay constraints are shown in Section 3.5. Section 3.6 concludes this chapter and discusses future work.

3.2 System Model

3.2.1 Network Model

We consider a multihop wireless network that can be defined by a graph $G(\mathcal{N}, \mathcal{L})$. The vertex set $\mathcal{N}$ represents the wireless network nodes. The edge set $\mathcal{L}$ represents the wireless links. An ordered pair of nodes (i, j) belongs to the edge set $\mathcal{L}$ if and only if node j can receive data packets directly from node i. For simplicity, we also use the symbol ℓ to denote a wireless link. We assume that all node-to-node communication is unicast, i.e. each packet transmitted by a node $i \in \mathcal{N}$ is intended for a unique $j \in \mathcal{N}$ with $(i, j) \in \mathcal{L}$. Each of the wireless links has a maximum capacity denoted by $c_{i,j}$. The interference constraints among transmission links will be described in a later subsection.

There is a set $\mathcal{F}$ of network traffic flows that share the wireless network resources and each flow $f \in \mathcal{F}$ is associated with a source node s. Each source node s in a subset $\mathcal{S} \subseteq \mathcal{N}$ generates data packets for a single destination node $d_s \in \mathcal{N}$. We assume that each source node s constructs multiple routing paths with multiple hops to d_s in order to distribute the traffic demand and satisfy the flow related QoS

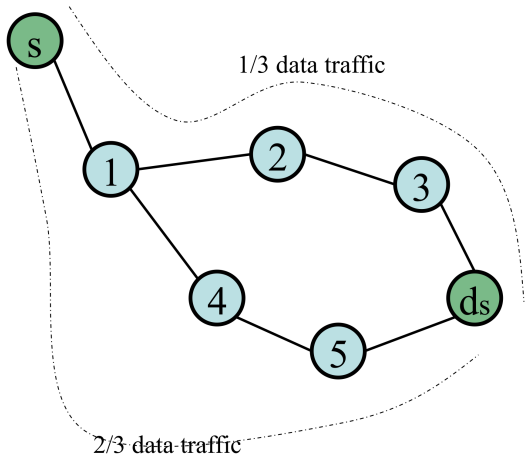


Figure 3.1: Network with two alternative paths for traffic flow from s to d.

requirements. We denote as $\mathcal{P}_s = \{p_{s1}, \ldots, p_{sP_s}\}$ the collection of the alternative paths P_s that can be used to route packets from s to d_s. Each path $p_{sk} \in \mathcal{P}_s$ is specified by a subset of wireless links and is assumed to be loop-free. An example of two different paths p_{s1} and p_{s2} from s to d_s is shown in Figure 3.1, where

$$p_{s1} = \{(s,1),(1,2),(2,3),(3,d_s)\}$$

$$p_{s2} = \{(s,1),(1,4),(4,5),(5,d_s)\}$$

Let $\mathbf{x_s}$ denote the $P_s \times 1$ traffic rate vector with which data packets are sent from s to d_s over multiple paths $p_{sk} \in \mathcal{P}_s$, and multiple hops. Each component of the vector x_{sk} denotes the proportion of traffic rate allocated to the corresponding path p_{sk}, which routes data packets from source node s to destination d_s. The total data rate of the source s is given by the summation of x_{sk} over $k = 1, \ldots, P_s$.

We assume that the traffic rate vector $\mathbf{x_s}$ of each flow is constrained to a non-negative orthant. The traffic rate allocated to each traffic flow should also not

exceed a maximum data rate $\mathcal{R}_s$. Therefore, each of the traffic rate vectors $\mathbf{x_s}$ should satisfy the following constraints

$$\mathbf{x_s} \geq \mathbf{0} \quad , \forall s \in \mathcal{S} \tag{3.1}$$

$$\mathbf{1}^T \mathbf{x_s} \leq \mathcal{R}_s \quad , \forall s \in \mathcal{S} \tag{3.2}$$

In Eq. (3.1) each component of the vector is nonnegative. These constraints define the convex set $\mathcal{X}_s$ of feasible traffic rate vectors $\mathbf{x_s}$ for source node s.

We denote by $\mathbf{R_s} = [(R_s)_{k(i,j)}]_{(P_s \times |\mathcal{L}|)}$ the routing matrix that indicates the different paths from source node s to destination d_s. Element $(R_s)_{k(i,j)}$ of the routing matrix is defined as follows

$$(R_s)_{k(i,j)} = \begin{cases} 1, & \text{if } p_{sk} \text{ passes through link } (i,j). \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

3.2.2 Security Considerations: Adversary Model and Trust

In this chapter, we study the network utility optimization problem with considerations of network security. All previous works on the network utility maximization (NUM) problems assume that nodes operate correctly. For example, intermediate nodes successfully forward all packets, and they follow the routing and scheduling protocols. However, nodes do not always function correctly in reality. They may be compromised by attackers, their communication may be blocked or interfered by attackers, or they may just be misconfigured. Wireless networks are especially vulnerable to attacks because of the inherent properties of the shared wireless medium. Therefore, we believe it is crucial to take the security aspect into consideration in the

NUM problems. We are going to define the adversary model, which describes the capabilities of an adversarial node, and the notion of trust, which indicates whether a node can be considered as trustworthy based on its observed behavior.

Adversary Model: We assume that the adversarial node is not following the network protocol and attempts to disrupt communication by dropping or modifying data packets. In this work, we mainly consider that the adversary is capable of dropping data packets in a deterministic or probabilistic way (selective forwarding attack). This type of attack leads to lack of **availability** of the network and constitutes a *denial of service* (DoS) attack. The DoS attack affects significantly some QoS requirements, such as end-to-end delay and packet delivery ratio. Thus, in order to support *time-critical* applications, the traffic allocation mechanisms should be resilient to these types of attacks. In general, the notion of trust can also address different types of attacks, such as the modification or fabrication of data messages. In this case, trust evaluation should incorporate authentication or inspection (filtering) mechanisms (e.g. Message Authentication Codes (MAC) or Deep Packet Inspection (DPI) [71]) at the receiver and intermediate nodes, in order to define the trustworthiness of a network node.

Trust Estimates: The concept of security, which we adopt to distinguish misbehaving nodes in this work is **trust**. Trust is a very critical concept not only in communication networks, but also in various other networks that involve intelligent decisions, such as social networks. All the connections and communications in these networks imply the existence of trust. Trust integrates with several components of network management, such as risk management, access control and authentication.

Trust management is to collect, analyze and present trust related evidence and to make assessments and decisions regarding trust relationships between entities in a network [72]. The collection of trust evidence and the decision of trust are beyond the scope of this work. We assume that there are mechanisms to efficiently distribute trust evidence, such that duplicates of evidence documents are stored in places where they are most needed. Different approaches of trust evidence distribution could be found in [73] and [74] (use of *network coding* to efficiently distribute trust credentials among network entities). Once the trust evidence is in hand, nodes could evaluate the trustworthiness of other nodes. For instance, in wireless environments, the monitoring mechanisms can help detect the behaviors of neighboring nodes and thus infer their trust values [64]. We define the trust estimated value (or trustworthiness) of node i as ν_i.

There are various ways to represent trust values ν_i numerically. In different trust schemes, continuous or discrete numerical values are assigned to measure the level of trustworthiness for a network entity. For example, in [4] the entity's opinion about the trustworthiness of a digital certificate is defined as a continuous value in $[0, 1]$. Using the same logic for our definition of trust, we denote that it takes a continuous numerical value in $[0, 1]$.

We define an *update period* of the trust estimates denoted by T_{update}. During the update period, represented by the time interval $[t-T_{update}, t]$, the trust evaluation mechanism provides fresh estimates of the trust values for nodes $i \in \mathcal{S}$, based on the interaction between network entities. Each node evaluates trust estimates for its neighbor nodes and then the trust mechanism propagates the trust estimates

throughout the network. Hence, at the time that we need to transmit data packets, we use the trust estimates derived at the latest update period.

In order to prevent significant variation in the trust estimate ν_i of node i and to include memory of the trust evaluation, we suggest using an exponential weighted moving average (EWMA) [55] to update the trust estimate as a function of the previous estimate, as indicated in [70]. Hence, the trust value of node i at time t is given by

$$\nu_i(t) = (1-\alpha)\nu_i(t - T_{update}) + \alpha\nu_i^{new}, \tag{3.4}$$

where $\alpha \in [0, 1]$ is a constant weight indicating the relative preference between updated and historic samples of trust values and ν_i^{new} is the fresh estimate of trust value for node i, given from the trust evaluation mechanism.

Given the trust values for the intermediate nodes across a path p_{sk}, the source node s evaluates the updated *aggregate trust value* for the path $p_{sk} \in \mathcal{P}_s$. The *aggregate trust value* of the path p_{sk} is denoted by t_{sk} and can be expressed as the product of the corresponding trust values along the path as follows

$$t_{sk} = \prod_{j:(i,j)\in p_{sk}} \nu_j \tag{3.5}$$

Source nodes evaluate the aggregate trust values for their alternative routing paths to destination d_s, in order to determine the optimal traffic allocation among the different paths.

One additional parameter that we should consider in the data traffic allocation process is *path reliability* [75]. In our work, path reliability is indicated by the corre-

sponding aggregate trust value t_{sk} over the path p_{sk}, which denotes the proportion of the allocated traffic flow that is actually received at destination node d_s. Hence, in order to maintain the reliability of the network the received traffic rate for each traffic flow should exceed a certain threshold. We denote this threshold for each source node s as $\mathcal{R}_s^{thres}$, which is proportional to the maximum allowable rate $\mathcal{R}_s$. Thus, our allocated traffic rate for each source node should satisfy the reliability constraint expressed as

$$\sum_{p_{sk} \in \mathcal{P}_s} t_{sk} x_{sk} \geq \mathcal{R}_s^{thres}, \quad \forall s \in \mathcal{S} \tag{3.6}$$

The convex set $\mathcal{X}_s$ of feasible traffic rate vectors for source node s should also satisfy the above reliability constraint.

3.2.3 Interference model and capacity region

In this subsection, we describe the interference model and the feasible capacity region. In order to model interference among wireless links of our original wireless multihop network, we use the concept of the *conflict graph* introduced in [76]. The conflict graph captures the contention relations among the links. Each vertex in the conflict graph indicates a wireless link and each edge indicates the interference between the two corresponding links.

We can detect all the independent sets of vertices in the conflict graph. We denote an independent set of links by e. The independent set e can be represented

as a $|\mathcal{L}| \times 1$ capacity vector r^e. The element $r^e_{i,j}$ is expressed as

$$r^e_{i,j} = \begin{cases} c_{i,j}, & \text{, if } (i,j) \in e. \\ 0 & \text{, otherwise.} \end{cases} \tag{3.7}$$

The links that belong in an independent set do not interfere and are allowed to transmit simultaneously. The feasible capacity region Λ [60] is defined as the convex hull of these capacity vectors and is expressed as

$$\Lambda = \left\{ r : r = \sum_e \beta_e r^e, \beta_e \geq 0, \sum_e \beta_e = 1 \right\} \tag{3.8}$$

Hence, the scheduling constraint indicates that the allocated capacity vector from the scheduling process, denoted by $\hat{\mathbf{c}}$, should satisfy $\hat{\mathbf{c}} \in \Lambda$.

3.3 Network Utility Maximization (NUM) Formulation

In this section, we present the optimization framework for trust-aware network utility maximization (NUM), in the case that the network nodes have an updated estimate of trust values. We first develop a set of constraints imposed to our utility optimization problem. These constraints are related to capacity of wireless links, average end-to-end delay and scheduling. Then, we formulate the trust-aware utility optimization problem, which gives an optimal solution to the traffic flow allocation problem.

3.3.1 Optimization Constraints

Link Capacity constraint: To define capacity constraints we first introduce the *link capacity margin* optimization variables, which were initially introduced

in [67] and [68], in order to capture the imposed delay constraints. We denote by $\sigma_{i,j}$ or simply σ_ℓ, the link capacity margin of link $(i,j) \in \mathcal{L}$. Link capacity margin is defined as the difference between scheduled (allocated) capacity of a wireless link and the maximum allowable traffic flow passing though it. Link capacity margin is used to control link delay and therefore the average end-to-end delay.

We also need to take into account our trust estimates for the capacity constraints of each link $(i,j) \in \mathcal{L}$. Based on the capabilities of the malicious nodes, described in Section 3.2.2, the initially allocated traffic rate x_{sk} can be significantly reduced at malicious intermediate nodes because of dropping attacks. The decrease of the traffic rate is proportional to the *aggregate trust value* of the selected path. To be more specific, the decrease of the rate observed at an intermediate node is proportional to the aggregate trust value up to this intermediate node. Let $p_{sk}^{(i,j)}$ denote the sub-path of p_{sk} from source node s to the intermediate node j through link $(i,j) \in p_{sk}$. Then the traffic rate forwarded by intermediate node $j \in \mathcal{N}$ is computed by $t_{sk}^{(i,j)} x_{sk}$, where $t_{sk}^{(i,j)}$ is evaluated as the product of trust estimates over the sub-path $p_{sk}^{(i,j)}$, given by Eq. (3.5).

Hence, the capacity constraint associated with each wireless link $(i,j) \in \mathcal{L}$ is formulated as follows

$$\sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} t_{sk}^{(i,j)} x_{sk} \leq \hat{c}_{i,j} - \sigma_{i,j}, \quad \forall (i,j) \in \mathcal{L}, \tag{3.9}$$

where $\hat{c}_{i,j}$ is the capacity allocated to the wireless link $(i,j) \in \mathcal{L}$.

To define the different sub-paths' aggregate trust values, we denote by $\mathcal{T}_s$ the $P_s \times |\mathcal{L}|$ *aggregate trust incidence matrix* for source s, with rows indexed by the

alternative paths p_{sk} and columns indexed by links (i,j). If a link (i,j) does not belong to any of the possible paths p_{sk} for source s, then the corresponding entry of the incidence matrix is equal to 0. The element $t(p_{sk},(i,j))$ or otherwise $t_{sk}^{(i,j)}$ for row p_{sk} and column (i,j) of $\mathcal{T}_s$ denotes the aggregate trust value of a possible sub-path $p_{sk}^{(i,j)}$ of path p_{sk} and is given by

$$t(p_{sk},(i,j)) = \begin{cases} \prod\limits_{j':(i',j')\in p_{sk}^{(i,j)}} \nu_{j'} & ,\text{if } (i,j)\in p_{sk} \\ 0 & ,\text{otherwise} \end{cases} \tag{3.10}$$

Average end-to-end delay constraint: By using the link margin variables $\sigma_{i,j}$, we define as $\phi(\sigma_{i,j})$ the delay of link $(i,j)\in\mathcal{L}$. The function $\phi(\cdot)$ is typically a strictly convex, nonnegative valued, function of σ. The packet arrival process model determines the way that $\phi(\cdot)$ depends on $\sigma_{i,j}$. As described in [67] and [68], for Poisson process arrival, we have

$$\phi(\sigma_{i,j}) = \phi_{i,j} = \frac{1}{\sigma_{i,j}} \tag{3.11}$$

We define by $\phi(\sigma)$ the vector that has components the delay of all links of the network.

Delay constraints indicate the QoS requirements imposed to a specific traffic flow. Traffic flows that serve time-critical applications should have strict delay constraints. The end-to-end delay is expressed by adding the link delays for each of the links over path p_{sk} of source node s. We denote the upper bound average delay constraint for each of the multiple paths of the source node s as $\mathcal{D}_s > 0$. Hence, we have that the average end-to-end delay constraint for every source node s is given

by (using the routing matrix expressed in Eq. (3.3))

$$\mathbf{R_s}\phi(\sigma) \leq \mathbf{1}\mathcal{D}_s, \quad \forall s \in \mathcal{S} \tag{3.12}$$

Scheduling constraint: The capacity $\hat{c}_{i,j}$ allocated to the wireless link (i, j) should lie on the capacity region specified by Λ, which we describe in Sec. 3.2.3. Hence, our scheduling constraint is expressed as

$$\hat{\mathbf{c}} \in \Lambda \tag{3.13}$$

3.3.2 Utility Optimization

To determine the optimal traffic rate allocation to the different paths $\mathcal{P}_s$, each source s chooses a utility function $\mathcal{U}_s(\cdot)$ that evaluates the total data rate delivered to the destination d_s. Utility functions $\mathcal{U}_s(\cdot)$ are chosen to be strictly concave, continuous, monotonically increasing and twice differentiable.

Trust estimates for the different paths p_{sk} of a source node (defined in Sec. 3.2.2) should be incorporated to the selected utility function $\mathcal{U}_s(\cdot)$. Source nodes should obtain greater utility when they decide to allocate higher traffic rate through routing paths with higher aggregate trust value t_{sk}. Hence, the utility function for each source node $s \in \mathcal{S}$ can be selected as

$$\mathcal{U}_s(\mathbf{x}_s) = \sum_{p_{sk} \in \mathcal{P}_s} \Big(t_{sk} \log\left(x_{sk}\right) \Big) \tag{3.14}$$

The *primal utility optimization problem* formulation, based on the capacity, average end-to-end delay and scheduling constraints described in Eq. (3.9), (3.12)

and (3.13), is given by

$$\max_{\mathbf{x},\sigma,\hat{\mathbf{c}}} \quad \sum_{s\in\mathcal{S}} \mathcal{U}_s(\mathbf{x}_s) \tag{3.15a}$$

$$\text{s. t.} \quad \sum_{s\in\mathcal{S}} \sum_{k:(i,j)\in p_{sk}} t_{sk}^{(i,j)} x_{sk} \leq \hat{c}_{i,j} - \sigma_{i,j}, \forall (i,j) \tag{3.15b}$$

$$\mathbf{R_s}\phi(\sigma) \leq \mathbf{1}\mathcal{D}_s, \qquad \forall s \in \mathcal{S} \tag{3.15c}$$

$$0 \leq \mathbf{1}^T \mathbf{x_s} \leq \mathcal{R}_s, \qquad \forall s \in \mathcal{S} \tag{3.15d}$$

$$\sum_{p_{sk}\in\mathcal{P}_s} t_{sk} x_{sk} \geq \mathcal{R}_s^{thres}, \quad \forall s \in \mathcal{S} \tag{3.15e}$$

$$\hat{\mathbf{c}} \in \Lambda \tag{3.15f}$$

The trust-aware utility optimization problem is a strongly convex optimization problem, due to the strict concavity assumption of $U_s(\cdot)$ and the convexity of the capacity region. Therefore, there exists a unique optimal solution for the above primal problem, which we refer to as $(\mathbf{x}^*, \sigma^*, \hat{\mathbf{c}}^*)$.

3.4 Dual Decomposition Algorithm

In this section, we solve the utility optimization problem described in Eq. (3.15a) by applying dual decomposition [65], [77]. The decomposition of the optimization problem provides distributed algorithms, which solve the underlying optimization problem. We note that strong duality holds for our optimization problem (duality gap is zero) and thus we can solve it through its dual function.

We define the Lagrange multipliers (dual variables) associated with the capacity and average end-to-end delay constraints. Let λ denote the $|\mathcal{L}| \times 1$ vector of *link prices* (dual variables) $\lambda_{i,j}$ (otherwise denoted by λ_ℓ) associated with the

capacity constraints for each wireless link. Also, let μ_s denote the $P_s \times 1$ vector of dual variables μ_{sk} associated with the average end-to-end delay constraints imposed to every traffic flow $s \in \mathcal{S}$.

In order to introduce the dual problem, we define the partial Lagrangian $L(\mathbf{x}, \sigma, \hat{\mathbf{c}}, \lambda, \mu)$ of the optimization problem by using the inequality constraints given from Eq. (3.15b) and (3.15c)

$$\begin{aligned} L(\mathbf{x}, \sigma, \hat{\mathbf{c}}, \lambda, \mu) &= \sum_{s \in \mathcal{S}} \mathcal{U}_s(\mathbf{x}_s) - \sum_{(i,j) \in \mathcal{L}} \lambda_{i,j} \Big(\sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} \Big(t_{sk}^{(i,j)} x_{sk} \Big) - \hat{c}_{i,j} + \sigma_{i,j} \Big) \\ &\quad - \sum_{s \in \mathcal{S}} \mu_s^T \Big(\mathbf{R_s} \phi(\sigma) - \mathbf{1}\mathcal{D}_s \Big) \\ &= \sum_{s \in \mathcal{S}} \Big(\mathcal{U}_s(\mathbf{x}_s) - \sum_{(i,j) \in \mathcal{L}} \lambda_{i,j} \sum_{k:(i,j) \in p_{sk}} \Big(t_{sk}^{(i,j)} x_{sk} \Big) \Big) \\ &\quad - \sum_{(i,j) \in \mathcal{L}} \Big(\phi_{i,j} \Big(\sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} \mu_{sk} [(R_s)_{k(i,j)}] \Big) + \lambda_{i,j} \sigma_{i,j} \Big) + \lambda^T \hat{\mathbf{c}} + \sum_{s \in \mathcal{S}} \mu_s^T \mathbf{1}\mathcal{D}_s = \\ &= \sum_{s \in \mathcal{S}} \Big(\mathcal{U}_s(\mathbf{x}_s) - (\lambda^s)^T \mathcal{T}_s^T \mathbf{x}_s \Big) - \sum_{(i,j) \in \mathcal{L}} \Big(\phi_{i,j} \mu^{(i,j)} + \lambda_{i,j} \sigma_{i,j} \Big) + \lambda^T \hat{\mathbf{c}} + \sum_{s \in \mathcal{S}} \mu_s^T \mathbf{1}\mathcal{D}_s, \end{aligned} \tag{3.16}$$

where λ^s is a sub-vector of the λ dual variable and is associated with the constraint in Eq. (3.15b). It defines the $|\mathcal{L}| \times 1$ column link price vector related to the links that belong to any of the paths $p_{sk} \in \mathcal{P}_s$ of a particular source node s and is given by

$$\lambda_{i,j}^s = \begin{cases} \lambda_{i,j} & \text{, if } (i,j) \in \bigcup\limits_{p_{sk} \in \mathcal{P}_s} p_{sk} \\ 0 & \text{, otherwise} \end{cases} \tag{3.17}$$

and $\mu^{(i,j)} = \sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} \mu_{sk} [(R_s)_{k(i,j)}]$ denotes the combination of dual variables μ, which are related to a specific link (i,j) and is associated with the constraint (3.15c).

