REAL-TIME ROUTING WITH PRIORITY SCHEDULING AND POWER ADJUSTMENT IN WIRELESS SENSOR NETWORKS

A THESIS
SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING
AND THE INSTITUTE OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By
Emine Büşra Çelikkaya
August, 2008

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

Assist. Prof. Dr. İbrahim Körpeoğlu (Advisor)

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

Dr. Cengiz Çelik

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

Prof. Dr. Hitay Özbay

Approved for the Institute of Engineering and Science:

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

ABSTRACT

REAL-TIME ROUTING WITH PRIORITY SCHEDULING AND POWER ADJUSTMENT IN WIRELESS SENSOR NETWORKS

Emine Büşra Çelikkaya
M.S. in Computer Engineering
Supervisor: Assist. Prof. Dr. İbrahim Körpeoğlu
August, 2008

Many wireless sensor network applications require real-time communication, and real-time applications require packets to reach destination on time. However, applications may send packets with different priorities and hence delay bounds for packets may vary significantly. Therefore packet differentiation in the network is essential for meeting the deadline requirements. We propose a routing protocol that supports real-time communication by utilizing transmit power adjustment in order to meet the deadline of urgent packets and use energy efficiently. Our protocol also provides packet scheduling and gives precedence to urgent packets. We have conducted experiments on our sensor network testbed to observe the effects of transmit power on end-to-end delay. As expected, increasing transmit power increases the range and link quality, and reduces the number of hops to reach destination. Therefore adjusting transmit power has a great effect on delivery time and can reduce the end-to-end delay. Our protocol, *Real-time Routing with Priority Scheduling and Power Adjustment*, uses different levels of transmit power for packets with different priorities. It sends urgent packets with maximum power to minimize end-to-end delay and lower priority packets with reduced power to save energy and balance the load on nodes. Simulation results show that our routing protocol increases the deadline meet ratio of packets and reduces the transmit energy spent per packet when compared to routing protocols that use fixed transmit power. Additionally, results indicate that our approach lessens the interference on sensor nodes that are caused by other transmissions and helps balancing the load on the nodes.

Keywords: Wireless Sensor Networks, Routing Protocol, Real-time Applications, Transmit Power Adjustment, Energy Efficiency.

ÖZET

KABLOSUZ ALGILAYICI AĞLARINDA PAKET ÖNCELİĞİNE GÖRE ZAMANLAMA VE GÜÇ YÖNETİMİ DESTEKLİ GERÇEK ZAMANLI YÖNLENDİRME

Emine Büşra Çelikkaya
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Dr. İbrahim Körpeoğlu
Ağustos, 2008

Kablosuz algılayıcı ağları için geliştirilmiş pek çok uygulama gerçek zamanlı iletişime gerek duymaktadır ve gerçek zamanlı uygulamalar paketlerin varmaları gereken noktaya zamanında ulaşmalarını gerektirmektedir. Ancak bu uygulamalar değişik öncelikte paketler gönderebilir ve paketlerin gecikme toleransları birbirinden farklı olabilir. Bu yüzden gönderilen paketleri önceliğine göre ayırdetmek hedefe zamanında ulaşmaları açısından büyük önem taşır. Bu bağlamda, önerdiğimiz yönlendirme protokolü ile radyonun iletim gücünü ayarlayarak acil paketleri zamanında yerlerine ulaştırmak ve mümkün olduğunda iletim gücünü azaltarak enerji tüketimini azaltmak istiyoruz ve bu şekilde gerçek zamanlı iletişimi desteklemeyi amaçlıyoruz. Önerdiğimiz protokol ayrıca paketleri önceliğine göre zamanlayarak acil paketlere öncelik verilmesini sağlıyor. Radyo iletim gücünün gecikme üzerindeki etkisini gözlemleyebilmek için algılayıcı ağları test ortamımızda çeşitli deneyler yaptık. Tahmin edildiği üzere, iletim gücünü artırmak ulaşım menzilini ve bağlantı kalitesini artırarak hedefe ulaşmak için gereken zıplayış sayısını azaltıyor. Bu nedenle radyo gücünü ayarlamak paketlerin varış zamanlarını büyük ölçüde etkiliyor ve aradaki gecikmeyi azaltabiliyor. *Paket Önceliğine Göre Zamanlama ve Güç Yönetimi Destekli Gerçek Zamanlı Yönlendirme* protokolümüz farklı öncelikte paketler için değişik seviyelerde iletim gücü kullanıyor. Acil olan paketleri aradaki gecikmeyi azaltmak için daha yüksek güçler kullanarak gönderiyor. Ayrıca enerji kaybını azaltmak ve algılayıcı birimlerine yükü orantılı dağıtmak için düşük öncelikteki paketleri düşük seviyede güç kullanarak gönderiyor. Simülasyon sonuçları önerdiğimiz protokolün sabit güç kullanan protokollerle karşılaştırıldığında daha çok paketi süresi bitmeden varması gereken yere ulaştırdığını ve radyoda harcanan enerjiyi azalttığını gösteriyor.

Ayrıca sonuçlar, yöntemimizin algılayıcılarda diğer radyoların sinyallerinden meydana gelen karışmayı azaltıp, paket yükünü ağ içinde dengeli dağıtmaya yardımcı olduğunu ortaya koyuyor.

Anahtar sözcükler: Kablosuz Algılayıcı Ağları, Yönlendirme Protokolü, Gerçek Zamanlı İletişim, İletim Gücü Ayarlanması, Enerji Verimliliği.

Acknowledgement

I would like to express my gratitude to my supervisor Assist. Prof. Dr. İbrahim Körpeoğlu for his insights and guidance in the creation of my thesis. His encouraging and constructive approach helped me greatly throughout my research.

I would like to thank Dr. Cengiz Çelik and Prof. Dr. Hitay Özbay for kindly accepting to spend their valuable time to evaluate my thesis.

I am grateful to my family for their continuous support in everything I have done. I believe nothing would be possible without their love and encouragement.

I am also grateful to my office friends for their support and feedback on my research.

Finally, I thank the Scientific and Technical Research Council of Turkey (TÜBİTAK) for supporting this work with project EEEAG-104E028 and also for providing me with the financial support during my graduate study.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The ever existing needs for connectivity and data exchange have enabled great advancements in wireless communications. These advancements when combined with simple low-power circuit design and small-size batteries have given rise to *Wireless Sensor Networks* (WSNs) which are suitable for a broad range of applications. A WSN consists of many sensor nodes and some base stations connected via wireless links. A *sensor node* is composed of a radio component, microcontroller, power supply and sensing unit and it converts the sensed data such as temperature, humidity, movement, light, pressure, and noise to a usable format. Figure 1.1 shows the block diagram of a sensor node [3, 7, 10].

Wireless sensor networks combine sensing the environment, processing the sensed data and communication facilities of a large number of nodes and form

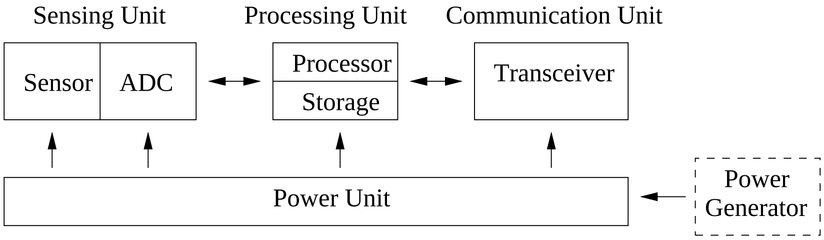


Figure 1.1: The block diagram of a sensor node.

a collaborative effort. A sensor network usually has one or more base stations (or sinks) which may control the network or serve as a gateway between other networks and the sensor network. Communication between a sensor node and the sink is accomplished by multi-hop routing. Sensor nodes can easily be embedded to a physical environment in large numbers and their deployment does not need to be pre-determined. These features make sensor networks suitable for reliable monitoring and analysis of different environments. Some application areas for sensor networks are industrial control and monitoring, home automation and consumer electronics, security and military sensing, asset tracking and supply chain management, intelligent agriculture, and health monitoring [7].

Most of the applications mentioned above require low bandwidth and do not have strict delay requirements. Recently, the availability of inexpensive CMOS camera and microphone sensors which can capture multimedia content has led the development of *Wireless Multimedia Sensor Networks* (WMSNs). These networks enable retrieval of audio and video streams, and processing and fusion of the data in real-time. Wireless multimedia sensor networks will extend the limits of environmental monitoring and tracking, and also lead to many new application areas, some of which are [2]:

- Multimedia surveillance sensor networks: Video and audio sensors can monitor an area or event and extend the capabilities of surveillance systems by using the features of a collaborative network.
- Storage of potentially relevant activities: Sensors can detect and record activities in case of events such as theft and car accidents.
- Traffic avoidance, enforcement and control systems: Sensor networks can enable monitoring the traffic and congestion of roads so that driers can have immediate guide for routes that are not crowded. Additionally, multimedia sensors can keep track of available parking spaces and give automated parking advice.
- Environmental monitoring: Some applications might necessitate time critical data from video and audio sensors for monitoring the rapid changes in

the environment. For instance, oceanographers use video sensors to determine the evolution of sandbars via image processing.

- Industrial process control: Multimedia sensors can collect and process real-time data such as temperature, pressure or images from the manufacturing process and provide automated systems. For example, quality control systems can detect a defect in a product by the help of multimedia sensor networks.

A sensor node is subject to unique constraints such as finite battery power, limited computational capability and small memory. Sensor nodes use wireless channels and broadcast communication which cause lossy links and limited bandwidth. Wireless medium is subject to issues like high path loss, channel fading, interference and noise disturbance which cause channel capacity and delay to vary continuously. The ad hoc deployment of sensors and frequent changes in topology due to wireless channel conditions necessitate sensor networks to be self-organizing and adaptable to rapid changes [2, 29].

Sensor networks are data centric and thus data delivery models constitute a major part in energy requirements. The data delivery model of a sensor network can be continuous, event driven, query driven or hybrid. Continuous models send data periodically while event and query driven systems wait for an event or query to start data transmission. Hybrid systems combine continuous and event or query driven models. Additionally, densely deployed sensors cause data redundancy in the network which makes data aggregation a desired property for sensor networks [2, 29].

Power control and topology control are two of the mechanisms that WSNs use to extend the lifetime of the network. Power control reduces energy consumed by the radio by adapting the transmission power. Topology control mechanisms deploy sleep schedules to keep a subset of nodes awake at a certain time and others at sleep to save battery power [29].