The dual objective function $h(\cdot)$ is then expressed as

$$h(\lambda,\mu) = \sup_{\mathbf{x}\in\mathcal{X}}\Big\{\sum_{s\in\mathcal{S}}\Big(\mathcal{U}_s(\mathbf{x}_s) - (\lambda^s)^T\mathcal{T}_s^T\mathbf{x}_s\Big)\Big\} \tag{3.18a}$$

$$+\sup_{\sigma\geq\mathbf{0}}\Big\{-\sum_{(i,j)\in\mathcal{L}}\Big(\phi_{i,j}\mu^{(i,j)} + \lambda_{i,j}\sigma_{i,j}\Big)\Big\} \tag{3.18b}$$

$$+\sup_{\hat{c}\in\Lambda}\Big\{\lambda^T\hat{\mathbf{c}}\Big\} \tag{3.18c}$$

$$+\sum_{s\in\mathcal{S}}\mu_s^T\mathbf{1}\mathcal{D}_s \tag{3.18d}$$

The dual optimization problem is defined by minimizing the dual objective function [78] over the dual vector variables λ and μ as follows

$$\min_{\lambda\geq\mathbf{0},\mu\geq\mathbf{0}} \quad h(\lambda,\mu) \tag{3.19}$$

For given dual variables λ and μ, we can identify in the above equation of $h(\lambda,\mu)$ three decoupled maximization problems which we can solve separately. These three problems correspond to source rate control in Eq. (3.18a), average end-to-end delay control in Eq. (3.18b), and scheduling in Eq. (3.18c) respectively.

By solving these three independent optimization problems we can derive the optimal values for the primal optimization problem $\mathbf{x}^*(\lambda,\nu)$, $\sigma^*(\lambda,\mu)$ and $\hat{\mathbf{c}}^*(\lambda,\mu)$ (described in Eq. (3.15a)). Given these values, we can then solve the dual problem by minimizing $h(\lambda,\mu)$ over $\lambda,\mu\geq\mathbf{0}$. There is no duality gap between the primal and the dual, because the capacity region Λ [60], [76] is a convex set.

In the following subsections, we describe the decomposition of the dual objective function that leads to the cross-layer optimization problem and we specify the optimal solutions by solving these independent subproblems.

3.4.1 Source rate control

Based on the dual decomposition the traffic rate vector of source node s is determined by the first maximization subproblem in Eq. (3.18a). $U_s(\cdot)$ is a strictly concave scalar function of the rate vector variable x_s. The maximization problem in (3.18a) is maximization of a concave function subject to the convex constraints (3.15d) and (3.15e). Thus, it has a unique solution. $U_s(\cdot)$ is continuously differentiable. Hence, the maximum will be given by the numerical solution of the equation

$$\nabla \mathcal{U}_s(x_s^*) = \mathcal{T}_s \lambda^s, \quad \forall s \in \mathcal{S} \tag{3.20}$$

as long as the resulting solution for x_s^* is in the interior of the constraint set defined by (3.15d) and (3.15e). Otherwise the solution will lie at the corners of the constrained set defined by (3.15d) and (3.15e).

It is important to note that each source node s is able to adjust its data rate vector using its local observations on the link prices $\lambda_{i,j}$ across the links of its multiple routing paths and the aggregate trust values of the different paths and their respective sub-paths.

3.4.2 Average End-to-End Delay Control

The second subproblem of the dual decomposition described in Eq. (3.18b) is related to average end-to-end delay control based on the optimal values for the link capacity margin $\sigma_{i,j}$. Eq. (3.18b) is a strictly convex, minimization problem, subject

to the constraint that all sigma are nonnegative. Thus, it has a unique solution. Function $\phi(\cdot)$ is a continuously differentiable function. Hence, the optimal values of $\sigma^*_{i,j}$ are obtained by solving the equations numerically

$$\frac{d\phi}{d\sigma}(\sigma^*_{i,j})\mu^{(i,j)} = -\lambda_{i,j}, \quad \forall (i,j) \in \mathcal{L} \tag{3.21}$$

By Eq. (3.21), we observe that the updated dual variable related to the corresponding wireless link is needed. Thus, explicit/implicit exchange of dual variables between the different data sources is enabled.

3.4.3 Scheduling policy

The third problem of the dual decomposition determines the scheduling policy. The optimal value for the allocated link capacity $\hat{c}^*_{i,j}$ is given by Eq. (3.18c)

$$\hat{c}^*_{i,j} = \underset{\hat{c}_{i,j} \in \Lambda}{\operatorname{argmax}} \sum_{(i,j)\in\mathcal{L}} \lambda_{i,j}\hat{c}_{i,j} \tag{3.22}$$

We need to find a scheduling policy so that the aggregate link weight $\sum_{(i,j)\in\mathcal{L}} \lambda_{i,j}\hat{c}_{i,j}$ could be maximized. The solution to this scheduling subproblem is based on the *maximum weight* scheduling policy introduced in [67]. This policy is described in Alg. 1 presented below.

3.4.4 Distributed Algorithm

In this section, we describe the distributed algorithm that solves the network utility optimization problem. Our solution is based on subgradient descent iterative

Algorithm 1 Maximum-Weight Scheduling Policy [67]

Start scheduling timer
while Timer is ON **do**
  if i has been scheduled by any of its neighboring nodes **then**
    Send messages to notify all nodes in its interference set and stop the process.
  else
    Each incident link (i, j) is assigned a weight $\lambda_{i,j}\hat{c}_{i,j}$.
    Find a link with the maximum weight among all incident links.
    if link (i, j) is found **then**
      Schedule the link with the effective allocated capacity $\hat{c}_{i,j}$.
      Notify all neighbors in the interference set.
    else
      Stop the process
    end if
  end if
end while

methods for the update of the dual variables. We first compute the subgradient of the dual function with respect to each of the dual variables and then we propose our distributed algorithm.

The subgradient of $h(\lambda, \mu)$ with respect to dual variable $\lambda_{i,j}$ is given by

$$\frac{\partial h}{\partial \lambda_{i,j}} = \hat{c}_{i,j} - \sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} \left(t_{sk}^{(i,j)} x_{sk} \right) - \sigma_{i,j}, \forall (i,j) \tag{3.23}$$

The subgradient of $h(\lambda, \mu)$ with respect to dual variable μ_{sk}, which is related to source node s and its respective routing path p_{sk}, is expressed as

$$\frac{\partial h}{\partial \mu_{sk}} = \mathcal{D}_s - \sum_{(i,j) \in \mathcal{L}} \phi_{i,j}(R_s)_{k(i,j)}, \forall p_{sk} \in \mathcal{P}_s, \forall s \tag{3.24}$$

In order to solve the dual problem of Eq. (3.19), we use a subgradient descent iteration method [78] to update at each iteration n the dual variables (Lagrangian multipliers) as follows

$$\lambda_{i,j}^{(n+1)} = \left\{ \lambda_{i,j}^{(n)} - \gamma \left(\hat{c}_{i,j}^{(n)} - \sum_{s \in \mathcal{S}} \sum_{k:(i,j) \in p_{sk}} \left(t_{sk}^{(i,j)} x_{sk}^{(n)} \right) - \sigma_{i,j}^{(n)} \right) \right\}^{+} \tag{3.25}$$

$$\mu_{sk}^{(n+1)} = \left\{ \mu_{sk}^{(n)} - \gamma \left(\mathcal{D}_s - \sum_{(i,j) \in \mathcal{L}} \phi(\sigma_{i,j}^{(n)})(R_s)_{k(i,j)} \right) \right\}^{+}, \tag{3.26}$$

where γ is a positive step-size that ensures convergence of the iterative solution (e.g. $\gamma = 0.01$) and $(v)^+ = \max(0, v)$ is the projection to the non-negative value.

Based on the primal and dual variable updates of Eq. (3.20), (3.21), (3.22), (3.25) and (3.26), we propose a distributed optimization algorithm, described in Alg. 2.

3.5 Simulation Results

In this section, we present simulation results for our trust-aware network utility maximization problem. Fig. 3.2 represents the sample wireless network scenario.

Algorithm 2 Distributed Cross-Layer Optimization

INITIALIZE primal and dual variables

while $\mathbf{1}^T|\mathbf{x_s^{(n)}} - \mathbf{x_s^{(n-1)}}| \leq \epsilon$ **do**

Dual Variables Update

Each link (i, j) updates its dual variable $\lambda_{i,j}$ (Eq. (3.25)).

Each source s updates the dual variables μ_{sk} (Eq. (3.26)).

Sources exchange dual variables

Each source s evaluates $\lambda^{s,(n)}$.

Each source s computes its traffic rate vector $\mathbf{x_s^{(n)}}$ by solving Eq. (3.20).

Each link (i, j) evaluates $\mu^{(i,j),(n)}$.

Each link (i, j) computes its $\sigma_{\mathbf{i,j}}^{\mathbf{(n)}}$ by solving Eq. (3.21).

Each node performs scheduling via Eq. (3.22) as in [67].

end while

The wireless network contains $\mathcal{N} = 8$ nodes and $\mathcal{L} = 11$ links, with maximum allowable capacity $c_{i,j}$ chosen in $[9, 11]$ *Kbps*. There is one traffic flow from s to d_s, which allocates traffic to different routing paths. Our simulation time is $T = 160$ time slots. The end-to-end delay constraint for the traffic flow is $\mathcal{D}_s = 2$ *msec*.

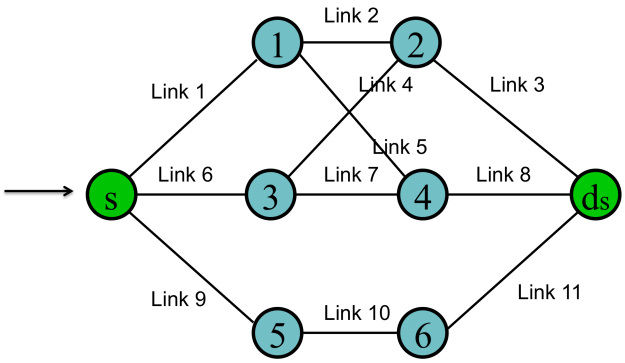


Figure 3.2: Wireless Network Scenario

There are five different paths, where the source node s can allocate data traffic to send to destination node d_s. The paths p_{sk} are

$$p_{s1} = \{(s, 1), (1, 2), (2, d_s)\}$$

$$p_{s2} = \{(s, 1), (1, 4), (4, d_s)\}$$

$$p_{s3} = \{(s, 3), (3, 2), (2, d_s)\}$$

$$p_{s4} = \{(s, 3), (3, 4), (4, d_s)\}$$

$$p_{s5} = \{(s, 5), (5, 6), (6, d_s)\}$$

We define four trust *update periods* (each period is defined every T_{update} time slots), in order to show the behavior of our approach for different trust values. For the simulations, we define $T_{update} = T/4 = 40$ time slots. Node trust estimates ν_i change dynamically at every update period, based on the trust evaluation mech-

anism. The different node trust values that we obtain from the trust evaluation mechanism for each of the four update periods are shown at the matrix below

$$\nu = \begin{array}{c} \begin{array}{cccccccc} s & 1 & 2 & 3 & 4 & 5 & 6 & d_s \end{array} \\ \begin{pmatrix} 1 & 1 & 1 & 0.7 & 1 & 0.7 & 0.5 & 1 \\ 1 & 0.9 & 0.9 & 0.5 & 0.9 & 0.2 & 0.2 & 1 \\ 1 & 0.9 & 0.9 & 0.3 & 0.7 & 0.1 & 0.1 & 1 \\ 1 & 0.9 & 0.9 & 0.2 & 0.5 & 0.1 & 0.1 & 1 \end{pmatrix} \end{array} \tag{3.27}$$

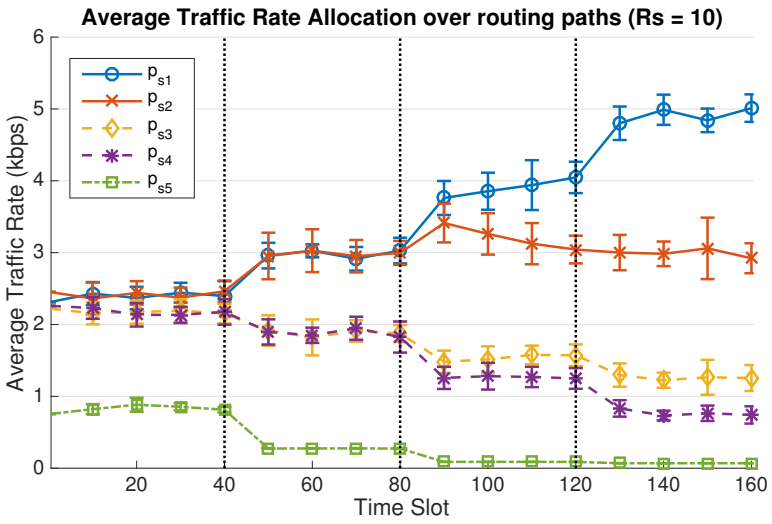
Trust values are adjusted using the EWMA algorithm expressed in Eq. (3.4), in order to prevent significant variations in the trust estimates over subsequent trust update periods. For our simulation, the EWMA algorithm uses $\alpha = 0.8$ to give more significance to the latest update.

Given the trust values estimates in Matrix (3.27), we can notice that path p_{s5} contains untrusted (malicious) nodes and should ideally be excluded from the traffic rate assignment. In addition, node 3 is detected to be malicious and hence our mechanism should ideally assign significantly less traffic to the paths p_{s3} and p_{s4} that contain this node. Finally, node 4 obtains a low trust value estimate at the last update period, which should lead to decrease in the traffic rate assignment even for path p_{s2} that contains this node.

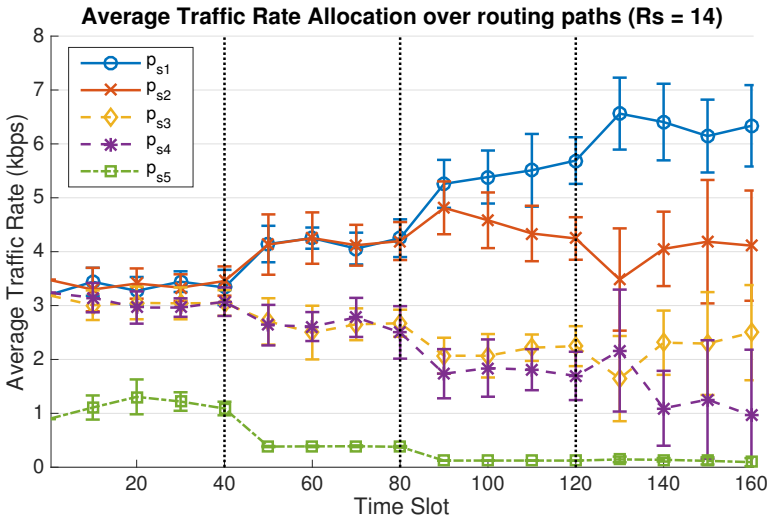
Figures 3.3a and 3.3b present the numerical results of the average traffic rate allocation for two different cases of maximum allowable traffic rate $\mathcal{R}_s$ (with the corresponding error bars). In the case of $\mathcal{R}_s = 10$ *Kbps*, the maximum traffic rate is close to the maximum allowable capacity of the wireless links, while in the case

of $\mathcal{R}_s = 14$ *Kbps*, the maximum traffic rate is greater than the maximum capacity of the links. We observe that in both cases the traffic rate assigned to each routing path changes at every update period based on the trust estimates. Our algorithm assigns to the path p_{s1} the maximum traffic rate, since it contains trusted nodes and to the path p_{s5} the lowest traffic rate, because it consists of untrusted nodes. For the rest of the paths, the traffic rate is being adjusted according to trust estimates of every update period. We also observe that in the case of $\mathcal{R}_s = 14$ *Kbps*, more traffic rate is allocated to untrusted paths to cover the demand.

Link capacity margins $\sigma_{i,j}$ for some links of our wireless network, in the case of $\mathcal{R}_s = 10$ *Kbps* and $\mathcal{R}_s = 14$ *Kbps*, are presented in Figure 3.4a and Figure 3.4b respectively. Link capacity margin is related with the average delay, since higher capacity margin indicates lower link delay and thus lower end-to-end delay. In our scenario, link 1 has the lowest capacity margin, because our scheme allocates significantly high traffic rate to this link. In addition, we notice that links belonging to untrusted paths have high capacity link margin (e.g. link 7 and 11), since they do not relay high data traffic. We also observe that in the case of higher maximum data rate, the link margin takes lower values at the more congested links, because of the higher traffic rate that should be allocated. Links that belong to untrusted paths, such as link 5, are being allocated more traffic rate in the case of higher data rate, in order to satisfy the underlying delay requirements. In general, the capacity margin is being adjusted in order to attain the delay constraints of the traffic flow, which in our scenario are being satisfied, even if our scheme has to reduce significantly the capacity margin of some wireless links.

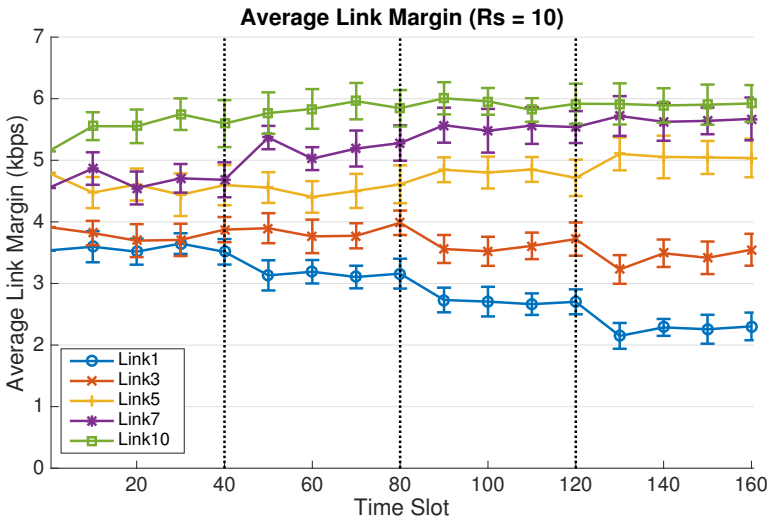


(a) Average Traffic Rate with $\mathcal{R}_s = 10$

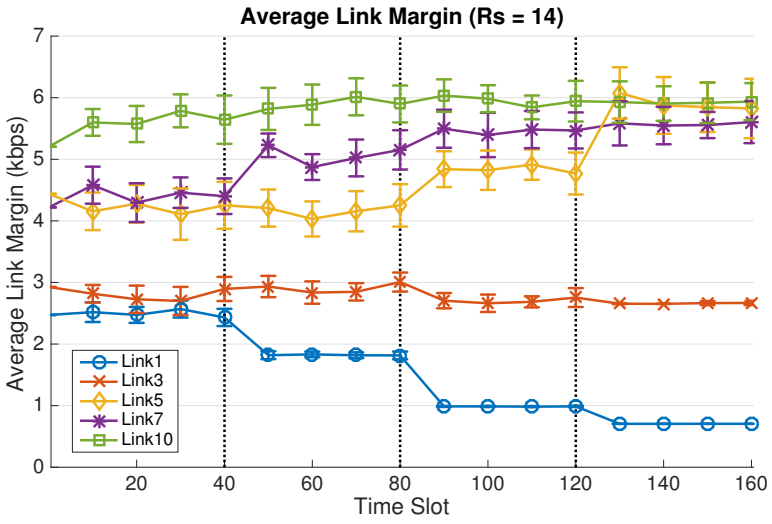


(b) Average Traffic Rate with $\mathcal{R}_s = 14$

Figure 3.3: Average Traffic Rate over paths for different maximum rates $\mathcal{R}_s$



(a) Average Link Margin with $\mathcal{R}_s = 10$



(b) Average Link Margin with $\mathcal{R}_s = 14$

Figure 3.4: Average Link Margin in wireless links over time

3.6 Concluding Remarks

In this chapter, we investigated an important application of performance and security tradeoff by introducing security considerations in the cross layer design of network protocols via network utility maximization (NUM). The specific concept of security we used is *trust*. Users get higher utility by transmitting data through nodes of higher trust values. Thus, trust values should be taken into account as parameters in the optimization problem, so that the resulting trust-aware protocols are resilient to network failures and to possible attacks. We also incorporated delay constraints in the utility optimization problem to capture QoS requirements. Finally, we proposed a distributed algorithm that achieves network utility maximization. As part of future work, we plan to investigate how dynamic changes in trust values affect the utility optimization problem, and to evaluate our approach in large scale scenarios.

CHAPTER 4

Distributed Sleep Management for Heterogeneous Wireless Machine-to-Machine Networks

In the previous two chapters, we investigated two security and resiliency problems in wireless multi-hop networks. Beginning from this chapter, we will provide efficient solutions for energy preservation in wireless multi-hop networks. In this chapter, we propose a distributed sleep management and some novel routing metrics that can be integrated to a standard routing protocol for machine-to-machine (M2M) networks in order to reduce energy consumption.

4.1 Overview

Machine-to-machine (M2M) communications are part of the Internet of Things (IoT) and connect resource-constrained smart objects (mainly sensor nodes) through a lossy network. Smart objects are often heterogeneous in resources and capabilities and have to operate with resource constraints, such as power and memory. Hence, efficient resource management in M2M communications has attracted interest from the research community.

The challenges that we address in this work are:

- Save battery energy in an heterogeneous wireless multi-hop M2M network, which consists of battery-powered nodes and mains-powered nodes.
- Manage efficiently battery node sleep schedules in a distributed manner.
- Route data packets when the majority of nodes is inactive and still satisfy performance guarantees.

4.1.1 Related Work

In battery powered networks, efficient battery energy management is crucial for network operation. Several schemes have been proposed to enable sleep management and duty cycling for different type of networks. In [79], the authors propose the Estimated Duty Cycle (EDC) metric, which takes into account duty cycling in wireless sensor networks. They show that the EDC metric outperforms the standard link metric Expected Transmission Count (ETX) [80] in terms of lifetime and end-to-end delay. A distributed sleep control framework was also proposed in [81]. However, we did not provide any method to realize distributed sleep control. In this work, we introduce new efficient sleep management techniques.

Extending network lifetime and also maintaining acceptable packet delivery rate is one of the major concerns in battery powered networks. The link metrics [82] involve the measurement of a particular quantity for the link between a pair of nodes. The link metrics aim to achieve high throughput but do not indicate energy level of a node. Hence, energy-related routing metrics have been proposed. In [18], the

authors use the residual energy (RE) as a routing metric and design a new objective function for routing decision. They show that the RE metric outperforms the link metric ETX in terms of network lifetime. To identify the energy-bottleneck nodes and to prolong network lifetime, the expected lifetime (ELT) metric is introduced in [19], which outperforms standard RE metric in terms of network stability. In [34], the authors propose a routing metric that aims at maximizing lifetime and takes into account network congestion, energy level and node degree. Simulation results indicate that by using the Optimized Link State Routing Protocol (OLSR), the proposed metric outperforms standard metrics in different environmental conditions.

Finally, energy efficient routing algorithms have been recently proposed. In [83], the authors introduce a central control algorithm that is based on cluster formation to extend the network lifetime. An adaptive routing framework that satisfies certain transmission cost requirements is proposed in [84].

4.1.2 Summary of Contributions

In this work, we propose new distributed sleep management techniques for heterogeneous wireless M2M networks to efficiently manage battery lifetime. Our network is heterogeneous in the sense that it consists of battery-powered nodes (BPNs) and a low percent of mains-powered nodes (MPNs). BPNs schedule variable length sleep intervals that start with an active period (AP), followed by a sleep period (SP). BPNs decide the length of sleep intervals in a distributed manner based on their local data traffic, which consists of buffered packets, self-generated packets,

and prediction of incoming relay data traffic. For this purpose, we introduce a model to estimate the number of incoming relay packets, in which we take into account buffer overflow, channel uncertainty and probability of the BPN being active. In this way, each BPN can predict the amount of incoming traffic and configure its AP appropriately. In addition, we have proposed an AP extension scheme, which gives BPNs flexibility to transmit or receive more data packets. The decision of AP extension depends on the number of buffered packets and the amount of traffic relayed in the last AP or AP extension. Hence, a BPN extends its AP in the case it has plenty of packets to transmit or it is proved to be a reliable relay node.

We also introduce two novel battery energy aware metrics: *Battery Node Energy Waste* (EW) and *Battery Node Relay Cost* (RC). These two metrics are node related metrics and take into account sleep management and different causes of energy depletion. They aim to discover the best route that satisfies some battery related properties. The EW metric indicates the level of BPN's energy consumption due to idle listening and overhearing. Ideally, the routing protocol should reduce the amount of energy waste in order to increase the network lifetime. Hence, selecting a route with minimum energy waste is preferred for energy savings. The RC metric exploits the energy consumption of the BPN due to receiving and transmitting relay data packets. By using this routing metric, the routing protocol intends to balance relay data traffic across the network in a way of prolonging network lifetime.

In addition, we propose some enhancements for the battery energy efficient routing protocol B-RPL [81], which is an evolution of the standard routing protocol RPL used in low-power and lossy networks, such as sensor networks. We introduce

a routing metric advertisement technique and a routing objective function that aims to find the routes to maximize network lifetime.

4.1.3 Chapter Organization

The chapter is organized as follows. Section 4.2 presents the network model that we consider in this chapter. Section 4.3 specifies the distributed sleep management scheme. We describe our novel battery energy aware routing metrics in Section 4.4. Enhancements of B-RPL are presented in Section 4.5. Section 4.6 provides extensive evaluation and analysis of the proposed distributed sleep management and the newly introduced routing metrics based on simulations. We conclude our work in Section 6.5.

4.2 Network Model and Assumptions

A multi-hop wireless machine-to-machine (M2M) network can be modeled as a graph $G(V, E)$. For example, in standard RPL [32], they model a network as the Destination Oriented Directed Acyclic Graph (DODAG). We consider a wireless M2M network that consists of N heterogeneous nodes such that power source is either *power grid* or *battery*. Nodes with *power grid* as power source are called mains-powered nodes (MPNs) and nodes with *battery* as power source are called battery-powered nodes (BPNs). MPNs have unlimited power supply and do not suffer from energy depletion. Hence, MPNs do not sleep. However, BPNs face significant energy depletion. There are several methodologies leading to energy

saving from BPNs such as sleep control and energy aware routing metrics. Both of them are being exploited in our work. Suppose we have M MPNs and B BPNs, where $N = B + M$. Then, the percentage of MPNs in the network is defined as $p_m = M/N$.

We consider multi-point to point (MP2P) dominant traffic pattern with a sink node. The parent of a node is defined as an immediate successor of the node on a path towards the sink node. The child of a node is defined as an immediate predecessor of the node on a path towards the sink node. Suppose we have a H-hop network after topology formation and each layer may consist of router (relay) nodes or leaf nodes or both. Leaf nodes do not have any child node in the topology and only generate packets, but do not relay data packets. Router nodes generate packets and at the same time they relay data packets generated or relayed by their children.

4.3 Distributed battery node sleep management

In this section, we introduce our distributed sleep management techniques to prolong network lifetime. To define network lifetime, we adopt the most widely used definition that is the time until the first BPN runs out of battery energy. Using distributed sleep management, a BPN b sends a wakeup message upon waking up. Once receiving such wakeup message, neighbors can send packets to node b. To reduce the packet collision and save energy, AP is divided into a reception (RX) period and a transmission (TX) period according to the traffic requirements and the relative position of node b in the topology graph. Node b dynamically determines

its RX period length and TX period length based on the number of buffered packets N_b^{BP}, the number of self-generated packets N_b^{SP} and the number of incoming (relay) packets N_b^{RP}. Node b includes its RX period length and TX period length in wakeup message.

4.3.1 Model for incoming data packet arrival rate estimation

A BPN b knows its N_b^{BP} and can compute N_b^{SP} based on its packet generation rate R_b^S. To compute N_b^{RP}, the incoming data packet arrival rate, denoted by R_b^I, needs to be calculated. We therefore propose a model to calculate R_b^I.

To estimate R_b^I, we assume a h-hop node selects (h-1)-hop nodes for its parent set (hierarchical structure). $\mathcal{C}_j^h$, $\mathcal{BRC}_j^h$ and $\mathcal{MRC}_j^h$ denote a h-hop node j's child set, battery-powered router child set and mains-powered router child set, respectively, where $\mathcal{BRC}_j^h \subset \mathcal{C}_j^h$ and $\mathcal{MRC}_j^h \subset \mathcal{C}_j^h$. Moreover, $\mathcal{BP}_i^h$ and $\mathcal{MP}_i^h$ denote a h-hop node i's battery powered parent set and a mains powered parent set, respectively.

We first model the successful packet delivery probability p_{ij}^s from node i to node j. This probability takes into account possible packet drops due to queue overflow and wireless channel conditions. We denote packet drop probability due to queue overflow for node i as p_i^q. This probability is estimated by using the number of packets deleted due to full queue, and the number of packets being pushed into queue measured in previous sleep interval. Queue overflow can lead to significant performance decrease. Let p_{ij}^c be the packet drop rate due to link condition from i to j, which can be approximated using ETX as $p_{ij}^c = 1 - \frac{1}{ETX}$.

Combining the two probabilities, we can estimate the probability for a packet to be successfully delivered from node i to node j as

$$p_{ij}^{s} = \frac{(1 - p_i^q)}{ETX} \tag{4.1}$$

The probability of a MPN m being active is equal to 1 since MPNs do not sleep. For a BPN b, we denote its AP length as T_b^a and SP length T_b^s in a sleep interval. Therefore, the probability of node b being active can be estimated by

$$p_b^a = \frac{T_b^a}{T_b^a + T_b^s} \tag{4.2}$$

Combing equations (4.1) and (4.2), the probability for a h-hop node i to successfully send a packet to a battery powered parent b is given by

$$P2B_{ib}^{h} = \frac{p_b^a * p_{ib}^s}{\sum_{k \in \mathcal{BP}_i^h} p_k^a * p_{ik}^s + \sum_{k \in \mathcal{MP}_i^h} p_{ik}^s} \tag{4.3}$$

and the probability for a h-hop node i to successfully send a packet to a mains powered parent m is given by

$$P2M_{im}^{h} = \frac{p_{im}^s}{\sum_{k \in \mathcal{BP}_i^h} p_k^a * p_{ik}^s + \sum_{k \in \mathcal{MP}_i^h} p_{ik}^s} \tag{4.4}$$

The estimation of incoming data packet arrival rate is conducted using the recursion approach. Let R_i^S be the data packet generation rate of node i. In a H-hop network, H-hop nodes are leaf nodes and therefore, do not relay packets. For a (H-1)-hop BPN b, R_b^I is given by

$$R_b^I = \sum_{i \in \mathcal{C}_b^{H-1}} R_i^S * P2B_{ib}^H \tag{4.5}$$

and for a (H-1)-hop MPN m, R_m^I is given by

$$R_m^I = \sum_{i \in \mathcal{C}_m^{H-1}} R_i^S * P2M_{im}^H \tag{4.6}$$

For h-hop BPN b and h-hop MPN m ($h = 1, 2, ..., H-2$), R_b^I and R_m^I can be estimated recursively by

$$R_b^I = \sum_{i \in \mathcal{C}_b^h} R_i^S * P2B_{ib}^{h+1} + \sum_{i \in \mathcal{BRC}_b^h} R_i^I * R_i^S * P2B_{ib}^{h+1} + \sum_{i \in \mathcal{MRC}_b^h} R_i^I * R_i^S * P2B_{ib}^{h+1} \tag{4.7}$$

$$R_m^I = \sum_{i \in \mathcal{C}_m^h} R_i^S * P2M_{im}^{h+1} + \sum_{i \in \mathcal{BRC}_m^h} R_i^I * R_i^S * P2M_{im}^{h+1} + \sum_{i \in \mathcal{MRC}_m^h} R_i^I * R_i^S * P2M_{im}^{h+1} \tag{4.8}$$

4.3.2 Active Period Length Estimation

For a BPN b, AP length T_b^a equals the sum of the RX period length T_b^{RX} and the TX period length T_b^{TX}. To efficiently estimate node b's T_b^a, our distributed algorithms consider the sleep state of the parent nodes as well as the current and future traffic amount. N_b^{BP} represents the current traffic amount, N_b^{SP} and N_b^{RP} indicate the future traffic amount. Node b has a large TX period if it has more packets to transmit and a large RX period if it needs to relay more packets.

Let Q_b^C be the queue capacity of node b. Estimation of RX period length $T_{b,e}^{RX}$ is given in Alg. 3, which considers buffered packets, self-generated packets and incoming relay packets. Ideally, node b receives until its queue is full and then starts transmitting. However, it may take a long time to fill up the queue if both R_b^S and R_b^I are small. Therefore, a threshold T_{TH}^{RX} is defined.

Estimation of the TX period length $T_{b,e}^{TX}$ is given in Alg. 4. In TX period, node b does not expect incoming relay packets and only considers buffered packets

Algorithm 3 RX Period Length Estimation

1: **if** $N_b^{BP} = Q_b^C$, i.e., queue is full **then**

2: $T_{b,e}^{RX} = 0$, i.e., no RX period

3: **else**

4: $T_{b,e}^{RX} = \frac{Q_b^C - N_b^{BP}}{R_b^S + R_b^I}$, i.e., time to fill up queue

5: **if** $T_{b,e}^{RX} > T_{TH}^{RX}$ **then**

6: $T_{b,e}^{RX} = T_{TH}^{RX}$

7: **end if**

8: **end if**

and self-generated packets. We assume R_b^S is less than the packet transmission rate R_b^T. Otherwise, node b cannot sleep. Ideally, node b transmits until its queue is empty. However, it may take a long time to empty queue if no parent is active. Hence, a threshold T_{TH}^{TX} is defined.

Algorithm 4 TX Period Length Estimation

1: $T_{b,e}^{TX} = \frac{N_b^{BP}}{R_b^T - R_b^S}$, i.e., time to empty queue

2: **if** $T_{b,e}^{TX} > T_{TH}^{TX}$ **then**

3: $T_{b,e}^{TX} = T_{TH}^{TX}$

4: **end if**

We use both estimated lengths $(T_{b,e}^{RX}, T_{b,e}^{TX})$ and measured lengths $(T_{b,p}^{RX}, T_{b,p}^{TX})$ in the previous sleep interval to configure the AP length using the exponential weighted moving average (EWMA) [55] with a weight parameter $0 \leq \alpha \leq 1$ as

$$T_b^a = \alpha * T_{b,p}^{RX} + (1-\alpha) * T_{b,e}^{RX} + \alpha * T_{b,p}^{TX} + (1-\alpha) * T_{b,e}^{TX} \tag{4.9}$$

4.3.3 Sleep Period Length Estimation

In sleep mode, a BPN b does not receive any packets and does not transmit any packets. Only the self-generated data packets are pushed into the queue. Ideally, node b can sleep until its queue is full. However, it may take a long time to fill up the queue if R_b^S is small. Therefore, a threshold T_{TH}^S is defined. The SP length T_b^S computation is provided in Alg. 5.

Algorithm 5 Sleep Period Length Estimation

1: $T_b^S = \frac{Q_b^C - N_b^{BP}}{R_b^S}$, i.e., time to fill up queue

2: **if** $T_b^S > T_{TH}^S$ **then**

3: $\quad T_b^S = T_{TH}^S$

4: **end if**

4.3.4 Active Period Extension

AP extension is a crucial part of the distributed sleep management. It is necessary in cases BPNs need to transmit or receive more packets. At the end of the AP or AP extension, a BPN b determines whether to extend its AP. We define two modes in which the AP is extended. The first mode is the transmission mode described in Alg. 6, which represents the case node b has plenty of buffered packets, i.e., N_b^{BP} is greater than a queue threshold Q_{TH}. The second mode is the reception mode described in the Alg. 7, which indicates that node b is either a reliable relay, i.e., it has received packets and has forwarded all of them, or has not received packets for a long period of time and therefore has more battery energy. In Alg. 6

and 7, T_b^{RE} is the latest RX period ending time among all active parents, T_b^{EQ} is the time needed for node b to transmit all buffered packets, T_b^{TE} is the latest TX period ending time among all active parents, T_b^{MT} is the time needed for node b to transmit the maximum number of allowed packets in parent's TX period (N_{PT}^{MA}), N_b^R is the number of packets received, and E_{TH} is the energy threshold.

Algorithm 6 Active Period (AP) Extension (Transmission Mode)

1: // **Transmission Mode** ($N_b^{BP} > Q_{TH}$)
2: **if** ∃Parent in RX period **then**
3: Extend AP by $\min\{T_b^{RE}, T_b^{EQ}\}$
4: **else if** ∃Parent in TX period **then**
5: Extend AP by $\min\{T_b^{TE}, T_b^{MT}\}$
6: **else if** ∄Parent active **then**
7: Pick probability $p_1 = C_1 * N_b^{BP}/Q_b^C$
8: Extend AP by N_b^{BP}/Q_b^C second with probability p_1
9: **if** Energy of node b is greater than E_{TH} **then**
10: Send AP extension notification message
11: **end if**
12: **end if**

Our AP extension is a combination of deterministic and probabilistic approaches. In transmission mode, if node b has any parent in RX period it attempts to transmit all buffered packets, and if node b has any parent in TX period it attempts to transmit N_{PT}^{MA} packets. Probabilistic extension depends on probabilities p_1, p_2 and p_3. The p_1 is related to queue utilization and it needs to be greater for

bigger queue utilization. The p_2 and p_3 are related to hop count of node b and are greater as node b approaches the sink node. Nodes having large hop count should not extend their AP with high probability if they are in reception mode. We choose appropriately the constants C_1, C_2 and C_3 based on our objective. We want to extend AP with relatively high probability if node b is in transmission mode and therefore C_1 has a large value. On the other hand, a node should extend its AP in reception mode only if it is proved to be a reliable relay, or if it has not received packets for several consecutive active periods or AP extensions. Hence, C_2 and C_3 are chosen to have smaller values than C_1. In simulation, we set $C_1 = 0.8$, $C_2 = 0.2$, $C_3 = 0.1$, $n_1 = 5$, $n_2 = 20$ and $T^a_{min} = 0.1$ sec (chosen parameters in our AP extension algorithm).

Alg. 6 and 7 show that node b only announces AP extension in the reception mode but not in the transmission mode. The number of times a BPN can extend its AP varies among different objectives. For higher packet delivery rate, a BPN can extend its AP as many times as needed. This process is called *PDR-oriented* extension. To achieve longer network lifetime, a BPN can extend its AP up to a threshold of consecutive times (5 in simulation). This process is called *Lifetime-oriented* extension.

4.3.5 Transmission (TX) Control in Active Period

Using distributed sleep management, all nodes must monitor and maintain parents' RX and TX schedules. MPNs are considered to be in RX period all the

time. BPNs announce their RX and TX schedules in the wakeup and AP extension messages. A node (BPN or MPN) should send packets to parents, which are in RX periods and it should select the best parent to send packets. A node sends packets to a parent during that parent's TX period only if its queue overflows and no parent is in RX period. Due to distributed sleep schedules, a node may miss parent's wakeup message or AP extension message. In the case of queue overflow, even if a node does not receive RX and TX schedules from a parent, it can still send a packet to that parent if (i) the node overhears transmission from that parent or (ii) the node overhears a neighbor's transmission to that parent.

4.4 Battery energy aware routing metrics for heterogeneous networks

Different metrics are being used to exploit diverse characteristics of the network and to provide QoS guarantees. Routing metrics can be characterized as node and link related metrics. Link metrics [82] involve the measurement of a particular quantity for the link between a pair of nodes. Typical link metrics are Expected Transmission Count (ETX) introduced in [80] that estimates the expected number of transmissions (including retransmissions) required to send a unicast packet over a link, Link Bandwidth (BW) introduced in [85] that measures link bandwidth using the Packet Pair method and Expected Transmission Time (ETT) metric that combines both ETX and BW metrics. On the other hand, node metrics indicate current node status based on the different routing objectives. Node status includes typical resources, such as CPU usage, available energy, remaining residual energy

or some variation of the above. Our objective is to propose metrics which take into account energy waste and distributed sleep management model. Hence, in our work we focus on node related routing metrics that provide some information related to sleep or energy consumption. We propose two novel routing metrics *Battery Node Energy Waste (EW)* and *Battery Node Relay Cost (RC)*, which can be used under various objective functions, and in the performance evaluation section we compare them with standard RE metric. In addition, we introduce a distributed measurement method for ETX metric related to the upward traffic, which is dominant for machine-to-machine communications that use RPL protocol. Both battery-aware routing metrics require an estimation of ETX metric to compute some underlying related quantities.

4.4.1 Battery Node Energy Waste (EW)

Battery node energy waste (EW) measures energy consumption from operations not related with transmission and reception of control and data packets. The mains causes of energy waste for battery powered nodes in heterogeneous machine-to-machine networks are idle listening and overhearing. Idle listening indicates how efficient is our proposed distributed sleep management model, because ideally battery powered nodes should consume as little energy from idle listening as possible and be active only during time periods that have to transmit or receive control or data packets. EW is sleep-aware metric and is used to forward traffic through more sleep effective intermediate nodes, which have consumed less energy to idle listening

and overhearing.

EW is an additive metric and can be used to define a path cost by exploiting different objective functions (OF). For MPNs, we define $EW = 0$. For a BPN b, computation of EW metric involves energy consumption on packet transmission (E_{TX}) and energy consumption on packet reception (E_{RX}), which are given by

$$E_{TX} = N_T * ETX * P_T * \frac{L_p}{D_R} \tag{4.10}$$

$$E_{RX} = N_R * P_R * \frac{L_p}{D_R} \tag{4.11}$$

where N_T is the number of packets transmitted by b, N_R is the number of packets received by b, P_T is the transmission power, P_R is the reception power, L_P is the average packet length and D_R is the PHY data rate. Let E_0 be the initial energy and RE_t be the residual energy at time t. The total energy consumption is given by $E_c = E_0 - RE_t$. Hence, EW at time t is computed as

$$EW = E_c - (E_{TX} + E_{RX}) \tag{4.12}$$

4.4.2 Battery Node Relay Cost (RC)

Battery node relay cost (RC) measures energy consumption related to transmission and reception of relayed traffic. By relayed data traffic, we define data packets that are not originated by this node, but are forwarded to the next hop towards the destination node. RC metric distributes relayed data traffic across intermediate nodes, because our objective function selects paths with small RC.

RC is also an additive metric. RC metric balances relay packets across intermediate nodes with an objective function of selecting paths with small RC. For MPNs,

we define $RC = 0$. For a BPN b, computation of the RC metric involves the energy consumption on relay packet transmission (ER_{TX}) and the energy consumption on relay packet reception (ER_{RX}), which are given by

$$ER_{TX} = NR_T * ETX * P_T * \frac{L_p}{D_R} \tag{4.13}$$

$$ER_{Rx} = NR_R * P_R * \frac{L_p}{D_R} \tag{4.14}$$

where NR_T is the number of relay packets transmitted by b and NR_R is the number of relay packets received by b. At time t, RC metric is computed as

$$RC = ER_{TX} + ER_{RX} \tag{4.15}$$

4.4.3 Distributed ETX Measurement

In this section, we briefly describe the distributed ETX measurement process. We take into account only the upward traffic for the MP2P traffic pattern. A child node maintains a sequence number for each parent. A parent node uses the sequence number to determine the upward ETX for each child. In order to send updated ETX to children, parents use different approaches depending on whether they are battery-powered or mains-powered. Battery-powered parents include a (Child, ETX) list in wakeup messages so that once a child receives the wakeup message, it also obtains updated ETX metric. On the other hand, the mains-powered nodes broadcast (Child, ETX) lists periodically so that the mains-powered children can receive the ETX update and unicast the ETX update to a battery-powered child once they receive a wakeup message from the child. By using this approach, every child obtains

updated ETX for each parent and can use it to compute routing metrics, such as EW and RC, or select routes based on ETX.

4.5 Battery metric aware B-RPL routing protocol

Distributed sleep management and battery energy aware metrics are used to enhance the B-RPL protocol [81]. The B-RPL protocol is an enhanced version of standard RPL [32] (used widely for routing in sensor networks) to support distributed sleep control in networks consisting of heterogeneous nodes. In this section, we present enhancements that we have proposed for B-RPL.

4.5.1 Metric Advertisement

Metric advertisement and updates are crucial in order to select the best route among all the candidate routes. In our protocol, routing metrics, e.g., EW, RC, RE and ETX, are not only included in the DODAG Information Object (DIO) messages but also in the wakeup messages. In this way, all the children have updated metrics regarding the candidate routes every time that they receive a wakeup message from a parent. This approach is lightweight and does not significantly increase the overhead since BPNs have to transmit wakeup messages.

4.5.2 Objective Function (OF) for Route Selection

Objective Function (OF) specifies how routing protocols select routes based on specific metrics. For example, for hop count metric and the shortest path OF,

a node will select the routes with minimum hop count to the sink node. For each metric, different objective functions can be defined.

Our objective is to maximize the network lifetime and maintain an acceptable packet delivery rate. Hence, we identify some OF candidates that satisfy our goal. To maximize the network lifetime, we focus on reducing the maximum energy consumption among BPNs and not the total energy consumption. Therefore, we choose an OF that selects path p from a set of candidate paths P as

$$p \in argmin_{p \in P}[max_{i \in p}(RoutingMetric(i))] \tag{4.16}$$

where i is a node on path p and $RoutingMetric(i)$ is the value of routing metric at node i. In case of EW, this OF tries to find a path that wastes the least battery energy and in case of RC, this OF tries to find a path that relays the least packets. Thus, in both cases, this OF tries to minimize the maximum energy consumption of a node in the network and maximize the lifetime.