Design of sensor networks are influenced mainly by the factors mentioned above, however, WMSNs demand a certain level of *Quality of Service* (QoS) which impose new factors. Some of these factors are summarized as follows [2]:

- Application-specific QoS requirements: The broad range of applications of WMSNs will have a variety of requirements. These requirements can be combinations of bounds on delay, energy consumption, reliability, network lifetime and distortion [2].
- High bandwidth demand: Multimedia data from video or audio sensors require higher amount of bandwidth than currently supported data rates.
- Multimedia in-network processing: Raw sensor data can be processed to extract relevant and necessary information before it is disseminated in the network. This necessitates distributed, collaborative and resource-constrained architectures. In-network processing can also increase scalability by reducing data redundancy.
- Power consumption: Sensor nodes have limited power supplies and thus power consumption is a serious concern in all WSNs. Multimedia applications require high bandwidth and extensive processing, so both radio communication and data processing require more energy. This makes power consumption more important for architectures and protocols that aim to extend network lifetime for WMSNs.
- Flexible architecture to support heterogeneous applications: Since WMSN architectures may have to support heterogeneous systems and independent applications, flexible protocols are necessary to meet all the requirements.
- Multimedia coverage: Multimedia sensors may have different coverage paradigms when compared with traditional sensors. Different factors such as a video sensor's view point and orientation require development of new coverage models.

Real-time applications have certain QoS requirements primarily focusing on strict end-to-end delay, bandwidth and jitter guarantees. Additionally, real-time traffic can have multiple priorities. For example in case of video streaming, packets containing the intra-frames (I) have the highest priority since the application has the lowest tolerance for delayed I frames. The predictor frames (P) or the bi-directional (B) frames have a lesser priority when compared to I frames because the application can recover from some delays in P and B frames. Hence, priority based scheduling of real-time data is important to meet the delay and reliability requirements [29].

Different characteristics of wireless medium such as path loss and channel fading makes multi-hop communication a favorable choice since it is economical and flexible. However, in some cases, multi-hop communication may introduce more delay, interference, packet loss and error as the number of hops increases. This can affect real-time communications because delay, interference and packet losses will make QoS requirements harder to accomplish.

In a sensor node, the majority of the power is consumed by the radio component. In general, power control mechanisms adapt the transmission power of a sensor to enable efficient use of energy. The energy needed for transmission changes according to the distance to receiver, and the path loss of radio transmission scales with distance in a greater-than-linear manner. Consequently, the energy required for transmission can be decreased by dividing a long distance into shorter ones, via multi-hop communication [17].

We can extend the use of power adaptation and use this paradigm to adjust the distance between a sender and receiver to reduce end-to-end delay and interference in order to support QoS requirements of real-time communications. The requirements of real-time applications vary according to application specifications and traffic types. Especially timeliness requirements of different priority packets may differ considerably. Therefore packet differentiation is essential for meeting the deadline requirements. We want to support real-time communications by using a routing protocol which supports packet scheduling and gives precedence to urgent packets. The routing protocol also utilizes transmit power adjustment in order to meet the deadline of urgent packets and save energy by reducing transmit power when possible. Additionally, increasing the radio transmission power has a negative effect on interference and we want to reduce these effects by transmit power adjustment.

There are studies in literature that deal with energy efficient routing protocols supporting real-time applications using transmit power adjustment. Most of these studies assume that sensor nodes know the locations of other nodes and make use of geographical routing. However, a localization service such as GPS [22] may not be suitable for applications operating indoors since obstacles disturb satellite communication that is necessary for the GPS system. Additionally, GPS usage requires large amounts of energy whereas sensor nodes operate on limited battery power [5].

Moreover, most of the related studies do not support packet differentiation and scheduling. If the routing protocol supports only one delay bound in the network, it may not meet all the deadlines of different priority packets. Alternatively, it will consume more energy and bandwidth resources to support the minimum delay bound of all packets for all traffic.

Our protocol, *Real-time Routing with Priority Scheduling and Power Adjustment*, aims to meet QoS requirements of applications with various types of data by using different levels of transmit power. Transmit power adjustment allows reaching further nodes when range is extended and also increasing packet reception rate in receivers. In our protocol, we send urgent packets with maximum power to minimize end-to-end delay, and packets with lower priority with reduced power to save energy. We use hop count information to estimate delay and find routes that provide necessary delay bounds for each packet.

In our protocol, we employ a distributed approach which is scalable and self-adaptive. Our protocol uses local information and does not require network-wide knowledge. Hence no power consuming localization service is necessary.

In the remaining of this thesis, we will present our routing protocol comprehensively. In Chapter 2, we will give background information on wireless sensor networks and explain the characteristics that affect our design, such as energy consumption. Then we will give information about systems that employ power control and give examples of such protocols and their properties. After this part, we will discuss real-time support in routing protocols and present a literature review of studies related to our work. Having talked about the basics of sensor networks and related studies, we will begin describing our approach in Chapter 3. First we will introduce our design objectives and then analyze the effects of transmission power adjustment by presenting some experimental results. Following this discussion, we will explicate our protocol design and give detailed information about the components and steps. Then, in Chapter 4, we will demonstrate the performance of our approach by presenting simulation results and discuss the outcomes. Lastly, we will complete the thesis with concluding remarks and future work discussion.

Chapter 2

Background and Related Work

In this chapter, we will first give some background information about the sensor network technology and the features of sensor motes which affect design of power aware protocols for wireless multimedia sensor networks, such as properties of radio component.

After the background information, we will briefly mention the studies about energy efficient routing and power control mechanisms. Following this discussion, we will give a review of related works in the literature which support real-time applications and provide QoS guarantees.

2.1 Background Information

Sensor nodes are devices that can capture the attributes of a given phenomenon via the sensing unit and process these attributes to obtain meaningful data. Then, sensors send information from their sensing area to sink when they are requested. Sensors communicate via their low frequency radios and since the communication range of sensors is limited they use multi-hop routing to reach to the sink. The communication architecture for sensor networks is shown in Figure 2.1.

The features of sensors vary according to the requirements of application.

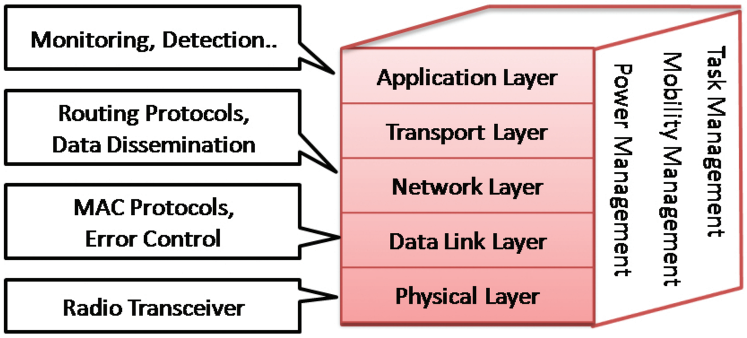


Figure 2.1: Protocol stack for wireless sensor networks.

Sensors can be equipped with various sensing units such as cameras and also localization services such as GPS. Some popular examples of current generic sensor platforms are Mica2, MicaZ, TelosB, and Firefly. Parameters of these platforms are shown in Table 2.1 [11, 12, 13, 4].

2.1.1 Energy Consumption

Sensor nodes are equipped with a limited power source and replacement of power resources is infeasible in most applications. The network lifetime depends on the limited battery power of sensor nodes. Therefore, minimizing the energy consumption of sensor networks is a key point and a challenging design problem. The energy consumption is related to the operations of three units of the sensor node which are sensing unit (sensing transducer and A/D converter), communication unit (transceiver radio), and computing/processing unit [10, 3].

Sensing unit is responsible for capturing the attributes of physical environment by doing physical signal sampling and converting into electrical signals. The energy consumed in this part depends on the hardware and application and it constitutes a small part of total energy consumption [10, 3].

Computing unit in a sensor node is a processor with memory which can control and operate the sensing, computing and communication units. The majority

Sensor Node	Microcontroller	Transceiver	Memory	OS Support
Mica2	ATmega 128L	Chipcon CC1000, 868/916MHz, 19.2Kbps	4K RAM,128K Flash	TinyOS, SOS, MantisOS
MicaZ	ATmega 128TI	Chipcon CC2420, 2.4GHz, 250Kbps	4K RAM, 128K Flash	TinyOS, SOS, MantisOS, Nano-RK
TelosB	TI MSP430	Chipcon CC2420, 2.4GHz, 250Kbps	10K RAM, 48k Flash	Contiki, TinyOS, SOS and MantisOS
FireFly	ATmega 1281	Chipcon CC2420, 2.4GHz, 250Kbps	8K RAM, 128K Flash, 4K EEPROM	Nano-RK RTOS

Table 2.1: Parameters of some generic sensor platforms.

of the energy consumed depends on the total capacitance switched by the computation and supply voltage. Energy expenses in data processing are much less compared to data communication.

Energy consumed for communication constitutes the main part of energy expenditure when compared with other functions. Radio transceiver uses up energy in transmitting, receiving and idle listening states, while transmitting being the most energy consuming state. The amount of energy necessary for transmission depends on the characteristics of radio transceiver, transmission range and packet bit length. Receiver energy does not change according to the message length and distance, and it depends only on transceiver hardware.

Radio signals fade in a greater than linear fashion as distance increases due to path loss and therefore a drop in transmission energy consumption is possible when a long distance is broken down into smaller distances. Radio transceivers support adjusting the transmission power and hence the communication range which enables controlling the energy use [10, 17]. The energy consumption values for the Chipcon CC2420 radio transceiver can be seen in Figure 2.2 [6, 9].

2.2 Related Work

Sensor nodes are densely deployed either inside or near the physical environment that will be sensed. As the routing algorithms proposed for traditional wireless ad hoc networks do not meet the different requirements of sensor networks, special multi-hop wireless routing protocols are needed to establish the communication between sensor nodes and the sink. In this part, we will give information about power control mechanisms for network layer and real-time supporting routing protocols.