In the case of the standard RE metric, we are trying to find the path that has the maximum residual energy. Therefore, we use the OF described as

$$p \in argmax_{p \in P}[min_{i \in p}(RE(i))] \tag{4.17}$$

4.6 Performance Evaluation and Analysis

In this section, we present an extensive performance evaluation of our distributed sleep management scheme and battery energy aware routing metrics using the NS2 simulator [86]. The IEEE 802.15.4 MAC and PHY are modified to support

the sleep operation. We simulated a heterogeneous wireless M2M network containing 500 nodes deployed in a 23×23 grid. Each unit in the grid represents a 10×10 square meter real-world field and contains a single node. Each node is randomly placed in its corresponding square field. A data sink is placed at the center of the grid. Our M2M network consists of BPNs and MPNs. All BPNs have 100% of battery level at the beginning of the simulation. We examine low percentage of MPNs that varies from 1% to 20%, but in this dissertation we present results only for the scenario that we have 5% MPNs in the network.

In the periodic sleep (PS) scenario, each BPN has a fixed sleep interval with fixed AP length and SP length and wakes up in a periodic pattern. In this case, there is no AP and SP configurations and no AP extension as described in Section 4.3. In simulation, the sleep interval for all BPNs is 60 seconds with 0.05 seconds of AP and 59.95 seconds of SP. We use PS scheme as the benchmark for comparing performance of our distributed sleep (DS) scheme.

In our simulation, we have also introduced the structured mains-powered node placement (SMNP) and we compare its performance with random mains-powered node placement (RMNP). In SMNP case, we place selected MPNs in specific grid units. For example, the selected MPNs are placed in horizontal, vertical or diagonal lines from the center of the grid that the sink node is placed.

The enhanced B-RPL is used as routing protocol. EW, RC and RE metrics are used in simulation. We simulated the data collection scenario in which each node (BPN or MPN) generates 1 packet per minute. The queue capacity of each node is 15 and our simulation runs for 15000 seconds. We use several performance

metrics for our evaluation but the main ones are packet delivery rate (PDR), network lifetime and idle time percentage of BPNs.

4.6.1 Distributed Sleep Management and Structured Mains-powered Node Placement Performance Evaluation

In this section, we present the performance evaluation of the DS scheme and the SMNP using newly introduced routing metrics. We compare the performance of the DS scheme with the performance of the PS scheme under different scenarios. Following figures show simulation results for 5% of MPNs.

Fig. 4.1 compares the PDR for different scenarios with EW metric. The PS with RMNP has only 10% of PDR. However, the DS with RMNP obtains 33% of PDR. Therefore, the DS improves PDR by 23%. The PS with SMNP obtains 46% of PDR. On the other hand, the DS with SMNP achieves 78% of PDR. In this case, the DS improves PDR by 32%. With the PS, SMNP improves PDR by 36% and with the DS, SMNP improves PDR by 45%. These results indicate the importance of the MPN placement and they also show that the DS utilizes MPNs better than the PS does. Fig. 4.2 shows the minimum battery node energy level, which indicates the network lifetime. The PS provides longer network lifetime in expense of PDR, but the DS still maintains comparable network lifetime.

Fig. 4.3 and Fig. 4.4 show results using the RC metric. It can be seen that the DS and SMNP achieve similar results as those using the EW metric. The DS achieves much higher PDR and comparable network lifetime compared with the PS.

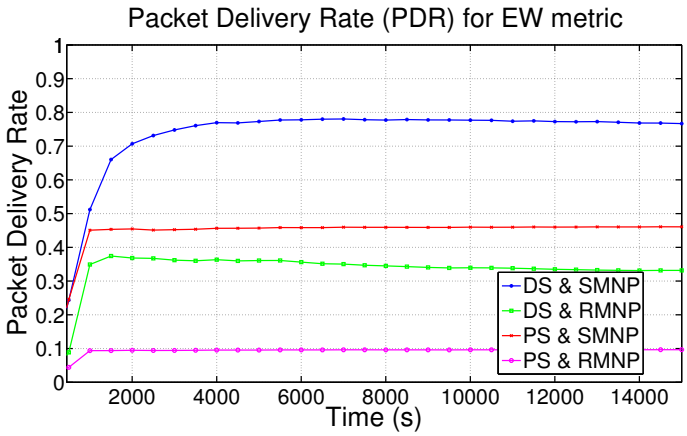


Figure 4.1: PDR Per Sleep Scheme

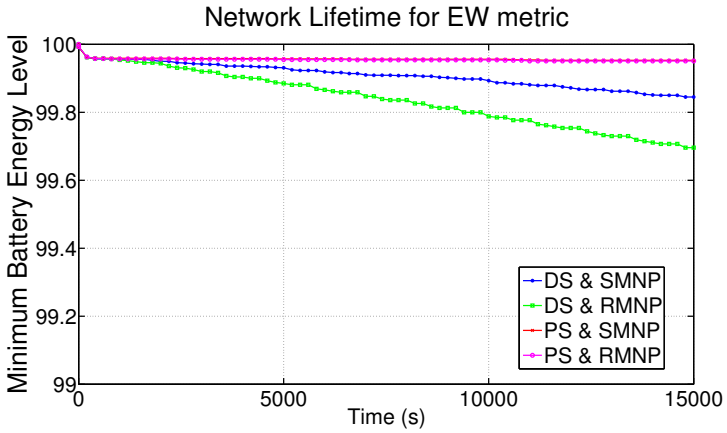


Figure 4.2: Lifetime Per Sleep Scheme

4.6.2 Routing Metrics Performance Comparison

In this section, we present the performance comparison for our newly introduced battery energy aware routing metrics EW and RC described in Section 4.4. We use the standard RE metric for benchmark comparison. The combined DS and SMNP scenario is used since it is the optimized case. Following figures show simulation results for 5% of MPNs.

Fig. 4.5 compares the PDR across all the routing metrics in the optimized

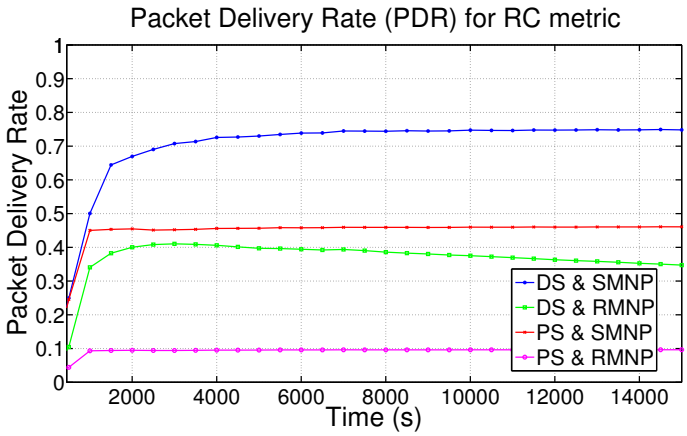


Figure 4.3: PDR Per Sleep Scheme

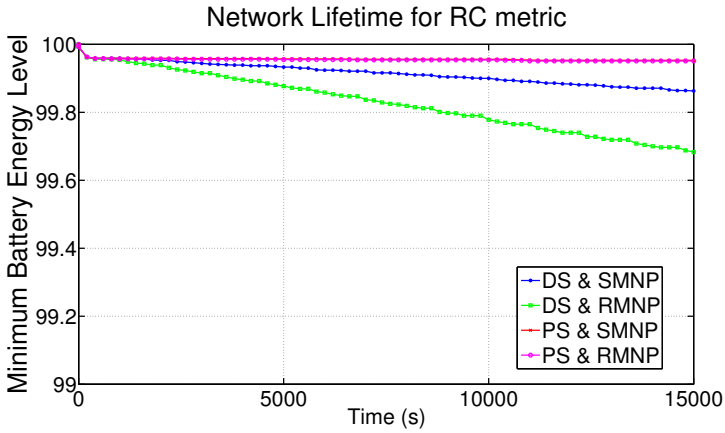


Figure 4.4: Lifetime Per Sleep Scheme

scenario. The EW and RC metrics have improved PDR by almost 10% compared with the standard RE metric. The EW achieves the highest PDR. Fig. 4.6 shows that the EW and RC metrics also perform better in terms of network lifetime than RE metric does. However, the RC metric achieves the best performance in terms of network lifetime. Fig. 4.5 and Fig. 4.6 indicate the efficiency of the proposed battery energy aware routing metrics.

Fig. 4.7 shows the idle time percentage of the BPNs. Across all routing metrics the idle time percentage is low for all BPNs, which indicates that the proposed

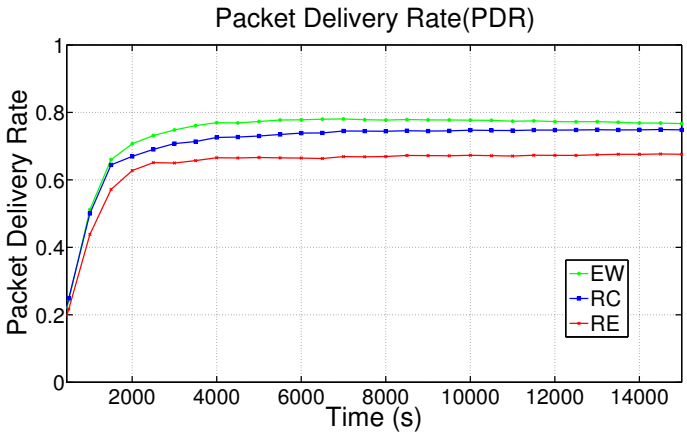


Figure 4.5: PDR Per Routing Metric

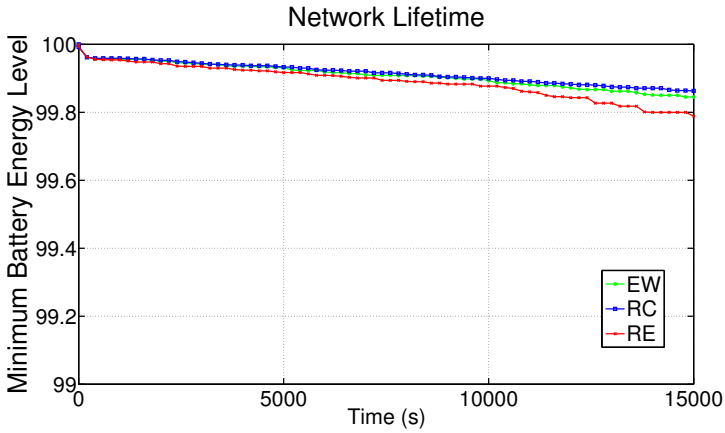


Figure 4.6: Lifetime Per Routing Metric

distributed sleep management scheme is very efficient so that BPNs are active only if they need to ensure reliability.

Finally, we also have results showing that the total energy consumption for all BPNs is lower using the EW and RC metrics in comparison with the standard RE metric.

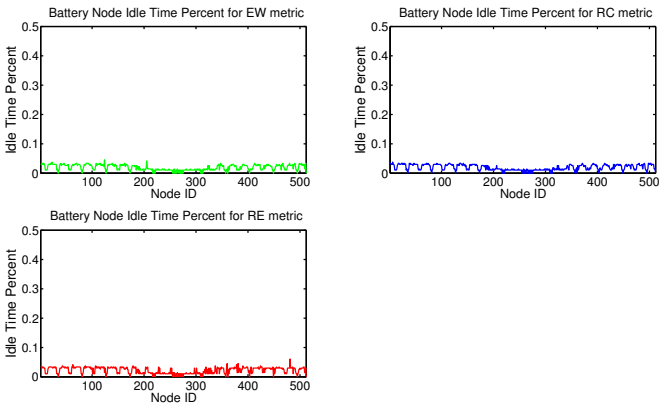


Figure 4.7: Battery Node Idle Time Percentage Per Routing Metric

4.7 Concluding Remarks

We propose new distributed sleep management techniques for battery powered nodes in heterogeneous wireless machine-to-machine networks. The distributed sleep management techniques enable each battery-powered node to self-determine the optimal times to sleep or wake up and process data based on traffic measurements and predictions. We also propose two novel battery energy aware routing metrics, battery node energy waste (EW) and battery node relay cost (RC), that take into account the main causes of energy depletion. Our distributed sleep management can improve packet delivery rate (PDR) by 32% compared with periodic sleep scenario while maintaining comparable network lifetime. Our EW and RC metrics outperform the standard residual energy (RE) metric in terms of PDR, network lifetime and idle time percentage. As part of future work, we plan to investigate the combination of energy aware routing metrics based on the routing objective and a formal optimization approach for defining the placement of mains powered nodes.

Algorithm 7 Active Period (AP) Extension (Reception Mode)

1: // **Reception Mode** ($N_b^{BP} \leq Q_{TH}$)
2: **if** Node b has received and forwarded data packets during last AP or AP extension **then**
3: Pick probability $p_2 = C_2/hop\_count$ with $C_1 > C_2$
4: **if** Node b is in this mode for n_1 consecutive times **then**
5: **if** Energy of node b is greater than E_{TH} **then**
6: Extend AP by N_b^R/Q_b^C second with probability p_2
7: Send AP extension notification message
8: **end if**
9: **end if**
10: **else if** Node b has not received data packets during last AP or AP extension **then**
11: Pick probability $p_3 = C_3/hop\_count$ with $C_2 > C_3$
12: **if** Node b is in this mode for n_2 consecutive times **then**
13: **if** Energy of node b is greater than E_{TH} **then**
14: Extend AP by T_{min}^a with probability p_3
15: Send AP extension notification message
16: **end if**
17: **end if**
18: **end if**

CHAPTER 5

Multi-Metric Energy Efficient Routing Scheme for Mobile Ad-hoc Networks

In the previous chapter, we introduced a new distributed sleep management scheme and novel routing metrics for heterogeneous machine-to-machine networks and we integrated and examined their performance in a standard routing protocol for this type of networks. In this chapter, we propose a novel unified routing metric for energy-aware routing in dynamic networks, such as mobile ad-hoc networks, which is also integrated to a standard routing protocol.

5.1 Overview

Mobile communication systems without central management have been gaining popularity in the form of multi-hop ad-hoc networks. Multi-hop ad-hoc networks have widespread applications ranging from military scenarios, handling emergencies and natural disasters to distributed processing of data. This type of mobile networks do not rely on any pre-established infrastructure and all nodes should be able to communicate with other nodes directly or indirectly through intermediate

nodes. Hence, routing is a key operation in this type of networks and the appropriate selection of routing scheme affects its performance. For this reason designing efficient routing schemes has been an extensive research area in the last decade. Mobile ad-hoc networks (MANET) is one form of wireless multi-hop ad-hoc networks, where network topology changes dynamically. Many proactive (e.g. OLSR [33]) and reactive routing protocols (e.g. AODV [38]) for MANET have been developed.

Limited battery life of mobile nodes impose an important limitation in the performance of this type of networks. Power depletion of a mobile node affects its ability to forward packets on behalf of other nodes and may lead to partitioning of the network. Therefore, efficient battery utilization and increase of network lifetime should be important design criteria for developing routing schemes in MANET.

While many energy efficient routing protocols have been presented [87], [88], [89] to optimize energy consumption across the network, these protocols are mostly based on a single routing metric, which is derived from energy measurements. For example, some of the current techniques use the reciprocal value of the residual energy to do minimum energy consumption routing [20], [21] or they use Minimum Drain Rate (MDR) mechanism [20], [22]. These approaches, based on a single metric, are myopic and do not consider all the possible causes of energy depletion in the network. Thus, the investigation of a combination of network parameters (multiple metrics), which are not strictly derived by residual energy but still indicate energy depletion levels in parts of the network, will contribute to designing more efficient routing schemes.

Our insight for this work is to determine a wider set of cross-layer parameters,

not solely based on energy measurements, that enable effective prediction of low energy paths, while encouraging uniform utilization of network resources. First, we identify the reasons that cause energy depletion in different parts of the network and then choose our metrics to mitigate their effect. We take into consideration three metrics in our routing scheme: *MAC queue utilization*, *residual energy* and *node degree*. *MAC queue utilization* is introduced to help our routing scheme to predict and avoid congested parts of the network, which are subject to high energy consumption due to the number of packet transmissions. *Residual energy* is crucial, because we want to choose paths, which include less depleted nodes. Finally, *node degree* contributes to reducing the energy consumption due to overhearing in the neighbor nodes of a possible intermediate node of the selected path.

5.1.1 Related Work

The problem of designing energy efficient routing protocols has received significant attention by the research community for over a decade. Many energy efficient routing schemes have been developed [87], [88], [89], which are typically based on residual energy derived metrics. In particular, a lot of research has been conducted on modifying standard routing protocols in MANET, such as AODV [89], [90], DSR [22] and OLSR [20], [21], [91], [92] and [93].

Many energy efficient variations of OLSR modify both the MPR selection and the route computation algorithm. For the MPR selection, some protocols [21] choose the 1-hop neighbors with the maximum residual energy, while some others

modify the willingness metric of the 1-hop neighbors [20], [91] based on the energy level. In addition, an algorithm that takes into account the residual energy of the 1-hop and 2-hop neighbors of the MPR candidate is presented in [92]. To make the routing decision, most of the techniques proposed above modify the routing metrics to take into account energy consumption. Commonly used metrics for computing path costs, such as reciprocal value of residual energy [20]. [21] and drain rate in intermediate nodes [20], [22], are only based on energy measurements.

Multiple metrics routing schemes, which are more related to our work, have also been proposed. An adaptive multiple metrics routing scheme for AODV is introduced in [90]. The authors take into account three routing metrics, which are hop count, traffic load and energy cost, and combine them to evaluate the cost of the paths. In addition, a predictive multiple metrics routing scheme for proactive protocols, like OLSR, is presented in [93]. The chosen routing metrics are mean queueing delay, energy cost and residual link lifetime. The authors designed a multi-objective routing schemes that evaluates the multiple metrics in a composite way and achieves better performance in terms of Packet Delivery Ratio (PDR) and network lifetime.

5.1.2 Summary of Contributions

In this chapter of the dissertation, we focus on designing a novel multiple metric routing scheme to be applied in dynamic wireless multi-hop networks, such as mobile ad-hoc networks (MANET), based on the described metrics, and integrate

it to standard OLSR to examine its effectiveness. We combine the multiple metrics to compute a weight for each node. These weights are efficiently propagated to the rest of the network. Then, we propose a weight-based routing scheme, which will use a greedy approach to choose the path with the lower cost, computed from the weights the nodes received. In this way, the routing scheme takes into account the multiple metrics introduced and switches paths in order to avoid energy-depleted, highly congested and dense areas of the network. The contributions of this chapter are:

- Introduction of a combination of routing metrics from multiple layers, which has not been examined in any previous work related to energy efficient routing
- Proposal of a novel multi-metric routing scheme, which takes into account the above metrics, and integrate it to standard OLSR
- Experimentation with the modified OLSR using NS3 [94] and examination of its energy behavior and its performance (in terms of Network Lifetime and PDR) compared to the standard OLSR

5.1.3 Chapter Organization

The rest of the chapter is organized as follows. We describe our multi-metric routing scheme in Section 5.2. We verify our claims by extensive simulations in Section 5.3.

5.2 Multi-Metric Energy Efficient Routing Scheme

The main goal of our routing scheme is to increase the network lifetime, without loss of performance, and we use OLSR as a case study. For the purpose of this work, lifetime is defined as the time until the battery of *any* mobile node of our ad-hoc network depletes. We adopt this definition because, in the worst case, the depletion of a node may possibly cause network partition. To prevent the energy depletion and increase the network lifetime, we need to take into account cross layer parameters. These parameters include network congestion, residual energy of mobile nodes, as well as, network topology parameters. We aim to modify OLSR to make routing decisions according to these parameters and measure the performance improvement of our approach compared with the standard OLSR, using various performance metrics.

In this section, we introduce the routing metrics for our scheme and describe the modifications that need to be done in the OLSR protocol. We present the metrics propagation mechanism in detail and a greedy approach to make routing decisions based on the current values of the metrics.

5.2.1 Routing Metrics

We are mainly interested in defining the network parameters that affect the network lifetime, which will be our main performance metric. An intuitive way to select an appropriate set of metrics for energy efficient routing is to first investigate the causes of energy depletion. The residual energy in mobile nodes is being depleted

in two ways [20]:

1. **Packet Transmission:** each transmission causes energy consumption at the mobile node

2. **Overhearing from the neighbor nodes:** Due to the broadcast nature of the wireless channel, all the nodes in the neighborhood of a sender node may overhear its packets transmission, even if they are not the receivers. Reception of these packets results to unnecessary expenditure of battery energy of the recipients.

The proposed routing scheme takes into account three routing metrics to estimate the path cost and make the routing decision:

1. **MAC queue utilization**: This parameter indicates network congestion. When a mobile node has to transmit a lot of packets then this will lead to a significant energy consumption. Thus, larger weight should be assigned to nodes with high MAC queue utilization.

2. **Residual energy**: This parameter is crucial in order to determine the next-hop node. The traffic should be directed to nodes that have enough residual energy to transmit. Hence, we should assign a large weight to nodes that have small residual energy to do forwarding.

3. **Node Degree**: The degree of a node is the number of nodes that belong to its one-hop neighborhood. As we mentioned before, one reason for energy depletion is overhearing. We will try to avoid forwarding packets through

nodes with high degree, because this will cause greater overall energy depletion. In addition, lower degree nodes also reduce the size of the interference graph, so fewer collisions will happen during our packet transmissions.

We use a weight-based routing scheme, where a weight is assigned dynamically at each node. Mobile nodes update their routing tables according to the path costs computed using the nodes' weights received at each time period. The metrics are normalized by their maximum values and they contribute additively to the node's weight computation with some multiplicative factors, as shown in Equation 5.1.

$$w_i = \alpha_1 \frac{L_i}{L_{max}} + \alpha_2 (1 - \frac{E_i}{E_{max}}) + \alpha_3 \frac{D_i}{D_{max}}, \tag{5.1}$$

where $\sum_{i=1}^{3} \alpha_i = 1$, L_i is the number of packets in the MAC queue, E_i is the residual energy at each time and D_i is the node degree. L_{max} is the maximum considered MAC queue size, E_{max} is the initial energy of a node and D_{max} is the number of nodes in the network minus one. By varying the weighting factors, we can change the importance of the three routing metrics during route discovery. For our experiments, we gave equal value to all α_i.