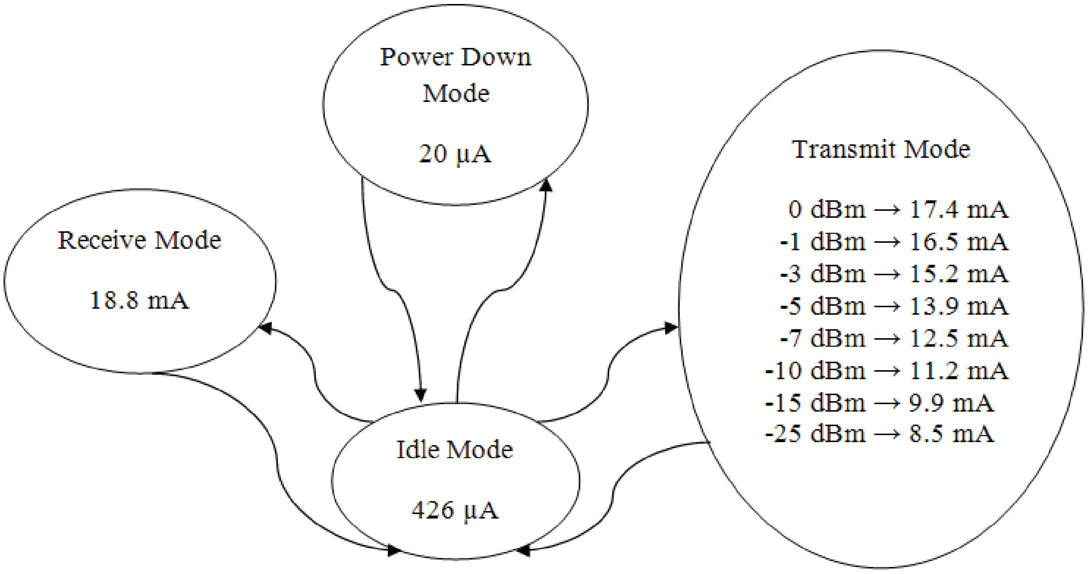


Figure 2.2: Energy consumption values for the Chipcon CC2420.

2.2.1 Power Control Mechanisms

There are many studies that try to optimize performance by adapting radio transmission power. The common idea is to find how many neighbors each node has and vary the transmission power of each node so that the number of neighbors stays within desired range. The neighbor selection method used by the previous studies base their selection on usually connectivity, packet-reception-rate (PRR), or received-signal-strength (RSS). These works aim to improve either throughput or power consumption [23].

In LINT/LILT [32] a node keeps a neighbors list in which neighbors with RSS values higher than a threshold are stored. Then, it adapts radio transmission power if the number of neighbors is outside a preset limit.

In LMA/LMN [25] a node selects its transmission range by counting the number of nodes that acknowledge its beacon message. In the algorithm that is proposed in [14], the neighbor selection is based on the RSS values. Each node ranks other nodes by their RSS values and then selects the top neighbors according to a predetermined number. The radio power is adjusted so that only these chosen neighbors are in communication range.

In PCBL [34], the nodes find the PRR of their neighbors and blacklist the ones that have very low PRR values. Then for each neighbor node the transmission power is minimized while ensuring that PRR is above a threshold.

ATPC [27] proposes a system in which each sensor node maintains the link quality information for neighbors, and adapts radio transmission power for each neighbor independently.

COMPOW [30], a power control protocol proposed for ad hoc networks, aims to optimize power control by establishing the minimum common power level which will keep the network connected and minimize the energy consumption. In CLUSTERPOW protocol [24] there are different power levels and each node runs a routing protocol at each power level. So, a routing table is constructed for each power level. When a node forwards a packet to a destination, this node consults the lowest power routing table in which that distance is present. Then the node forwards the packet at that routing table's power level to the next hop indicated by the routing table [24].

2.2.2 Real-time Support in Routing Protocols

Routing layer is important for real-time applications when providing QoS support because it finds the routes which meet the end-to-end delay requirements, use energy efficiently, and also stay stable. Moreover, the routing layer provides a transition between MAC layer and application layer since it can exchange performance parameters [29]. Real-time applications have extensive requirements while wireless sensor networks have scarce resources. Hence supplying hard real-time guarantees is very difficult for the routing layer. However, providing soft or probabilistic real-time guarantees can be accomplished by routing protocols. We will review some of the routing protocols which are QoS aware and have support for real-time applications.

The SPEED protocol [19] provides three types of real-time communication services, namely, real-time unicast, real-time area-multicast and real-time area-anycast. It uses geographical location for routing and it takes into account timely delivery of the packets. The protocol supports soft real-time communication based on feedback control and stateless algorithms. It is specifically tailored to be a stateless, localized algorithm with minimal control overhead. End-to-end soft real-time communication is achieved by maintaining a desired delivery speed across the sensor network through a combination of feedback control and non-deterministic geographic forwarding. The core module is the stateless non-deterministic geographic forwarding which sends packets to the downstream node capable of maintaining the desired delivery speed. If there is no neighbor node which can support the desired speed, it probabilistically drops packets to regulate the workload. At the same time, a back pressure packet is used for re-routing around large-delay links. Back-pressure re-routing aims to reduce or divert the traffic injected to a congested area. A desired network wide speed is maintained such that soft real-time end-to-end delivery is obtained with a theoretical delay bound [2, 19, 26].

MMSPEED [16] is an extension over SPEED, which supports service differentiation between flows with different delay and reliability requirements. It is based on a cross-layer approach between network and the MAC layers. For delivery timeliness, multiple network-wide packet delivery speed options are provided for different traffic types according to their end-to-end deadlines. Probabilistic multi-path forwarding is used while supporting service reliability in order to control the number of delivery paths based on the required end-to-end reaching probability. The mechanisms for QoS provisioning are intended to be achieved in a localized way without global network information. Localized geographic packet forwarding is supplemented with dynamic compensation, which compensates for local decision inaccuracies as a packet travels towards the destination. The important aspect is that MMSPEED tries to guarantee end-to-end requirements in a localized way and supports service differentiation. However, both SPEED and MMSPEED does not take into consideration the energy efficiency of the operations.

RAP [28] is another geographical routing protocol which proposes a real-time communication architecture for large-scale sensor networks. Sensing and control applications interact with RAP through a set of queries and event services. Communication is supported by network components including a transport-layer Location Addressed Protocol (LAP), a Geographic Forwarding (GF) routing protocol, a Velocity Monotonic (packet) Scheduling (VMS) layer, and a prioritized MAC. VMS is a deadline-aware and distance-aware packet scheduling algorithm which relates a packet's priority to its deadline and its distance from the destination. RAP protocol uses local urgency or requested velocity. This way, a packet must continue towards its destination with the determined velocity in order to meet its deadline. VMS differentiates packets according to their required velocity and hence improves deadline miss ratio.

In [1], an energy-aware QoS routing protocol which can find energy-efficient paths for best-effort traffic is proposed. They assume each node can classify the type of incoming packets and distribute real-time and non-real-time traffic to different priority queues. In this protocol, the delay requirement is converted to bandwidth requirement. This approach does not consider the delay that occurs due to channel access at the MAC layer. Additionally, the class-based priority queuing system is too complicated and costly for wireless sensor networks.

In [31], the authors present a heuristic solution for the problem of finding energy-efficient paths for traffic with delay bounds. They employ topology control for sensor networks and they propose a network architecture and a routing framework. They have a modeling of contention delay caused by the MAC layer. A set of paths between source and sink nodes are identified and indexed in the increasing order of their energy consumption. Then, the end-to-end delay is estimated along each of these identified paths. The path that has the lowest index and also satisfies the delay bound is selected. This solution assumes that nodes are equipped with two radios. One of them is a low-power radio and it is for short-range communication. The other one is a high-power radio for long-range communication which can reach to the sink node directly. These assumptions might not be feasible and energy-efficient.

The authors present a routing algorithm in [15] that maximizes the lifetime of a sensor network in which all data packets are destined for a single collection node. They formulate the lifetime maximization as a linear programming (LP) problem by excluding the delay constraint in order to determine optimal routing paths and maximize the minimum lifetime of each node in the network. They implement the solution of this problem in a centralized way and then approximate it by an iterative algorithm based on least cost path routing. After that, the delay constraint is introduced and the length of routing path from each node to the sink is limited according to delay bound. The simulation results show that they achieve to limit the maximum delay to a certain level. On the other hand, this does not guarantee that the solution can be flexible to meet application specified delay bound generally.

RPAR [8] is a real-time power-aware routing protocol which is proposed to achieve application specific communication delay at low energy cost. The routing protocol dynamically changes routing decisions and adapts the radio transmission power according to these decisions. The delay bounds are specified by the application as deadlines for each packet so that the application handles the trade-off between energy and delay. The algorithm employs geographical routing and forwards packet to a neighbor which is closer to the sink. For each packet, a required velocity is computed according to the distance between the node and sink and also the packet's deadline. The neighborhood manager finds energy-efficient forwarding choices which can support the packet's required velocity. The delay estimator is responsible for estimating the delay of forwarding choices. It takes into account the retransmission rate of forwarding choices. When there is an eligible forwarding choice, the neighborhood manager decreases radio transmission power for energy efficiency. If no eligible forwarding choice is found, the neighborhood manager increases the radio transmission power to increase the velocity by reducing the number of retransmissions. If the required velocity is not supported by current neighbors, it tries to discover new neighbors. This solution increases the number of packets that meet their deadline while reducing the transmission energy. However, it assumes the nodes are equipped with a localization service such as GPS [22] which consumes large amounts of energy and therefore is not recommended for wireless sensor networks. The energy cost of localization service is not considered in the computations. Moreover, since GPS uses satelite communication, it may not be available in indoor environments or areas surrounded with obstacles. Hence, GPS usage is not practical for applications with indoor settings [5].

As we review the related studies we see that power control is widely used in sensor network protocols in order to improve network performance in terms of energy efficiency and throughput. Routing protocols that support real-time applications also benefit from power control. Most real-time routing protocols employ geographical routing and packet scheduling with different approaches. In the next chapter, we will describe our approach which uses routing trees instead of geographical routing and utilizes transmit power adjustment in a different way.

Chapter 3

Proposed Routing Protocol

Many wireless sensor network applications require real-time communication and real-time communication necessitates packets to reach destination on time. Our protocol aims to provide soft real-time guarantees for applications while employing efficient use of energy and network resources. Applications can have packets with different priorities, and some packets may not have as strict deadlines as the others. Therefore, our protocol supports packets with tight deadlines and uses the resources of a node generously for such packets. On the other hand, while sending less urgent packets, only sufficient amount of these resources are used. We achieve efficient use of energy and increased network capacity while providing soft real-time guarantees by utilizing transmit power control.