5.2.2 Modifications in the standard OLSR

5.2.2.1 **TC packet format**

The proactive nature of OLSR indicates that nodes need to periodically receive other nodes weights, in order to compute the path costs and update their routing tables. Thus, it is crucial to find a way to propagate nodes weights to the network

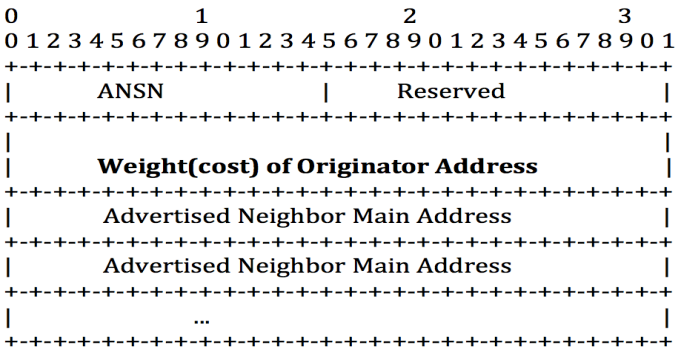


Figure 5.1: TC packet format

without increasing network overhead. An effective way is to embed this information to the TC packets that are periodically generated by each node. This does not introduce significant overhead, in contrast to new packets being generated to advertise the weights, and is easily implementable in routing protocols. Therefore, we extend the TC packet to include a field for the updated weight (computed locally using equation (1)) of the originator node, as shown in Figure 5.1. Nodes receive and process the new TC packets and create topology tuples in their topology table. Topology tuples include the originator node of TC, the destination node, which is one of the nodes advertised in the TC packet, and also the weight of the originator node.

5.2.2.2 MPR selection

We have not modified the way that OLSR selects MPRs in our scheme. It is based on the heuristic that the MPR set should cover all the nodes that are two hops away (two-hop neighborhood).

5.2.2.3 Routing Table Computation

OLSR is based on minimum hop routing to proactively determine the next-hop for a specific destination node. Thus, we need to change the way the routing table is computed using OLSR. The update of routing tables should be based on the path costs computed from the nodes weights and not on the number of hops. In addition, routing tables should include path costs to the destination address instead of the number of hops, as in standard OLSR. We define path cost as the sum of weights of the intermediate nodes along the path plus a constant weight, equal to 1, that corresponds to the source node. We do not consider the receiver's weight, because the energy consumption during transmission of a packet from an intermediate node is much greater than the power consumption of the receiver. To periodically update the routing table based on the computed path costs, we use the greedy heuristic scheme outlined in Algorithm 8.

The algorithm assigns cost equal to 1 to paths towards the 1-hop neighbors. Then, it examines the topology tuples stored in the topology table and three cases are considered in order to update the routing table. The first one occurs when there is an entry in the routing table for the originator node (last_addr) of the topology tuple. In this case, we add a new entry to the routing table for the destination node of TC (dest_addr) with cost equal to the sum of the cost corresponding to route to the originator node (last_addr.cost) and the originator node's weight. The second case, where the greedy nature of our algorithm applies, occurs when there are entries for both the originator and the destination node of the topology tuple. Then, we

Algorithm 8 Greedy Heuristic Algorithm

1: Clear Routing Table ()
2: **for** $i \leftarrow 1, NumberofNeighbors$ **do**
3: Add New Entry in the Routing Table
4: $cost \leftarrow 1$
5: **end for**
6: **for** $i \leftarrow 1, NumberofTopologyTuples$ **do**
7: $have\_last \leftarrow Lookup(RouteTable, last\_addr)$
8: $have\_dest \leftarrow Lookup(RouteTable, dest\_addr)$
9: **if** $have\_last \leftarrow true \wedge have\_dest \leftarrow false$ **then**
10: Add New Entry in the Routing Table:
11: $dest \leftarrow dest\_addr$
12: $next\_addr \leftarrow last\_addr.next\_address$
13: $interface \leftarrow last\_addr.interface$
14: $cost \leftarrow last\_addr.cost + tc\_weight$
15: **else if** $have\_last \leftarrow true \wedge have\_dest \leftarrow true$ **then**
16: UPDATE_ROUTING_ENTRY(last_addr, dest_addr, tc_cost) {If Needed}
17: **else**
18: Not adding Routing Table Entry
19: **end if**
20: **end for**

choose greedily the new path detected through the originator node or we maintain the old path, by comparing their costs. Finally, in the case we do not have entries neither for the originator node nor the destination node, we do not create any new entry.

An interesting result observed using the proposed modified routing scheme is that the nodes tend to choose longer paths, when the intermediate nodes of the currently selected paths have consumed a lot of energy. The observed path switching indicates that our routing scheme disperses the traffic across the network and utilizes the network resources in a more uniform way to increase network lifetime.

5.3 Performance evaluation

In this section, we will describe the different performance metrics used to evaluate our scheme, the simulation setup and the simulation results. For our simulations, we used the NS3 network simulator [94] and we created a modified version of OLSR to execute our experiments.

5.3.1 Performance Metrics

The performance metrics should be selected appropriately, such that we are able to show the efficiency of our scheme in comparison with the standard routing protocol. The metrics that we consider for the evaluation of our modified scheme are:

1. **Packet Delivery Ratio (PDR) (in %)**: The ratio of the number of packets

delivered to the destination nodes over the number of packets sent by the source nodes. We use the normalized PDR, which is defined as the number of packets delivered divided by the number of packets that should ideally been transmitted in this data rate. This gives more representative results.

2. **Network Lifetime (in sec)**: The time until the battery of a mobile node of our mobile ad-hoc network depletes. There are several different definitions for network lifetime introduced in the literature, but we assume that the depletion of a mobile node may lead to network partition, due to the dynamically changing network topology in this type of networks.

3. **Average Node Residual Energy vs Time**: This metric was first proposed in [91]. It shows how the average energy consumption (total residual energy[J]/number of nodes) changes over time.

4. **Distribution of node residual energy**: This metric [92] indicates how the energy is consumed across the network. It shows how many nodes have the same percentage of residual energy at the end of the simulation. Ideally we should have a big number of nodes that will have 30-70 % of their residual energy at the end of the simulation. This would illustrate that our modified scheme disperses the traffic across different paths in the mobile network and utilizes more uniformly the network resources.

Table 5.1: Simulation parameters

Area	2000m x 2000m
Nodes	30
Traffic Sources	3
Traffic Type	CBR/UDP
Packet Size	512 bytes
Start of Traffic	30 sec
Initial Node Energy	7 Joules
Transmission Power	5 dbm
Simulations/Scenario	3
Link bandwidth	1 Mbps

5.3.2 Simulation Setup

We simulated a MANET with 30 nodes in a dense 2000 x 2000 meter square area, which has 4 hop network diameter. There are 3 CBR/UDP sources generating packets of 512 bytes with different data rates. To compute the average value of the metrics we want to derive, we simulated each scenario 3 times. In each different simulation, we choose different source-destination pairs. We have two variations of the simulation setup to evaluate the performance of our modified routing scheme compared to the standard OLSR. The common simulation parameters of the two variations are summarized in Table 5.1.

5.3.2.1 Setup A

In this setup, we want to evaluate the modified OLSR using the performance metrics that will give us clear view of the performance in terms of energy consumption. These metrics introduced in the previous section are: Average residual energy vs time and Distribution of node residual energy. For this setup we consider a static scenario with the parameters described in Table I. In addition, the packet interarrival time is 0.1sec, which means that the source nodes send 10 packets/sec (around 41 Kbps). The simulation time is set to 250 sec.

5.3.2.2 Setup B

In this setup, we want to examine the performance of our modified OLSR compared to standard OLSR in a variety of static and mobile scenarios. These scenarios have the common parameters described in Table I. The two performance metrics that we will consider are Packet Delivery Ratio (PDR) and Network Lifetime (in sec). The simulations are done with 4 different packet interarrival times to study the effect of low, medium and high traffic rate in our scheme. The intervals are 0.1 sec (10 packets/sec or around 41 Kbps), 0.075 sec (14 packets/sec or around 55 Kbps), 0.05 sec (20 packets/sec or around 81 Kbps) and 0.025 sec (40 packets/sec or around 163 Kbps). At shorter intervals, a lot of packets will be lost due to network congestion and the bandwidth limitations, that we have set up for our experiments in NS3. Thus, we will notice a significant decrease in PDR as the interval becomes smaller. We also consider three different scenarios in terms of mobility: static, low

mobility and high mobility. In the low mobility scenario, mobile nodes move in the area based on a Random Waypoint mobility model with maximum speed of 2 m/sec. In the high mobility scenario, nodes move based on a Random Waypoint mobility model with maximum speed of 20 m/sec. Finally, the simulation time is set to 150 sec for taking, under different data rates, comparative measurements for PDR. For network lifetime measurements, we execute the simulations until a node is completely depleted.

5.3.3 Simulation Results

In the following subsections, the results of our simulations in NS3 for Setup A and Setup B are presented.

5.3.3.1 **Simulation Setup A**

In this setup, we consider a static scenario with data rate of 10 packets/sec for our traffic flows. The purpose is to perform an energy analysis of the behavior of our modified scheme in comparison with standard OLSR. Hence, we examine the two performance metrics described in the section 5.3.1.: Average Node Residual Energy vs Time and Distribution of node residual energy.

For the first performance metric considered, we present the results in Figure 5.2. The observation is that our modified scheme leads to a lower average power consumption than standard OLSR. We notice that there are 10-15% energy savings by using the modified OLSR.

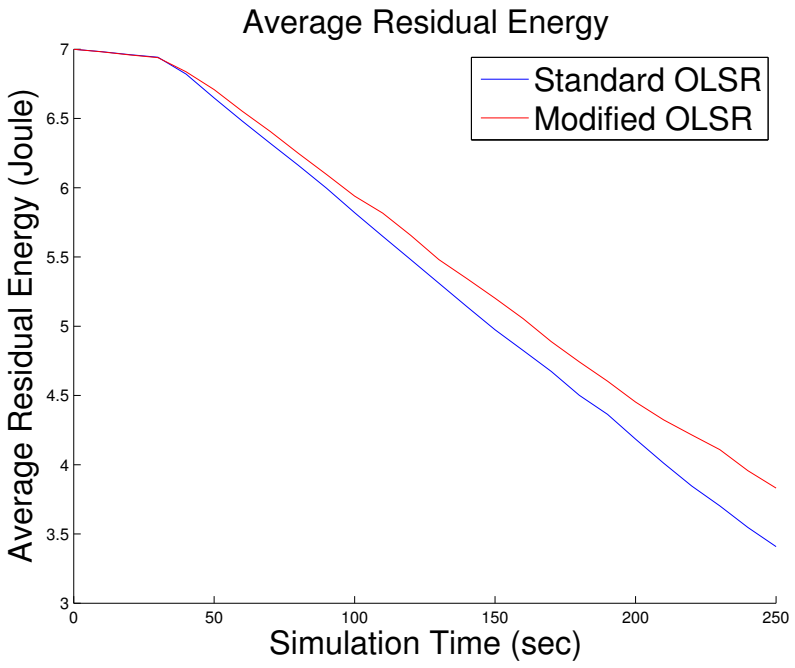


Figure 5.2: Average Residual Energy in Simulation Setup A

For the distribution of node residual energy, we present the results in Figure 5.3. We observe that in the modified OLSR than half of the nodes that have residual energy between 30-70% at the end of the simulation time (250 sec). On the other hand, in the standard OLSR there are a lot of nodes that have small percentage of residual energy or large percentage of residual energy. Thus, this indicates that our modified scheme achieves more uniform utilization of network resources by adjusting the weights and dispersing the traffic through different paths to reduce energy consumption. On the other hand, the standard OLSR selects the same paths to the destination nodes and utilizes the same intermediate nodes, which leads to fast depletion of their energy.

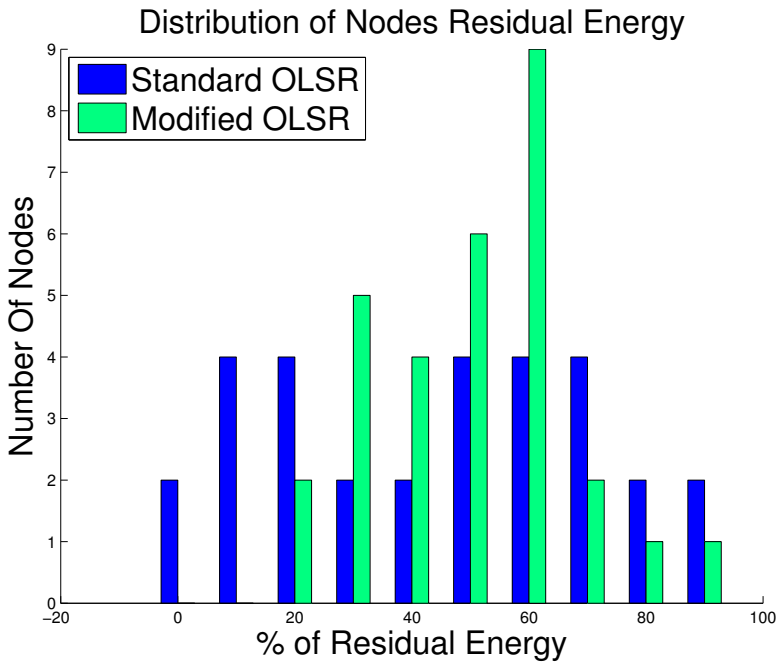


Figure 5.3: Distribution of node residual energy in Setup A

5.3.3.2 Simulation Setup B

In this setup, we consider static and mobile scenarios and also different data rates. The purpose of this experimentation is to examine whether the modified scheme contributes to the increase of network lifetime, but at the same time without loss of performance (in terms of PDR). The different mobility patterns indicate the behavior of the modified scheme under different network dynamics. We run three simulations with different source and destination pairs in the network.

In the static scenario, we observe in Figure 5.5 that the modified OLSR achieves improvement of 5-20% in network lifetime, in comparison with the standard OLSR. This indicates that the modified scheme selects alternative paths, maybe longer paths, when an intermediate node has consumed most of its energy, and by-

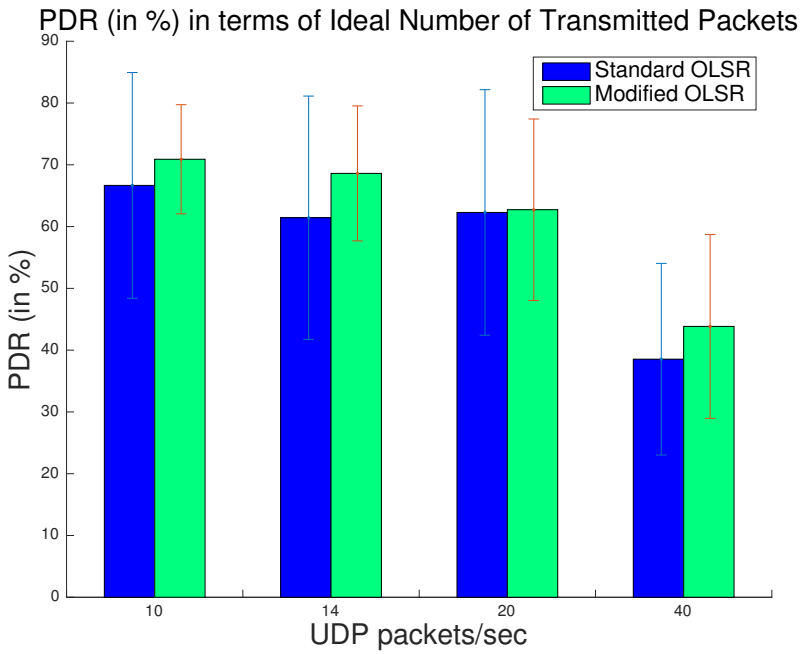


Figure 5.4: PDR for different packet interarrival times (static scenario)

passes the energy "weak" node. In addition, PDR has an improvement of up to 10% in the case of the modified OLSR (Figure 5.4). This is justified from the fact that queue utilization is taken into account as part of our weight function. Thus, the modified OLSR imposes higher weight to congested nodes, that are most likely to cause loss of packets, and avoids to choose to forward packets through them in the routing procedure. This causes this slight improvement in PDR.

In the low mobility scenario, we observe (Figures 5.6 and 5.7) similar results with the static scenario. There is 5-15% improvement in network lifetime and PDR for the reasons explained above. In the high mobility scenario, the improvement in the network lifetime is lower than in the other cases (Figures 5.8 and 5.9), due to the fact that the mobility imposes dynamic change of intermediate nodes selected for routing. In this scenario, both the standard and the modified OLSR select

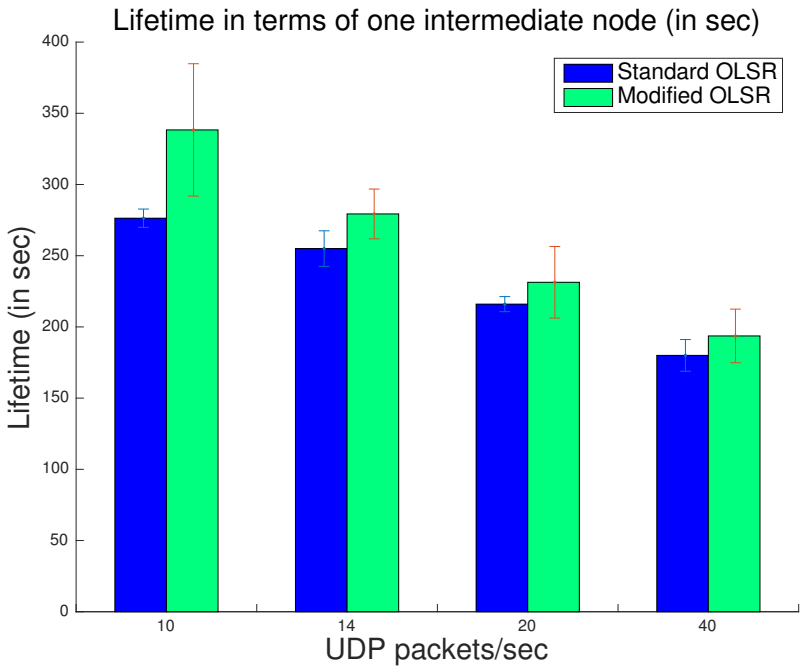


Figure 5.5: Network Lifetime for different packet interarrival times (static scenario)

several different routes during simulation. In addition, PDR remains the same as in the standard OLSR and at some rates is being slightly decreased. This effect comes from the lack of robustness of our scheme to adapt to highly changing dynamic environments. In high mobility conditions, the modified OLSR cannot learn the weights fast enough to take the routing decisions and this leads to more than expected packet losses.

Overall, in static and low mobility scenarios, we notice better PDR than in high mobility scenarios, due to the fact that our protocols have slow convergence in rapidly changing dynamics. In addition, in high mobility scenarios, we achieve greater network lifetime than in the other two scenarios, because the protocols choose a lot of different alternative paths due to the dynamic changes in the network topology. Finally, in all three scenarios, as we increase the data rate we observe

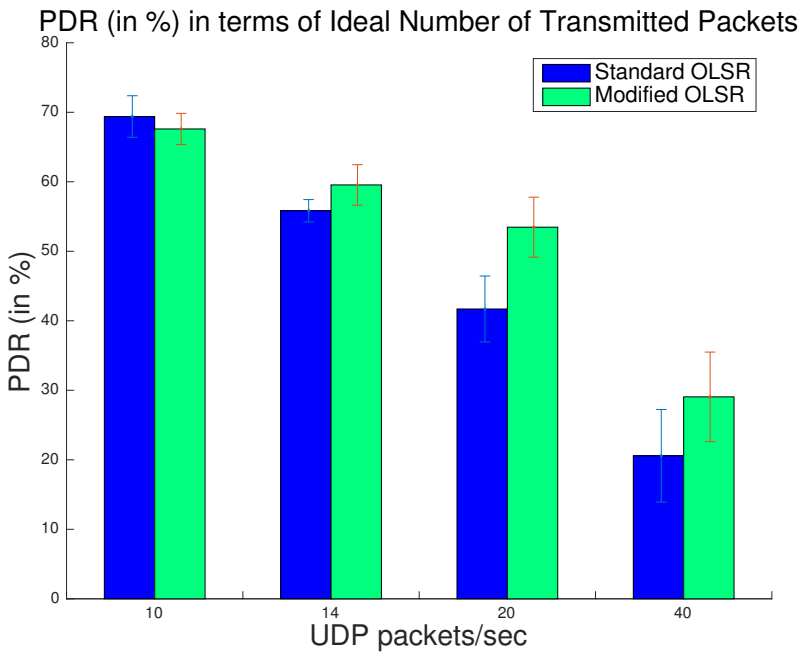


Figure 5.6: PDR for different packet interarrival times (low mobility)

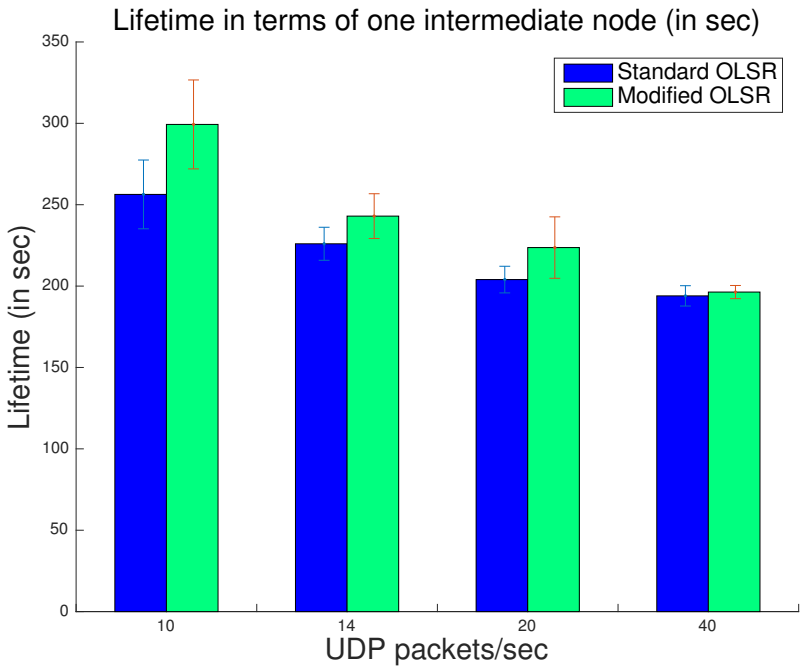


Figure 5.7: Network Lifetime for different packet interarrival times (low mobility)