In this section, we will explain the details of our proposed protocol. We will start with presenting our design objectives and afterward, we will elaborate the effects of transmit power adjustment on end-to-end delay and energy consumption. After this discussion, we will describe the design of our routing algorithm in detail.

3.1 Design Objectives

In this part, we will explain some design goals for our Real-time Routing with Priority Scheduling and Power Adjustment routing scheme.

3.1.1 Delay Bounds

In case of real-time communications, the delay bounds for packets are very strict and thus we aim to reduce the end-to-end delays that packets endure. The applications determine the delay requirements for packets and our routing algorithm tries to find the routes that can meet these requirements.

3.1.2 Packet Differentiation

Additionally, real-time traffic can have multiple priorities. Different types of applications might request diverse delay requirements from the routing layer, or one application might have different priority packets. Hence, scheduling of real-time data according to priority is necessary to meet the delay deadlines. We aim to differentiate packets according to their priorities which are defined by the application. This way we can also utilize network resources better.

3.1.3 Energy Consumption

The radio component is usually the most energy consuming unit of a sensor node. Power consumption of the radio has three sources: power consumed by the transmitter electronics, power consumed by receiver electronics and the power consumed by the power amplifier to transmit a packet at the actual power level in the medium. If the energy consumed for transmission dominates other components, then efficient use of energy becomes directly proportional to the power level of transmission [24].

3.1.4 Network Capacity

Since wireless channel is a shared medium, transmissions cause interference at the nodes in communication range. The area of interference can be reduced if the range of the transmission is reduced, and this requires power to be adjusted to a lower level. On the other hand, if transmit power is reduced, then packets will be routed along an increased number of shorter hops. More hops mean more sensor nodes relaying traffic.

If we assume that transmission range is d, then the area of interference becomes proportional to d^2. Also if transmission range is d, then the number of hops becomes inversely proportional to d. The whole area interfered by a packet transmission is the number of hops multiplied by interference range of these hops, which becomes proportional to $d^2 \times 1/d = d$. Consequently, smaller d means increased network capacity and reducing transmit power level will increase network capacity. Hence, we need to adjust transmit power in order to optimize network capacity [24, 18].

3.2 Preliminary Analysis on Power Control

Power control problem deals with selecting the appropriate transmit power level for each packet at each node, in a distributed manner. This is a complex problem because the selection of transmit power level influences many aspects of the process of the network. Transmit power level [24]:

- specifies the link quality between sender and receiver.
- specifies the range of transmission.
- determines the level of interference caused to other receivers in range.

Consequently, transmit power level is one of the definitive factors for the performance of the system. Its effects on the performance can be summarized as follows:

- The connectivity of network and the delivery probability of a packet to its destination depend on transmit power level.
- The throughput capacity of a network is affected by the transmit power level [24, 18].
- Transmit power affects the contention for the medium.
- Power control influences the number of hops, which in turn affects end-to-end delay.
- Transmit power control also affects the energy consumption of nodes in the network.

Multi-hop transmission enables energy efficiency and increased network lifetime in wireless sensor networks. However, the queuing and processing delays introduced on each intermediate node may cause an increased delay. As the number of hops increases, the end-to-end delay is also expected to increase therefore there is a tradeoff between energy and delay. It is the job of routing protocol to find an optimal point between the number of hops and delay requirement in order to provide delay guarantees. Our routing protocol is founded on this concept, also known as the *energy-latency tradeoff*. In our algorithm, we utilized transmit power adjustment to strike a balance between resource consumption and delay.

In the next two sections we will present some delay measurements for different transmit power levels and a simple analysis of the effects of transmit power control on end-to-end delay and energy consumption.

3.2.1 Experiments on Transmit Power and Delay Relationship

In order to analyze the effects of transmit power on end-to-end delay we conducted some experiments on Mica2 motes. The motes have a Chipcon CC1000 radio transceiver which operates at 868/915 MHz and has an outdoor range of 152 m. CC1000 radio allows transmit power adjustment between -20 dBm and 5 dBm. The data rate for Mica2 motes is 38.4 Kbaud (19.2 Kbps) and they run TinyOS, an open-source operating system for wireless sensor networks. In the experiment, B-MAC, the default MAC protocol adopted by TinyOS is used as the MAC protocol [11].

For this experiment, we placed 9 sensor motes in an office environment, along a corridor. We used the sensor mote which was connected to a PC and placed at one side of the corridor, as the source. This source mote generated packets that are destined for the mote at the other end and transmitted them with power levels changing from -20 dBm to 5 dBm. We used a shortest hop routing scheme such that motes forward the packets to the outmost mote in range. So, each mote selected the next hop according to the chosen power level and the number of hops between the source and destination changed accordingly. We ensured that all resulting routes maintained a packet reception rate of at least 75%. The destination mote that is at the end of the corridor reversed the direction of the packet and sent it back to the source along the same route. End-to-end delays were computed from the round trip time of packets. In each power level we had two runs and in each run the source sent 50 packets at a rate of 1 packet per second. For the first experiment, we positioned each sensor approximately 9 m away from each other and we used all the power levels from -1 dBm to 5 dBm. Power levels lower than -1 dBm could not preserve the 75% packet reception rate for 9 m distance so we conducted a second experiment with a shorter distance. For the second experiment, we positioned the motes with 4.5 m intervals and used power levels between -11 dBm and -2 dBm and also 5 dBm. As the transmit power level changes, the resulting average end-to-end delays for the first and second experiments are shown in Figure 3.1 and 3.2 respectively.