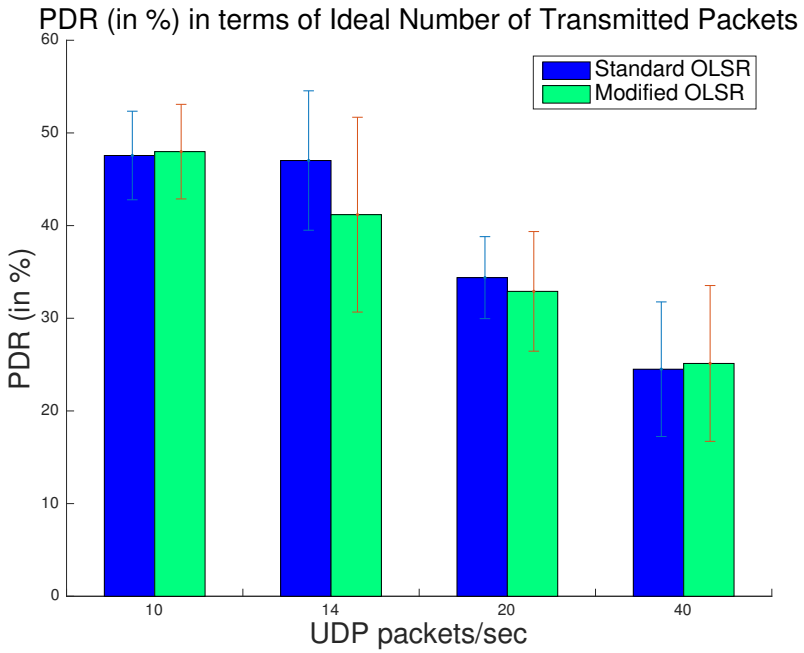


Figure 5.8: PDR for different packet interarrival times (high mobility)

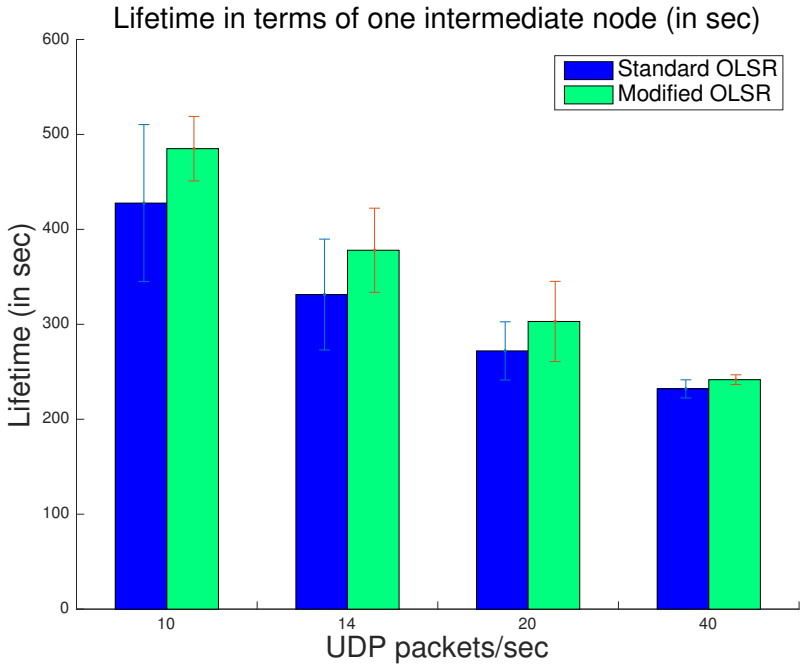


Figure 5.9: Network Lifetime for different packet interarrival times (high mobility)

decrease in PDR and in network lifetime. In the case of higher rates, a lot of packets are lost due to network congestion and bandwidth limitations (link bandwidth 1 Mbps), which results to the decrease of the PDR. In addition, network lifetime is decreased due to the transmission (and retransmission from the MAC layer) of large number of packets in the case of high data rates.

5.4 Concluding Remarks

In this chapter, we proposed a novel multi-metric energy efficient routing scheme, integrated in the OLSR routing protocol. Three cross-layer parameters, which indicate energy depletion, are considered to form a weight, representing the cost of routing through this node. Node weights are updated and sent to the network through TC packets periodically to be taken into consideration for routing decisions. We evaluated the Modified OLSR under a range of different scenarios, varying traffic load and mobility pattern. The modified version of the OLSR achieves significant increase in network lifetime (5-20%), without loss of performance (in terms of PDR).

Future work includes experimenting on larger networks. In large networks we may have slow routing protocol convergence by relying to TC messages, hence the robustness of our scheme needs to be investigated. In addition, we are currently working on the formulation of an optimization problem, in order to assign appropriate weights according to the current state of the network.

CHAPTER 6

Distributed Energy-Aware Sensor Coverage: A Game Theoretic Approach

In the two previous chapters, we proposed and implemented energy efficient schemes, which can be easily integrated to existing routing protocols used for different types of wireless multi-hop networks. In this chapter, we investigate energy-aware sensor coverage, which is a crucial aspect of the operation of wireless sensor networks.

6.1 Overview

Efficient monitoring of large areas has attracted great interest from the research community in the recent past. Monitoring incorporates the collection and process of measurements related to temperature, humidity and other quantities of interest. For this purpose, mobile sensor networks (MSN) are being used to cover large areas under surveillance. Examples include coverage via large video cameras, environmental monitoring, monitoring for threats, monitoring for transportation congestion and efficiency, monitoring for medical and other emergencies. Sensor

networks are also used in smart grid technologies. Mobile sensors have limited memory and energy resources and this needs to be taken into consideration while deploying such a sensor network, in order not these resources to be depleted.

In this chapter, we examine the mobile sensor coverage problem, while addressing the energy limitations of such types of networks. Mobile nodes aim to collectively optimize a global objective by making optimal local decisions. The objective is to maximize the benefits from coverage in a particular area and simultaneously minimize energy consumption due to sensing.

The energy-aware coverage problem is modeled using a game theoretic approach, where all the mobile sensor nodes participate in a game. Similar approaches, which use game theory for coverage problems have been presented in [95], [26], [27]. The goal of this approach is to introduce an efficient learning rule that ensures the existence of a pure Nash Equilibrium (NE) [96] and guarantees the convergence to NE.

We propose a distributed learning rule, based on [25], which enables a bit-valued information exchange over a communication graph. Our learning strategy focuses on optimizing the welfare of the game (i.e., sum of individual agents' utilities) for any arbitrary design of utility functions. We show that our algorithm induces a perturbed Markov chain and moreover, we prove convergence to a pure NE.

6.1.1 Related Work

Several approaches have been proposed for the sensor coverage problem including some recent advancements [26], [27]. In [97], an optimization problem is defined to maximize sensors' coverage while taking into account the communication cost. Another method for sensor coverage has been introduced in [98], where the authors propose an estimation model consisting of a summation of n distributions and the estimation algorithm adjusts the weighting functions of these distributions. However, the proposed estimation scheme does not scale satisfactorily with n.

Game theoretic approaches have also been widely adopted to optimize sensor coverage problems. The approach was introduced in [23] and [24] and has been used for solving decision making problems in a multi-agent setup. Martinez et al. [26] propose a game-theoretic approach for distributed coverage using a mobile sensor network and they introduce a novel utility function that captures the trade-off between efficient coverage and energy consumption. [27] studies a sensor coverage potential game, using reinforcement learning, with a utility function that takes into consideration energy consumption due to sensing and movement.

A variety of decentralized learning rules have been proposed for optimal action selection. Some of them are independent of the utility design, but are still proved to lead to a pure NE. We propose a similar learning rule in our work, which is optimal for a broad class of utility functions. In [99], Marden et al. proposed a decentralized learning algorithm to address the issue of unknown payoff structure. The proposed algorithm allows agents to learn actions that lead to welfare maximization without

any knowledge of the functional form of the utilities. The algorithm can be used for optimization of complex systems with many distributed components, such as the routing of data packets in networks and the design and control of wind farms. However, the convergence of this algorithm is guaranteed only under an assumption on the form of the utilities called "interdependence". To overcome this, new decentralized learning rules were proposed by Menon et al. in [25] and [100]. The learning rule in [25] uses a bit-valued inter-agent communication and is proved to converge to a NE, without the interdependence property for the utility functions. In our work, we modify this distributed learning algorithm to model the energy-aware mobile sensor coverage game and derive new (relaxed) conditions for convergence to NE. We have introduced a variation of this algorithm for continuous action space, which is then partitioned into a finite set of states.

6.1.2 Summary of Contributions

In this chapter, we present three main contributions in this problem. Firstly, we design a suitable utility function to capture the trade-off between sensing/processing and energy consumption. We have defined novel coverage utility and optimization functions for our overall utility function. Secondly, we propose a distributed learning rule that enables bit-valued communication between the agents. Finally, we analyze and prove the convergence to a NE. Our proposed distributed learning algorithm also supports the case of continuous action space, which is essential for the mobile sensor coverage problem and it remained unaddressed in the vast majority of the

past work.

6.1.3 Chapter Organization

The rest of the chapter is organized as follows. Section 6.2 describes the preliminaries on game theory and perturbed Markov chains for the energy-aware coverage problem. Section 6.3 describes the problem statement and gives an introduction to our approach. Section 6.4 contains the description of the distributed learning algorithm, the analysis of the proposed algorithm, and the proof of convergence. Finally, we conclude our work in Section 6.5.

6.2 Preliminary Background

6.2.1 Game Theory Background

In this section, we provide some basic definitions from game theory [26], [96], [101] that we use for our model. Based on these principles, we formulate the energy-aware optimal coverage problem as a cooperative game among the sensor nodes.

Definition 6.2.1. *A strategic game* $\Gamma =< \mathcal{V}, \mathcal{A}, \mathcal{U} >$ *consists of:*

1. *A set* $\mathcal{V}$ *of heterogeneous players, where* $i \in \mathcal{V} = \{1, \ldots, N\}$.
2. *An action set* $\mathcal{A} := \prod_{i=1}^{N} \mathcal{A}_i$*, the space of all actions, where* $\alpha_i \in \mathcal{A}_i$ *is the action of player* i *and an (multiplayer) action* $\alpha \in \mathcal{A}$ *has components* $\alpha_1, \ldots, \alpha_N$.
3. *The utility function* $U_i : \mathcal{A} \rightarrow \mathbb{R}$*, which models the payoff of player* i *over action profiles.*

Definition 6.2.2. *Let α_{-i} be the action profile of all the other players except i and $\mathcal{A}_{-i} = \prod_{j \neq i} \mathcal{A}_j$.*

The notion of NE [101] is crucial in game theory setup and is defined as follows:

Definition 6.2.3. *Consider the strategic game Γ. An action profile $\alpha^* := (\alpha_i^*, \alpha_{-i}^*)$ is a NE of the game Γ, if for all $i \in \mathcal{V}$ and for all $\alpha_i \in \mathcal{A}_i$ it holds that $U_i(\alpha^*) \geq U_i(\alpha_i, \alpha_{-i}^*)$.*

An action profile corresponding to a NE indicates an action in which no player has benefit to deviate. The objective of the multi-agent system is to collaboratively maximize the welfare function $W^* = max_{\alpha \in \mathcal{A}} W(\alpha)$, where $W(\alpha) = \sum_{i=1}^{N} U_i(\alpha)$.

6.2.2 Perturbed Markov Chains

In this section, we describe the definitions and the theory of perturbed Markov chains [25], [102]. Let $P(0)$ be the 1-step transition probability matrix of a Markov chain on a finite state space S. We refer to this chain as the *unperturbed chain*.

Definition 6.2.4. *A regular perturbation of $P(0)$ consists of a stochastic matrix valued function $P(\epsilon)$ on some non-degenerate interval $(0, a]$ that satisfies, for all $x, y \in S$,*

1. *$P(\epsilon)$ is irreducible and aperiodic for all $\epsilon \in (0, a]$,*
2. *$\lim_{\epsilon \to 0} P_{x,y}(\epsilon) = P_{x,y}(0)$ and*
3. *if $P_{x,y}(\epsilon) > 0$ for some ϵ, then $\exists\, r(x, y) \geq 0$ such that $0 < \lim_{\epsilon \to 0} \epsilon^{-r(x,y)} P_{x,y}(\epsilon) < \infty$.*

From the first condition in Def. (6.2.4) we conclude that there exists a unique stationary distribution $\mu(\epsilon)$, which satisfies $\mu(\epsilon)P(\epsilon) = \mu(\epsilon)$ for each $\epsilon \in (0, a]$. The other two conditions indicate how the perturbed chain converges to the unperturbed one as $\epsilon \to 0$.

Let $\mathcal{L} = \{f \in \mathcal{C}^\infty |\ f(\epsilon) \geq 0, f(\epsilon) = \sum_{i=1}^{L} a_i \epsilon^{b_i}$ for some $a_i \in \mathbb{R}, b_i \geq 0,\ Dom(f) = (0, \infty)\}$ for some large enough but fixed $L \in \mathbb{N}$, where $\mathcal{C}^\infty$ is the space of smooth functions.

We introduce some notation that will be helpful while stating the main result regarding perturbed Markov chains. The parameter $r(x, y)$ is called the *1-step transition resistance* from state x to y. Notice that $r(x, y) = 0$ holds only for the one step transitions $x \to y$ allowed under $P(0)$. A *path* $h(a \to b)$ from a state $a \in S$ to $b \in S$ is an ordered set $\{a = x_1, x_2, \ldots, x_n = b\} \subseteq S$, such that every transition $x_k \to x_{k+1}$ in the sequence has positive 1-step probability according to $P(\epsilon)$. The resistance of the path is define as

$$r(h) = \sum_{k=1}^{n-1} r(x_k, x_{k+1}) \tag{6.1}$$

Definition 6.2.5. *For any two states x and y, the* resistance *from x to y is defined by $\rho(x, y) = \min\{r(h) |\ h(x \to y)$ is a path$\}$.*

Definition 6.2.6. *Given a subset $A \subset S$, its* co-radius *is given by $CR(A) = \max_{x \in S \setminus A} \min_{y \in A} \rho(x, y)$.*

Hence, $\rho(x, y)$ can be defined as the minimum resistance over all possible paths starting at state x and ending at state y. The co-radius indicates the maximum resistance that must be overcome in order to enter it from outside. We extend the

definition of resistance to include resistance between two subsets $S_1, S_2 \subset S$:

$$\rho(S_1, S_2) = \min_{x \in S_1, y \in S_2} \rho(x, y). \tag{6.2}$$

Since $P(\epsilon)$ is irreducible for $\epsilon > 0$, $\rho(S_1, S_2) < \infty$ for all $S_1, S_2 \subset S$.

Definition 6.2.7. *A* recurrence or communication class *of a Markov chain is a non-empty subset of states $E \subseteq S$ such that for any $x, y \in E$, $\exists\ h(x \to y)$ and for any $x \in E$ and $y \in S \setminus E$, $\nexists\ h(x \to y)$.*

Let us denote the recurrence classes of the unperturbed chain $P(0)$ as $E_1, ..., E_M$ and $E = \{E_1, E_2, \cdots E_M\}$. Let us consider a directed graph $\mathcal{G}_{RC}$ on the vertex set $\{1, ..., M\}$ with each vertex corresponding to a recurrence class. Let a *j-tree* be a spanning subtree in $\mathcal{G}_{RC}$ that contains a unique directed path from each vertex in $\{1, ..., M\} \setminus \{j\}$ to j and denote the set of all j-trees in $\mathcal{G}_{RC}$ by $\mathcal{T}^j_{RC}$.

Definition 6.2.8. *The* stochastic potential *of a recurrence class E_i is*

$$\gamma(E_i) = \min_{T \in \mathcal{T}^i_{RC}} \sum_{(j,k) \in T} \rho(E_j, E_k). \tag{6.3}$$

Let $\gamma^* = \min_{E_i} \gamma(E_i)$. Finally, we can state the main result regarding perturbed Markov chains, based on [102].

Theorem 6.2.9 ([102])**.** *Let $E_1, ..., E_M$ denote the recurrence classes of the Markov chain $P(0)$ on a finite state space S. Let $P(\epsilon)$ be a regular perturbation of $P(0)$ and let $\mu(\epsilon)$ denote its unique stationary distribution. Then,*

1. *As $\epsilon \to 0$, $\mu(\epsilon) \to \mu(0)$, where $\mu(0)$ is a stationary distribution of $P(0)$ and*

2. *A state is stochastically stable i.e. $\mu_x(0) > 0 \Leftrightarrow x \in E_i$ such that $\gamma(E_i) = \gamma^*$.*

6.3 Problem Statement

In this work, we analyze a scenario of a Mobile Sensor Network (MSN), where sensors are randomly deployed in an area and their task is to monitor this area. The tasks of the sensors are to optimally move in the area since not all parts of the area are equally valuable at all time. In addition, sensors are equipped with limited battery power that should be spent judicially throughout the monitoring procedure. Hence, mobile sensors aim at maximizing the payoff from sensing specific portions of the area, while minimizing the overall energy consumption. Since the ultimate task is global for all the sensors, they unanimously have to decide their actions, in order to perform the above mentioned trade-off. This multi-agent trade-off problem can be formalized as a multi-player game [26].

A motivation example for the sensor coverage problem using game theoretic approach is presented in Fig. 6.1. In this example, there are 5 sensors (agents), which aim at maximizing the coverage over the monitoring area and the function of significance is stepwise. The function of significance defines the utility of coverage and is defined a priori by doing an inspection on the area based on the objective. In our example, there are two parts of the coverage area that have significance function equal to 10 and 5 respectively. For example, if our objective is to measure the temperature and the humidity in a building, we could use a similar stepwise function and allow more significance to the offices rather than the corridors of the building. In the given example, we observe that by using the game theoretic approach, the different nodes cooperate and avoid deploying additional coverage (in the case they

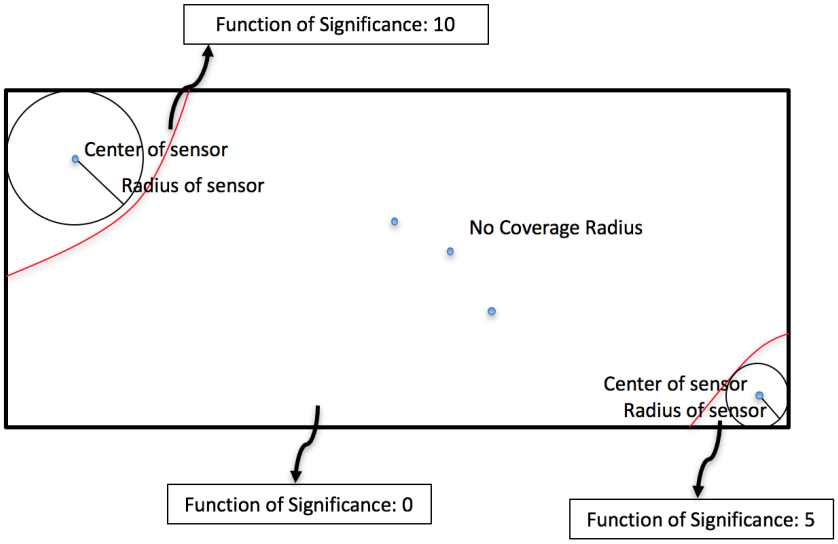


Figure 6.1: Coverage Example

did not cooperate), which would lead to energy consumption. Hence, in this example, one node covers the area with significance function equal to 10, the other node covers the area with significance function equal to 5 and the other 3 nodes remain idle in order not to reduce energy due to coverage.

We define the action profile (in game theoretic sense) at time t by a_t and the corresponding action for the the i-th agent is $(a_t)_i$ in our game. The action in this scenario consists of selecting the position o_t and the radius of sensing r_t. We treat the sensors to be points in the space and they can sense a circular region centering itself for some prespecified radius. We consider a two-dimensional area, which is discretized into a lattice. Each square of the lattice has unit dimensions and is labeled with the coordinate of its center $p = (p^x, p^y)$. The collection of all squares of the lattice is denoted by P. The sensors can place themselves only at these lattice centers and they have the privilege to choose a sensing radius around

them. As can be predicted the higher the radius the higher the energy expenditure to sense the region. The location of the sensor (agent) in our scenario is denoted by $(o_t)_i = ((p_t^x)_i, (p_t^y)_i) \in P$. The sensing area is defined by a disc with radius $(r_t)_i$ that takes values within a range $[r_{min}, r_{max}] \subset \mathbb{R}$. Each agent's action can be modeled as a tuple of the position and the radius. Hence, for agent i the action is denoted as $(a_t)_i := ((o_t)_i, (r_t)_i) \in (\mathcal{A}_{t-1})_i$, where $(\mathcal{A}_{t-1})_i$ is the available action set for agent i at time t. The action profile for all agents is $a_t = ((a_t)_1, \ldots, (a_t)_N) \in \mathcal{A} = \prod_{i=1}^{N} (\mathcal{A}_{t-1})_i$. The current available action set contains the time index $t-1$, since this set may be constrained based on the action chosen at time $t-1$, i.e. formally speaking $\mathcal{A}_{t-1} = \mathcal{A}(a_{t-1})$.

6.4 Game Theoretic Approach

In this section, the utility function for the game is formulated and we also describe a distributed algorithm, which can be used to achieve a NE for the underlying game.

6.4.1 Utility Design

As described before, the utility function should capture the trade-off between the effectiveness of sensor coverage and the energy consumption caused by sensing. At this point, suppose that we have examined a priori the coverage space and that we have defined a function of significance over the space X. We denote this function by $f : X \to \mathbb{R}$. The lattice structure P is constructed over this space X. In order

to find the covered area by a sensor, we define a disk $D_i((o_t)_i, (r_t)_i)$ centered at $(o_t)_i$ with radius $(r_t)_i$ around the sensor-agent. Hence, the total area which is sensed by the sensors can be expressed as $X_{covered}(a_t) = (\bigcup_{i=1}^{N} D_i((o_t)_i, (r_t)_i) \bigcap P)$.

The coverage gain is expressed with the function $F(a_t)$ that is defined as

$$F(a_t) = \int_{X_{covered}(a_t)} f(\xi) d\xi \tag{6.4}$$

We also need to define the energy consumption caused by sensing, transmitting and receiving packets from other agents. The energy consumption due to sensing and reception is proportional to the covered region. This can be modeled by $E_i^{cons} = C_i((r_t)_i)^2$, where $C_i > 0$ is a constant that depend on the sensor.

The trade-off between the coverage gain and energy consumption for agent i is captured in U_i in the following way

$$U_i(\alpha_t) = F((\alpha_t)_i) - E_i^{cons}((\alpha_t)_i) \tag{6.5}$$

In our case, the objective is to maximize $W = \sum_{i=1}^{N} U_i$ and hence we seek for actions:

$$\mathcal{A}^* = \{\arg\max_{a \in \mathcal{A}} W(a)\}$$

In the following section, we state the distributed learning strategy for our energy-aware coverage game. However, it should be noted that the algorithm defined and analyzed in the following section is general for all similar multi-agent games and we use the sensor coverage problem as an direct application of the algorithm.

6.4.2 Distributed Learning Strategy

In this section, we describe the distributed learning strategy that is used to reach a NE. The distributed algorithm is based on the recent work [25], [99]. However, several modifications were needed for our purpose. Our framework consists of the communication graph $\mathcal{G}_c$, which is a directed graph representing the explicit information exchange between the agents. The directed edge (i, j) in $\mathcal{G}_c(a_t)$ indicates that agent i is able to send a message to agent j at time t, when the joint action a_t is chosen. We define the neighbors of agent i at time t for the action profile a_t in the communication graph as $\mathcal{N}_i(a_t)$.

We partition the continuous action space into finite number of states such that $R = \{r_i | i = 1, 2, .., k\}$ where r_i's are disjoint intervals within $[r_{\min}, r_{\max}]$ satisfying $\cup_{i=1}^{k} r_i = [r_{\min}, r_{\max}]$. Each agent selects the radius by a Gibbs distribution, given in Eq. (6.6). Let $(\mathcal{A}_{t-1})_i$ be the set of feasible actions for agent i at time t, $(\mathcal{A}_{t-1})_i^r$ denote the feasible components corresponding to the radius and $(\mathcal{A}_{t-1})_i^o$ correspond to the position of the mobile sensor.

The conditional probability for agent k choosing a radius from set r_j given the center of the sensor to be at o and the immediate past join action to be a_j is considered to be

$$p_k^t(r \in r_i | o, a_j) = \frac{1}{m(r_i)} \frac{\int_{r \in r_i} \epsilon_t^{-U_k(r,o,a_{-j})} dr}{\sum_{r_l \in (\mathcal{A}_{t-1})_k^r} \int_{r \in r_l} \epsilon_t^{-U_k(r,o,a_{-j})} dr} \tag{6.6}$$

where $m(r_i)$ is the measure of the set r_i i.e. the length of the corresponding interval in this case.

Using Mean-value Theorem for integrals, each of the integrals in (6.6) can be represented as $\int_{r\in r_i} \epsilon^{U_k(r,o,a_{-j})}dr = m(r_i)\epsilon^{U_k(\hat{r}_i,o,a_{-j})}$; where $\hat{r}_i$ is an interior point of the interval r_i determined by the mean value theorem. Let us denote a finite set $\hat{R} = \{\hat{r}_1, \hat{r}_2, \cdots \hat{r}_k\}$. With slight abuse of notation, representing $U_k(a, b, c) = U_k^a$ for fixed b and c, we can write (6.6) as follows

$$p_k^t(r \in r_i|o, a_j) = \frac{\epsilon_t^{-U_k^{\hat{r}_i}}}{\sum_{r_j\in(\mathcal{A}_{t-1})_k^r} m(r_j)\epsilon_t^{-U_i^{\hat{r}_j}}} \tag{6.7}$$

Let $\hat{r}_* \in (\mathcal{A}_{t-1})_k^r \subseteq \hat{R}$ such that $U_i^{\hat{r}*} = \max_{r_j\in(\mathcal{A}_{t-1})_k^r} U_i^{\hat{r}_j}$. It can be easily verified that

$$\lim_{\epsilon\to 0} \frac{p_k^t(r \in r_i|o, a_j)}{\epsilon^{(U_i^{\hat{r}*} - U_i^{\hat{r}_i})}} = \frac{1}{m(\hat{r}_*)} \tag{6.8}$$

This gives us the resistance between actions [Def. 6.2.4].

Each agent i is endowed with a state $(x_t)_i = [(a_t)_i, (m_t)_i]$ at time t, where $(a_t)_i$ corresponds to the action taken and $(m_t)_i$ is a $\{0, 1\}$-valued *mood* of the agent i at time t. As described in [25], $(m)_i = 1$ is defined as the *content* state and $(m)_i = 0$ is defined as the *discontent* state of the agent i. The moods of the agents are being exchanged through the communication graph $\mathcal{G}_c(t)$ at time t. We assume that the communication radius is different than the coverage radius and that the energy consumption due to communication is insignificant in comparison to the energy consumption due to coverage, because we only enable bit-value communication between the agents and no other information exchange.

The collection of the states of all agents at time t is represented as $x_t =$

$[a_t, m_t]$. For a given state x, we denote the joint action by a^x and joint mood by m^x; and similarly the action and the mood of i-th agent is denoted by $(a^x)_i$ and $(m^x)_i$ respectively.

Let $\{\epsilon_t\}_{t\in\mathbb{N}}$ with $\lim_{t\to\infty}\epsilon_t = 0$ and constant $l > 0$, are pre-specified. The agent i performs the following rules sequentially to update its action when the joint action in the last step was a_{t-1}. The performance does not depend on the initialization of the algorithm and it can be initialized randomly.

Algorithm 6.4.1.

Start

Step 1: *Receive $(m_{t-1})_j$ from all $j \in \mathcal{N}_i(t-1)$ i.e. the neighbors of i in $\mathcal{G}_c(t-1)$. Calculate the temporary mood $\tilde{m}_i$ as follows:*

1. *If $(m_{t-1})_j = 1\ \forall j \in \{i\} \cup \mathcal{N}_i(t-1)$ set $\tilde{m}_i = 1$;*

2. *else set $\tilde{m}_i = 0$.*

Step 2: *Pick $(a_t)_i = (o_i, r_i)$ as follows:*

$p_i^t(o, r|a_{t-1}) = p_i^t(o|a_{t-1})p_i^t(r|o, a_{t-1})$, and $p_i^t(r|o, a_{t-1})$ as given in (6.7). The choice of o is independent of a_{t-1} i.e. $p_i^t(o|a_{t-1}) = p_i^t(o)$.

1. *If $\tilde{m}_i = 1$, pick o_i from $(\mathcal{A}_{t-1})_i^o$ according to the following rules:*

$$p(o) = \begin{cases} 1 - \epsilon_t^l & \text{if } o = (o_{t-1})_i \\ \frac{\epsilon_t^l}{|(\mathcal{A}_{t-1})_i^o| - 1} & \text{otherwise} \end{cases} \tag{6.9}$$

2. *Else if* $\tilde{m}_i = 0$, *pick* o_i *uniformly from* $(\mathcal{A}_{t-1})_i^o$ *i.e.*

$$p(o) = \frac{1}{|(\mathcal{A}_{t-1})_i^o|} \tag{6.10}$$

Step 3: *Update the payoff* $(U_t)_i = U_i((a_t))$ *and define* $U_i^* = \max_{(a_t)_i}(U_t)_i$.

Step 4: *Update the mood* $(m_t)_i$ *as follows:*

1. *if* $\tilde{m}_i = 1$ *and* $((a_t)_i, (U_t)_i) = ((a_{t-1})_i, (U_{t-1})_i)$, *set* $(m_t)_i \sim Ber(1 - \epsilon_t^l)$;

2. *if* $\tilde{m}_i = 0$ *or* ($\tilde{m}_i = 1$ *and* $((a_t)_i, (U_t)_i) \neq ((a_{t-1})_i, (U_{t-1})_i)$)

   set $(m_t)_i \sim Ber(\epsilon_t^{U_i^* - (U_t)_i})$.

   where $Ber(\cdot)$ *is the Bernoulli distribution.*

Step 5: *Broadcast* $(m_t)_i$ *to the neighbors in* $\mathcal{G}_c(t)$.

Stop

Thus the learning strategy induces a non-homogeneous perturbed Markov chain $P(\epsilon_t)$ with state space in $\mathcal{A} \times \{0,1\}^N$. Let us denote $U_i^* - (U_t)_i$ by β_3^i which in general depends on the joint action a and hence on the state x. Sometimes, we will refer the same as $\beta_3^i(x)$ to explicitly show the dependence of β_3^i on x.

6.4.3 Algorithm Analysis

Let $\mathcal{E} = \left\{ \frac{n(\epsilon)}{d(\epsilon)} |\ n(\epsilon), d(\epsilon) \in \mathcal{L} \text{ and } deg(n(\epsilon)) \geq deg(d(\epsilon)) \right\}$, where $deg(f(\epsilon))$ is the lowest exponent of ϵ present in $f(\epsilon)$.

Proposition 6.4.2. *The distributed learning (Alg. 6.4.1) induces a perturbed Markov chain.*

Proof. Firstly, it is trivial to check that $\forall x, y \in S \lim_{\epsilon \to 0} P_{x,y}(\epsilon) = P_{x,y}(0)$. This is a direct consequence of the fact that $P_{x,y}(\epsilon) \in \mathcal{E}$ for all $x, y \in S$.

Secondly, consider any state $x_{t-1} = [a_{t-1}, m_{t-1}]$ at time $t-1$; for an agent i, irrespective of the modes of itself and others, it can choose the same action $(a_t)_i = (a_{t-1})_i = [o_i, r_i]$ at time t with a probability at least $\min\{(1-\epsilon_t^l), 1/|(\mathcal{A}_{t-1})_i^o|\}p(r_i|o_i, a_{t-1})$ [Alg. 6.4.1, Step 2]. Similarly the agent can choose any other action at time t with some probability strictly great than 0 (the exact lower bound on this probability can be calculated from step 2 of Alg. 6.4.1). The mood $(m_t)_i$ can be changed to 1 with probability at least $\min\{(1-\epsilon_t^l), \epsilon_t^{\beta_3^i}\}$, and can be set to 0 with probability greater than $\min\{\epsilon_t^l, 1-\epsilon_t^{\beta_3^i}\}$. Hence the chain is irreducible and aperiodic at the same time.

From the structure of the probabilities defined in (6.7) and the steps 1.3, 2.1 and 4 in the Alg. 6.4.1, it is clear that for every state $x, y \in S$, $P_{x,y}(\epsilon) \in \mathcal{E}$. Let, $P_{x,y}(\epsilon) = \frac{n(\epsilon)}{d(\epsilon)}$, where $n(\epsilon) = \sum_{i=0}^{L_n} \alpha_i^n \epsilon^{b_i^n}$; $\alpha_i^n \in \mathbb{R}$, $b_{i+1}^n > b_i^n \geq 0$, $L_n \in \mathbb{N}$ and similarly $d(\epsilon) = \sum_{i=0}^{L_d} \alpha_i^d \epsilon^{b_i^d}$; $\alpha_i^d \in \mathbb{R}$, $b_{i+1}^d > b_i^d \geq 0$ and $L_d \in \mathbb{N}$. Therefore $deg(n(\epsilon)) = b_0^n$ and $deg(d(\epsilon)) = b_0^d (\leq b_0^n)$. Hence, $\lim_{\epsilon \to 0} \epsilon^{deg(d(\epsilon)) - deg(n(\epsilon))} P_{x,y}(\epsilon) = \frac{\alpha_0^n}{\alpha_0^d}$. Therefore, $P_{x,y}(\epsilon)$ satisfies all the three properties of a perturbed Markov chain enlisted in definition 6.2.4. □

Remark 6.4.3. *A direct consequence of Proposition 6.4.2 is that $P(\epsilon)$ is a regular perturbation of $P(0)$ and $P(\epsilon)$ has a stationary distribution $\mu(\epsilon)$ that converges to*

$\mu(0)$ *(a stationary distribution of* $P(0)$*) as* $\epsilon \to 0$ *[Theorem 6.2.9].*

Definition 6.4.4. *Let,* $C^0 = \{x \in S |\ m^x = \mathbf{1},\ (a^x)_i = (o, r)\ s.t.\ r = \hat{r}_*(o, a^x)\}$ *and* $D^0 = \{x \in S |\ m^x = \mathbf{0},\ (a^x)_i = (o, r)\ s.t.\ r = \hat{r}_*(o, a^x)\}$

where $\hat{r}_*(o, a^x) = arg\min_{r \in \hat{R}} p(r, o, (a^x)_{-i})$ *and* $p(r, o, (a^x)_{-i}) = \frac{\epsilon_t^{U_i(r,o,(a^x)_{-i})}}{\sum_{\hat{r}_n \in \hat{R}} m(r_n)\epsilon_t^{U_i(\hat{r}_n,o,(a^x)_{-i})}}$.

$r_n \in R$ *is the interval such that* $\hat{r}_n \in r_n$.

Lemma 6.4.5. *([25]) If for every* $a \in \mathcal{A}$, $\mathcal{G}_c(a)$ *is strongly connected, the recurrence classes of the unperturbed chain* $P(0)$ *are* D^0 *and the singletons* $z \in C^0$.

Proof. Setting $\epsilon_t = 0$ in the Alg. 6.4.1, we can easily notice that $m_{t-1} = \mathbf{0}$ implies $m_t = \mathbf{0}$. So D^0 is a recurrence class according to $P(0)$. Similarly, $m_{t-1} = \mathbf{1}$ implies all $(m_t)_i = 1$ by step 1 of the algorithm. The step 2.1 of Alg. 6.4.1 along with (6.7) ensures all the agents select their previous actions. Hence each element of C^0 is a separate recurrence class. □

Lemma 6.4.6. *Under the same assumption as in Lemma 6.4.5, for any* $x, x' \in C^0$, $y, y' \in D^0$, *and* $z \in S \backslash (C^0 \cup D^0)$*:*

$$\rho(x, y) = kl, \tag{6.11}$$

$$\rho(y, x) = \sum_{i=1}^{N} \beta_3^i(x), \tag{6.12}$$

$$\rho(x, x') = l|\eta|, s.t.\ \eta = \{i : (o^x)_i \neq (o^{x'})_i\}, \tag{6.13}$$

$$\rho(y, y') = 0, \tag{6.14}$$

$$\rho(z, y) = 0, \tag{6.15}$$

Proof. Let k be the smallest number such that one can choose a set $I \subset \{1, 2, .., N\}$ of k agents in a way that $I \cup (\cup_{i \in I} \mathcal{N}_i)$ is the whole set of agents $\mathcal{V} = \{1, 2, \cdots, N\}$.

To change from a state in C^0 to a state in D^0, the agents $i \in I$ should change their moods using either step 4.1 or the combination of steps 2.1 (changing action) and 4.2. Both of these changes incur the same resistance l. $\forall j \in \mathcal{N}_i$, $\tilde{m}_j = 0$ as soon as $m_i = 0$. Mood m_j can be changed to 0 via a zero resistance path by step 4.2. Therefore k such agents need to change their moods so that all the agents can change their moods and hence the new state belongs to D^0. Note that, the change of the action $a = [o, r]$ under $\tilde{m}_i = 0$ can be done with zero resistance using step 2.2 and (6.7). This proves Eq. (6.11) and obviously $k \leq N$.

For a change of state from D^0 to any state in C^0, the actions can be selected via a zero resistance path as in step 2.2. Since all $\tilde{m}_i = 0$, m_i has to be made equal to 1 via step 4.2 with a cumulative resistance of $\sum_{i \in \mathcal{V}} \beta_3^i(x)$ and hence Eq. (6.12) is obtained.

For a change from $x \in C^0$ to $x' \in C^0$, if any agent i has its center o_i different from its previous value, it can make such a change in action via a path of resistance l by step 2.1 or it can change its mood with resistance l and then choose the action $a^{x'}$ with a zero resistance path using step 2.2, and finally change its mood with resistance β_3^i by step 4.2. However, for the latter case, since agent i's change of mood will affect $\tilde{m}_j$ $\forall j \in \mathcal{N}_i$. Thus, the neighbors need a change from $\tilde{m}_j = 0$ to $m_j = 1$ by resistance β_3^j. Therefore, the minimum resistance for such a change will be used to adopt the former strategy, i.e. changing action using step 2.1, incurring a resistance l. By denoting $\eta = \{i : (o^x)_i \neq (o^{x'})_i\}$ and the cardinality of η by $|\eta|$, we arrive at Eq. (6.13), where $(o^x)_i$ is the center of the i-th agent at state x.

All the states in D^0 are accessible from one another under the unperturbed

Markov chain $P(0)$ and Eq. (6.14) holds.

Note that the $\mathcal{G}_c(a)$ is strongly connected and we divide the agents into two groups $\mathcal{V}_0 = \{i|\ \tilde{m}_i = 0\}$ and $\mathcal{V}_1 = \{i|\ \tilde{m}_i = 1\}$. Due to the strong connectivity assumption, for all $i \in \mathcal{V}_1$, $\exists j \in \mathcal{V}_0$ such that there is a path from j to i. Therefore, agents in $\mathcal{V}_0$ can change their actions with 0 resistance (step 2.2) in a way that affects the utility of some $i \in \mathcal{V}_1$ and as a consequence $m_i = 0$ with zero resistance (step 4.2). Thus finally for all $i \in \mathcal{V}$, $m_i = 0$. This fact along with Eq. (6.14) implies (6.15). □

Lemma 6.4.7. *The stochastically stable set of states is* $\{x_i \in C^0|\ W(a^{x_i}) = W^*\}$.

Proof. The proof follows the similar line of thoughts as done in [25] by constructing the $j - trees$ (Def. 6.2.8) rooted at $\{x_i \in C^0|\ W(a_i^x) = W^*\}$ and comparing it to the other $j - trees$ rooted at other nodes. However few difference should be noted here that:

1. An outward edge from D^0 to x_i has a resistance of 0 (Alg. 6.4.1 step 4.2). In [25], it was W^*.

2. The above fact required $l > W^*$ in [25] but we do not require any such constraint on l.

□

Theorem 6.4.8 (Main Result)**.** *Let for every action* $a \in \mathcal{A}$, $\mathcal{G}_c(a)$ *be strongly connected. Let* $x_t = [a_t, m_t]$ *denotes the state of all agents at time t, then*

$$\lim_{t\to\infty} P(a_t \in \mathcal{A}^*) = 1$$

Proof. This Theorem is similar to Theorem 1 in [25]. Only difference in our theorem is that we have relaxed the condition $\sum_{t=1}^{\infty} \epsilon_t^{\kappa} = \infty$ where $\kappa = \min_{E_i \in E} CR(E_i)$ and E is the set of recurrence classes of $P(0)$. By careful observation, we can say that $\kappa = 0$. To show this, we proceed by finding the co-radius of $x_i \in C^0$ such that $W(a^{x_i}) = W^*$. Let us take $v \in D^0$, then $\rho(v, v') = 0$ by Lemma 6.4.6 for all $v' \in D^0$. Let us choose $v' = (a^{v'}, m^{v'})$ such that $a^{v'} = a^{x_i}$. Therefore, clearly $\rho(v', x_i) = 0$ [Alg. 6.4.1 step 4.2] and hence $\rho(v, x_i) = \rho(v, v') + \rho(v', x_i) = 0$. Now, if $v \in S \backslash (C^0 \cup D^0)$, then $\rho(v, v') = 0$ for all $v' \in D^0$ and since we already have proved that $\rho(v', x_i) = 0$ for all $v' \in D^0$, we can conclude $\rho(v, x_i) = 0$ for all $v \in S \backslash (C^0 \cup D^0)$. Therefore $CR(x_i) = 0$ and that implies $\kappa = 0$. □

6.5 Concluding Remarks

In this chapter, we present a game theoretic methodology to solve the energy-aware coverage problem for mobile sensor networks (MSN) in a decentralized fashion. The utility function captures the trade-off between the efficient coverage and the energy consumption due to sensing, receiving packets and localization. The decentralized learning algorithm incorporates the exchange of certain bit-valued information between the agents over a directed communication graph. Finally, we prove that this algorithm converges to a NE. However, unlike the previous work [25], the convergence of ϵ_t is not constrained and consequently the convergence to NE.

CHAPTER 7

Conclusion and Future Work Directions

7.1 Conclusion

In this dissertation, we propose lightweight and efficient schemes from a practical, but also a more theoretical point of view, for secure and energy efficient protocol design for wireless multi-hop networks.

First of all, we present a novel framework for mitigating control and data plane attacks against wireless network protocols. We use the component-based protocol design to change the protocol functionality in a modular way in response to a variety of attacks. We have proposed a set of mitigation techniques, which are incorporated into protocol components and are reusable across different protocols of the same class. These mitigation techniques utilize the trust estimates to modify each component's functionality. The performance evaluation of these techniques indicates high performance recovery in different attack scenarios. Furthermore, we investigate the network performance and security tradeoff by introducing security considerations in the cross layer design of network protocols via network utility maximization. We use trust values propagated through the networks as the notion for

security. This approach provides a unified framework to introduce security through soft constraints in the optimization problem. In this way, users get higher utility by forwarding data through nodes of higher trust values and the resulting trust-aware protocols are resilient to attacks and network failures. We have also introduced delay constraints in the utility optimization problem to capture QoS requirements.

In the second part of this dissertation, we propose efficient distributed sleep management techniques for battery powered nodes in heterogeneous wireless machine-to-machine networks. The distributed sleep management techniques enable each battery-powered node to decide the optimal times to sleep or wake up and process data based on traffic measurements and predictions. We also propose two novel battery energy aware routing metrics, battery node energy waste (EW) and battery node relay cost (RC), which are used for path selection. The sleep management schemes and the energy aware routing metrics can be easily incorporated into existing protocols. In this work, we integrated the proposed schemes into a standard routing protocol for M2M networks, called B-RPL. We illustrate the effectiveness of our techniques by conducting extensive simulations. The results show significant improvement in terms of network lifetime and packet delivery ratio in comparison with benchmark cases. In addition, we introduce a novel multi-metric energy efficient routing scheme for mobile ad-hoc networks, which we integrated in the standard OLSR routing protocol to create the Modified OLSR routing scheme. Three cross-layer parameters, which indicate energy depletion, are considered to form a weight, representing the cost of routing through this node. We evaluated the Modified OLSR under a range of different scenarios, varying traffic load and

mobility pattern to indicate its efficiency in comparison with standard OLSR protocol. Finally, we present a game theoretic methodology to capture the tradeoff between sensor coverage and energy consumption in mobile sensor networks (MSN). We formulate an appropriate utility function to capture this trade-off and we propose a decentralized learning algorithm that incorporates the exchange of certain bit-valued information between the agents over a directed communication graph. Finally, we prove that this algorithm converges to a Nash Equilibrium (NE).

7.2 Future Work Directions

7.2.1 Component Activation Engine

As part of the future work for Chapter 2, we propose an automated mechanism to select the appropriate mitigation technique to be activated based on the network state and the observed performance of the implemented mitigation techniques under different environmental conditions. The decision making module is called *Component Activation Engine* and will use a combination of selection-rules and learning algorithms to handle the network dynamics, the noisy and false trust estimates, and the unpredictability of the attacks. Rule-based approach is running in the foreground of the engine in order to take fast decisions regarding the mitigated components that are going to be activated under a specific network state. We define by network state the number, the position of compromised nodes and the nature attack that they deploy. The established rules have been tested for their efficiency in terms of performance recovery in specific network states (environmen-

tal conditions). However, the engine needs additional mechanisms to deal with the dynamics of the environment, the noisy trust estimates and the unpredictability of the attacks. In order to adapt to these dynamics, the engine uses a numerical reward feedback derived from network performance measurements (e.g. throughout and latency) to evaluate the effectiveness of the activated mitigation technique. If the technique is ineffective, then its potential is being decreased for the specific network state. To provide ongoing, feedback-driven refinement of the rule set, we propose to use a reinforcement learning framework as introduced in [103] and [104] and in particular Collaborative Reinforcement Learning (CRL) as proposed in [105]. CRL is a self-organizing technique with the ability to perform complex collective tasks and to adapt system behavior or structure in a dynamic network environment, i.e., the one under attack or undergoing other frequent and unpredictable changes. It also enables distributed agents to solve decision-making problems online in dynamic networks using only partial knowledge of the environment. Therefore, CRL framework is crucial for our Component Activation engine, because it will establish new rules and also refine the already existing ones. CRL framework will run in the background and will compute the potential (effectiveness) of various mitigation techniques (actions taken by the engine) in different network states by taking into account the network performance measurements, which will take as feedback from this action. When the potential for a specific action passes a certain threshold, then the engine inserts the action in the rule set. Therefore, CRL framework will continuously generate and update rules for different network states.

7.2.2 Structured Node Placement in Heterogeneous M2M Networks

As part of the future work for Chapter 4, we investigate an algorithm for efficient placement of mains-powered nodes in the network in order to increase the efficiency of our scheme. We observed from the performance evaluation of this chapter that there was a significant increase in packet delivery ratio in the scenario that the mains-powered nodes were placed in a structured way around the gateway node. However, our proposed approach that was used in terms of performance evaluation is heuristic and we plan to introduce a more formal optimization approach. Krause et. al in [106] proposed a submodular optimization algorithm for efficient sensor placement, which maximizes the mutual information received from the sensor nodes. Another algorithm for efficient sensor placement was proposed in [107], which indicated that provides sufficient grid coverage of the sensor field. Inspired by the above mentioned and other similar algorithms, we could introduce a new formal optimization method for our mains-powered sensor placement that will lead to increase of packet delivery ratio and other performance metrics based on our optimization criteria.

Bibliography

[1] Sherin Abdel Hamid, Hossam S. Hassanein, and Glen Takahara. *Introduction to Wireless Multi-Hop Networks*, pages 1–9. Springer New York, New York, NY, 2013.

[2] Hoang Lan Nguyen and Uyen Trang Nguyen. A study of different types of attacks on multicast in mobile ad hoc networks. *Ad Hoc Networks*, 6(1):32–46, 2008.

[3] Baruch Awerbuch, Reza Curtmola, David Holmer, Cristina Nita-Rotaru, and Herbert Rubens. ODSBR: An On-demand Secure Byzantine Resilient Routing Protocol for Wireless Ad Hoc Networks. *ACM Transactions of Information Systems Security*, 10(4):6:1–6:35, January 2008.

[4] Ueli M. Maurer. Modelling a public-key infrastructure. In *Proceedings of the 4th European Symposium on Research in Computer Security: Computer Security (ESORICS)*, pages 325–350, London, UK, 1996.

[5] Virgil Gligor and Jeannette M. Wing. Towards a theory of trust in networks of humans and computers. In *Proceedings of the 19th International Conference on Security Protocols*, SP'11, pages 223–242, Berlin, Heidelberg, 2011. Springer-Verlag.

[6] G. Theodorakopoulos and J.S. Baras. On trust models and trust evaluation metrics for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):318–328, Feb 2006.

[7] R. Chadha, A. Ghosh, A. Poylisher, and C. Serban. Trend: Trust estimation system for wireless networks via multi-pronged detection. In *Proceedings of the IEEE Military Communications Conference, (MILCOM)*, pages 13–18, Oct 2015.

[8] T. Jiang and John S. Baras. Trust evaluation in anarchy: A case study on autonomous networks. In *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–12, April 2006.

[9] Yan Lindsay Sun, Zhu Han, Wei Yu, and K. J. Ray Liu. A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks. In *IEEE INFOCOM*, pages 230–236, 2006.

[10] M.A. Austin, V. Tabatabaee, and J.S. Baras. Systems engineering challenges for design and realtime management of component-enabled wireless ad-hoc networks. In *Proceedings of the 8th Conference on Systems Engineering Research (CSER 2010), Hoboken, NJ*, March 2010.

[11] J.S. Baras, V. Tabatabaee, P. Purkayastha, and K. Somasundaram. Component based performance modelling of wireless routing protocols. In *Proceedings of the IEEE International Conference on Communications*, pages 1 –6, June 2009.

[12] He Huang and John S. Baras. Component based routing: A new methodology for designing routing protocols for manet. In *Proceedings of the 25th Army Science Conference, Orlando, FL*, November 2006.

[13] J.S. Baras, V. Tabatabaee, and K. Jain. Component based modeling for cross-layer analysis of 802.11 mac and olsr routing protocols in ad-hoc networks. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 1–7, 2009.

[14] Di Tian and Nicolas D. Georganas. A coverage-preserving node scheduling scheme for large wireless sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, WSNA '02, pages 32–41, New York, NY, USA, 2002. ACM.

[15] Wei Ye, John Heidemann, and Deborah Estrin. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the IEEE Infocom*, pages 1567–1576, New York, NY, USA, June 2002. USC/Information Sciences Institute, IEEE.

[16] Jing Deng, Yunghsiang S. Han, Wendi B. Heinzelman, and Pramod K. Varshney. Scheduling sleeping nodes in high density cluster-based sensor networks. *Mob. Netw. Appl.*, 10(6):825–835, December 2005.

[17] R. Zheng and R. Kravets. On-demand power management for ad hoc networks. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 1, pages 481–491 vol.1, March 2003.

[18] Patrick Olivier Kamgueu, Emmanuel Nataf, Thomas Djotio Ndié, and Olivier Festor. Energy-based Routing Metric for RPL. Research Report RR-8208, January 2013.

[19] Oana Iova, Fabrice Theoleyre, and Thomas Nol. Using Multiparent Routing in RPL to Increase the Stability and the Lifetime of the Network. *Ad Hoc Networks*, 29, June 2015.

[20] F. De Rango, M. Fotino, and S. Marano. EE-OLSR: Energy Efficient OLSR routing protocol for Mobile ad-hoc Networks. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 1–7, Nov 2008.

[21] T. Kunz. Energy-Efficient Variations of OLSR. In *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 517–522, Aug 2008.

[22] J.-E. Garcia, A. Kallel, K. Kyamakya, K. Jobmann, J.-C. Cano, and P. Manzoni. A novel DSR-based energy-efficient routing algorithm for mobile ad-hoc networks. In *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, volume 5, pages 2849–2854 Vol.5, Oct 2003.

[23] Ragavendran Gopalakrishnan, Jason R. Marden, and Adam Wierman. An architectural view of game theoretic control. *SIGMETRICS Perform. Eval. Rev.*, 38(3):31–36, January 2011.

[24] Tansu Alpcan, L. Pavel, and N. Stefanovic. A control theoretic approach to noncooperative game design. In *Proceedings of the 48th IEEE Conference on Decision and Control held jointly with the 28th Chinese Control Conference (CDC/CCC)*, pages 8575–8580, Dec. 2009.

[25] Anup Menon and John S. Baras. A distributed learning algorithm with bit-valued communications for multi-agent welfare optimization. In *Proceedings of the 52nd IEEE Annual Conference on Decision and Control (CDC)*, pages 2406–2411, Dec. 2013.

[26] M. Zhu and S. Martinez. Distributed coverage games for energy-aware mobile sensor networks. *SIAM Journal on Control and Optimization*, 51(1):1–27, 2013.

[27] S. Rahili and Wei Ren. Game theory control solution for sensor coverage problem in unknown environment. In *Proceedings of the 53rd IEEE Annual Conference on Decision and Control (CDC)*, pages 1173–1178, Dec. 2014.

[28] M. Fecko, K. Manousakis, K. Young, Jaewon Kang, A. Pachulski, and W. Phoel. Mitigation of control plane attacks at the network layer. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 444–449, Oct 2015.

[29] Mariusz A. Fecko, Kyriakos Manousakis, Evripidis Paraskevas, Jaewon Kang, John S. Baras, Ken Young, and Wayne Phoel. COBRAM: Component-Based Reusable Adaptive Mitigation for Mobile Ad-Hoc Networks. *under preparation.*

[30] Evripidis Paraskevas, Tao Jiang, and John S. Baras. Trust-aware network utility optimization in multihop wireless networks with delay constraints. In *Proceedings of the 24th Mediterranean Conference on Control and Automation (MED)*, pages 593–598, June 2016.

[31] Evripidis Paraskevas, Jianlin Guo, Philip Orlik, and Kentaro Sawa. Distributed sleep management for heterogeneous wireless machine-to-machine networks. In *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, April 2016.

[32] T Winter, P. Thubert, and et al. RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks. `http://tools.ietf.org/html/rfc6550`, March 2012.

[33] P. Jacquet, P. Mhlethaler, T. Clausen, A. Laouiti, A. Qayyum, and L. Viennot. Optimized link state routing protocol for ad hoc networks. In *Proceedings of the IEEE International Multi Topic Conference INMIC*, pages 62–68, 2001.

[34] Evripidis Paraskevas, Kyriakos Manousakis, Subir Das, and John S. Baras. Multi-metric Energy Efficient Routing in Mobile Ad-Hoc Networks. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 1146–1151, October 2014.

[35] Evripidis Paraskevas, Dipankar Maity, and John S. Baras. Distributed energy-aware mobile sensor coverage: A game theoretic approach. In *Proceedings of the American Control Conference (ACC)*, pages 6259–6264, July 2016.

[36] M. Yu and K. K. Leung. A Trustworthiness-based QoS routing protocol for wireless ad hoc networks. *IEEE Transactions on Wireless Communications*, 8(4):1888–1898, April 2009.

[37] Feng He, Kuan Hao, and Hao Ma. S-MAODV: A trust key computing based secure Multicast Ad-hoc On Demand Vector routing protocol. In *Proceedings of the IEEE International Conference on Computer Science and Information Technology (ICCSIT)*, volume 6, pages 434–438, July 2010.

[38] Charles E. Perkins and Elizabeth M. Royer. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, 1997.

[39] G. Cervera, M. Barbeau, J. Garcia-Alfaro, and E. Kranakis. Mitigation of topology control traffic attacks in olsr networks. In *Risks and Security of Internet and Systems (CRiSIS), 2010 Fifth International Conference on*, pages 1–8, Oct 2010.

[40] Stefano Paris, Cristina Nita-Rotaru, Fabio Martignon, and Antonio Capone. Cross-layer Metrics for Reliable Routing in Wireless Mesh Networks. *IEEE/ACM Transactions on Networking*, 21(3):1003–1016, June 2013.

[41] Panagiotis Papadimitratos, Zygmunt J. Haas, and Senior Member. Secure Data Communication in Mobile Ad hoc Networks. *IEEE Journal On Selected Areas In Communications*, 24:343–356, 2006.

[42] K. Manousakis, K. Young, C. Graff, and M. Patel. On the design of power and delay aware k-connected topologies. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 1863–1868, Oct 2010.

[43] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[44] Junaid Ansari, Elena Meshkova, Wasif Masood, Arham Muslim, Janne Riihijärvi, and Petri Mähönen. CONFab: Component Based Optimization of WSN Protocol Stacks Using Deployment Feedback. In *Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '12, pages 19–28, 2012.

[45] Kevin Klues, Gregory Hackmann, Octav Chipara, and Chenyang Lu. A Component-Based Architecture for Power-efficient Media Access Control in Wireless Sensor Networks. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 59–72, 2007.

[46] Changbin Liu, Yun Mao, Mihai Oprea, Prithwish Basu, and Boon Thau Loo. A Declarative Perspective on Adaptive Manet Routing. In *Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow (PRESTO)*, pages 63–68, 2008.

[47] Charles E. Perkins and Pravin Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244, October 1994.

[48] David Johnson, Ntsibane Ntlatlapa, and Corinna Aichele. A simple pragmatic approach to mesh routing using BATMAN. In *2nd IFIP International Symposium on Wireless Communications and Information Technology in Developing Countries*, 2008.

[49] Evripidis Paraskevas and John S. Baras. Component based modeling of routing protocols for Mobile Ad Hoc Networks. In *Proceedings of the 49th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6, March 2015.

[50] Evripidis Paraskevas and John S. Baras. QoS Aware Component-Based Routing in Resource-Constrained Wireless Multi-hop Networks. In *Proceedings of the IEEE 23rd International Conference on Network Protocols (ICNP)*, pages 476–478, Nov 2015.

[51] Andres Medina and Stephan Bohacek. A performance model of neighbor discovery in proactive routing protocols. In *Proceedings of the 7th ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks, PE-WASUN 2019, Bodrum, Turkey, October 17-18, 2010*, pages 66–70, 2010.

[52] Kiran K. Somasundaram and John S. Baras. Semiring Pruning for Information Dissemination in Mobile Ad Hoc Networks. In *Proceedings of the 2009 First International Conference on Networks & Communications*, NETCOM '09, pages 319–325, Washington, DC, USA, 2009. IEEE Computer Society.

[53] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint relaying for flooding broadcast messages in mobile wireless networks. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on*, pages 3866–3875, Jan 2002.

[54] Kiran K. Somasundaram and John S. Baras. Path Optimization and Trusted Routing in MANET: An Interplay between Ordered Semirings. In *Advances in Networks and Communications*, volume 132 of *Communications in Computer and Information Science*, pages 88–98. Springer Berlin Heidelberg, 2011.

[55] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 42(1):97–101, 2000.

[56] João Luís Sobrinho. Algebra and Algorithms for QoS Path Computation and Hop-by-hop Routing in the Internet. *IEEE/ACM Transactions on Networking*, 10(4):541–550, August 2002.

[57] J. Ahrenholz. Comparison of CORE network emulation platforms. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, pages 166–171, Oct 2010.

[58] Multi-Generator (MGEN), U.S. Naval Research Laboratory (NRL). `http://www.nrl.navy.mil/itd/ncs/products/mgen`.

[59] Mung Chiang, Steven H. Low, A. Robert Calderbank, and John C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. *Proceedings of the IEEE*, 95(1):255–312, January 2007.

[60] Lijun Chen, S.H. Low, Mung Chiang, and J.C. Doyle. Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–13, April 2006.

[61] E. Stai, S. Papavassiliou, and J.S. Baras. Performance-aware cross-layer design in wireless multihop networks via a weighted backpressure approach. *IEEE/ACM Transactions on Networking*, PP(99):1–1, 2014.

[62] L. Tassiulas and A Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transaction on Automatic Control*, 37(12):1936–1949, December 1992.

[63] Xiaojun Lin, Ness B. Shroff, and R Srikant. A tutorial on cross layer optimization in wireless networks. *IEEE Journal on Selected Areas in Communications*, 24(8):1452–1463, August 2006.

[64] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 255–265, Boston, Massachusetts, United States, 2000. ACM Press.

[65] Daniel P. Palomar and Mung Chiang. A tutorial on decomposition methods for network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(8):1439–1451, August 2006.

[66] Nikolaos Trichakis, Argyrios Zymnis, and Stephen Boyd. Dynamic network utility maximization with delivery contracts. In *in Proceedings of the IFAC World Congress*, pages 2907–2912, 2008.

[67] Fan Qiu, Jia Bai, and Yuan Xue. Towards optimal rate allocation in multi-hop wireless networks with delay constraints: A double-price approach. In *Proceedings of the IEEE International Conference on Communications (ICC)*, pages 5280–5285, June 2012.

[68] Mohammad Hassan Hajiesmaili, Mohammad Sadegh Talebi, and Ahmad Khonsari. Utility-optimal dynamic rate allocation under average end-to-end delay requirements. *CoRR*, abs/1509.03374, 2015.

[69] J.S. Baras, Tao Jiang, and P. Purkayastha. Constrained coalitional games and networks of autonomous agents. In *Proceedings of the 3rd International Symposium on Communications, Control and Signal Processing (ISCCSP)*, pages 972–979, March 2008.

[70] P. Tague, S. Nabar, J.A. Ritcey, and R. Poovendran. Jamming-aware traffic allocation for multiple-path routing using portfolio selection. *IEEE/ACM Transactions on Networking*, 19(1):184–194, Feb 2011.

[71] Justine Sherry, Chang Lan, Raluca Ada Popa, and Sylvia Ratnasamy. Blindbox: Deep packet inspection over encrypted traffic. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM, pages 213–226, New York, NY, USA.

[72] Matt Blaze, Joan Feigenbaum, John Ioannidis, and Angelos D. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, pages 185–210, 1999.

[73] Laurent Eschenauer, Virgil D. Gligor, and John Baras. On trust establishment in mobile ad-hoc networks. In *Proceedings of the Security Protocols Workshop*, pages 47–66. Springer-Verlag, 2002.

[74] Tao Jiang and J.S. Baras. Trust credential distribution in autonomic networks. In *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–5, Nov 2008.

[75] Jang-Won Lee, Mung Chiang, and A.R. Calderbank. Price-based distributed algorithms for rate-reliability tradeoff in network utility maximization. *IEEE Journal on Selected Areas in Communications*, 24(5):962–976, May 2006.

[76] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 66–80, New York, NY, USA, 2003.

[77] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[78] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1999.

[79] E. Ghadimi, O. Landsiedel, P. Soldati, and M. Johansson. A Metric for Opportunistic Routing in Duty Cycled Wireless Sensor Networks. In *Proceedings of the 9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, June 2012.

[80] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A High-throughput Path Metric for Multi-hop Wireless Routing. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, September 2003.

[81] Kaikai Liu, Jianlin Guo, Philip Orlik, Kieran Parsons, and Kentaro Sawa. Battery Energy Management in Heterogeneous Wireless Machine-to-Machine Networks. In *Proceedings of IEEE Vehicular Technology Conference (VTC-Fall)*, September 2015.

[82] Saumitra M. Das, Himabindu Pucha, Konstantina Papagiannaki, and Y. Charlie Hu. Studying Wireless Routing Link Metric Dynamics. In *Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement*, October 2007.

[83] N. Javaid, M. Aslam, and et al. ATCEEC: A New Energy Efficient Routing Protocol for Wireless Sensor Networks. In *Proceedings of IEEE International Conference on Communications (ICC)*, June 2014.

[84] De bin Zou and Yong-Bin Wang. Adaptive Energy-aware Routing Framework in Transmission Cost Constrained Wireless Sensor Networks. In *Proceedings of IEEE Global Communications Conference (GLOBECOM)*, December 2013.

[85] Srinivasan Keshav. A control-theoretic approach to flow control. In *Proceedings of the Conference on Communications Architecture &Amp; Protocols*, SIGCOMM '91, pages 3–15, New York, NY, USA, 1991. ACM.

[86] The Network Simulator NS-2. `http://www.isi.edu/nsnam/ns/`.

[87] Chansu Yu, Y Ben Lee, and Hee Yong Youn. Energy Efficient Routing Protocols for Mobile Ad Hoc Networks. *Wireless Communications and Mobile Computing (WCMC) Journal*, pages 959–973, 2003.

[88] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware Routing in Mobile Ad Hoc Networks. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, MobiCom '98, pages 181–190, New York, NY, USA, 1998. ACM.

[89] Christine E. Jones, Krishna M. Sivalingam, Prathima Agrawal, and Jyh Cheng Chen. A Survey of Energy Efficient Network Protocols for Wireless Networks. *Wirel. Netw.*, 7(4):343–358, September 2001.

[90] Lijuan Cao, K. Sharif, Yu Wang, and T. Dahlberg. Adaptive Multiple Metrics Routing Protocols for Heterogeneous Multi-Hop Wireless Networks. In *Proceedings of the 5th IEEE Consumer Communications and Networking Conference (CCNC)*, pages 13–17, Jan 2008.

[91] Floriano De Rango and Marco Fotino. Energy Efficient OLSR Performance Evaluation Under Energy Aware Metrics. In *Proceedings of the 12th International Conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems*, SPECTS'09, pages 193–198, Piscataway, NJ, USA, 2009. IEEE Press.

[92] S. Mahfoudh and P. Minet. An Energy Efficient Routing Based on OLSR in Wireless Ad Hoc and Sensor Networks. In *Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on*, pages 1253–1259, March 2008.

[93] Zhihao Guo and B. Malakooti. Predictive Multiple Metrics in Proactive Mobile Ad Hoc Network Routing. In *Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on*, pages 755–762, Oct 2007.

[94] The ns-3 simulator. `http://www.nsnam.org/`.

[95] J.R. Marden and A. Wierman. Distributed welfare games with applications to sensor coverage. In *Proceedings of the 47th IEEE Conference on Decision and Control (CDC)*, pages 1708–1713, Dec. 2008.

[96] Drew Fudenberg and David K. Levine. *The Theory of Learning in Games*, volume 1 of *MIT Press Books*. The MIT Press, June 1998.

[97] Wei Li and C.G. Cassandras. Distributed cooperative coverage control of sensor networks. In *Proceedings of the 44th IEEE Conference on Decision and Control, and European Control Conference (CDC-ECC)*, pages 2542–2547, Dec. 2005.

[98] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *Int. J. Rob. Res.*, 28(3):357–375, 2009.

[99] J.R. Marden, H.P. Young, and L.Y. Pao. Achieving pareto optimality through distributed learning. In *Proceedings of the 51st IEEE Annual Conference on Decision and Control (CDC)*, pages 7419–7424, Dec. 2012.

[100] Anup Menon and John S. Baras. Convergence guarantees for a decentralized algorithm achieving Pareto optimality. In *Proceedings of the 2013 American Control Conference (ACC)*, pages 1932–1937, June 2013.

[101] Drew Fudenberg and Jean Tirole. *Game Theory*. MIT Press, Cambridge, MA, 1991.

[102] H. P. Young. The evolution of conventions. *Econometrica: Journal of the Econometric Society*, pages 57–84, 1993.

[103] Richard S. Sutton and Andrew G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[104] Hsien-Po Shiang and Mihaela van der Schaar. Online learning in autonomic multi-hop wireless networks for transmitting mission-critical applications. *IEEE Journal Selected Areas in Communications*, 28(5):728–741, June 2010.

[105] J. Dowling, E. Curran, R. Cunningham, and V. Cahill. Using feedback in collaborative reinforcement learning to adaptively optimize MANET routing. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(3):360–372, May 2005.

[106] Andreas Krause, Ajit Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, June 2008.

[107] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar. Sensor placement for grid coverage under imprecise detections. In *Proceedings of the Fifth International Conference on Information Fusion*, volume 2, pages 1581–1587 vol.2, July 2002.