When transmit power increases, two scenarios are possible in the shortest hop routing scheme we used: either the link quality between sender and receiver will increase, or sender will reach to a farther node and shorten the number of hops. So when the transmit power is adjusted, different values of delay are possible. The results confirm our expectation and indicate that increasing transmit power

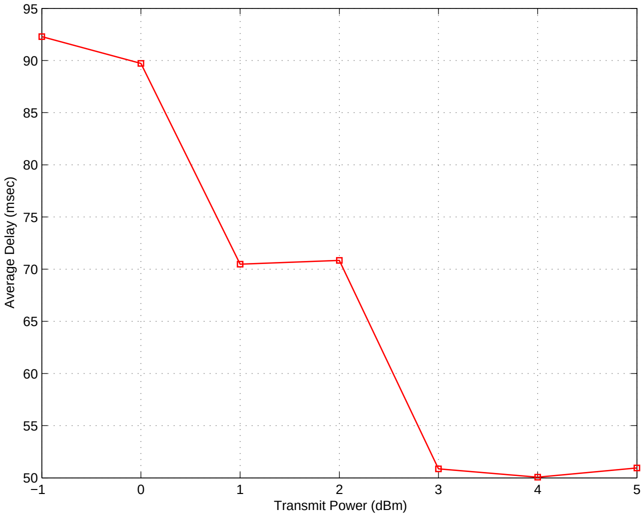


Figure 3.1: Transmit power vs. average end-to-end delay for the first experiment.

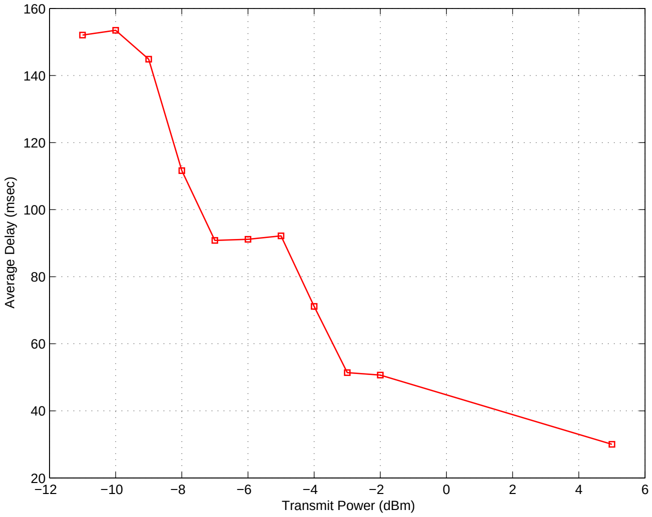


Figure 3.2: Transmit power vs. average end-to-end delay for the second experiment.

can help reducing end-to-end delay.

3.2.2 Simple Analysis Relating Transmit Power vs. Delay and Energy

In order to examine the delay and energy consumption in relation with transmit power for a system, we modeled an ideal network and assumed that the nodes can change transmit power from 1 to 50 mW with a step size of 1 mW. We suppose the threshold for received signal (*receptionLimit*) is -43 dBm (5×10^{-5} mW) which enables a range of 1000 m when transmit power is 50 mW and $\alpha = 2$. We use the signal attenuation function shown in Equation 3.1 as radio propagation model and suppose the transmission is successful if received power is greater than the threshold ($P_{RX} \geq receptionLimit$).

$$P_{TX} \times 1/(1 + d^{\alpha}) = P_{RX} \tag{3.1}$$

We suppose that a finite number of nodes are uniformly distributed in a circular area so that there are exactly n nodes in 1 m^2. The sink is located at the center of this circle which has a radius (R) of 10000 m. We compute the appropriate range d for each transmit power level, and therefore the area is divided into different levels of circles, each with a width of d. We assume each node except sink, injects one packet into the network. The nodes forward packets to their parents and packets reach to the sink by going through a number of hops which change according to the level of the data generating node. We suppose that a node sends the packet it generated to a node at the border of inner level and only the nodes located at the border of levels forward packets until the sink node is reached.

If R is the radius of the area and d is the range for the selected power level, then the number of tree levels (L) in the network for that power level will be $L = R/d$. If each node generates one packet, the total number of packets will equal to the total number of nodes:

$$totalNoOfPackets = \pi R^2 n \tag{3.2}$$

Then, the total number of point-to-point transmissions (including forwarding of packets) will be:

$$totalNoOfTransmissions = \left(\pi d^2 n\right) \times \left[\sum_{k=1}^{L} \left(2k^2 - k\right)\right] \tag{3.3}$$

We consider only the transmission energy for the energy consumption and use the energy model from [20]. If the packet length is l then the function of transmission energy (E_{TX}) with respect to range (d) is computed according to [21]:

$$E_{TX}(d) = l \times (E_{elec} + \epsilon_{amp} \times d^{\alpha}) \tag{3.4}$$

$$E_{elect} = 50 \times 10^{-6} mJ \tag{3.5}$$

$$\epsilon_{amp} = 100 \times 10^{-9} mJ/m^2 bit \tag{3.6}$$

The nodes generate packets and send them to their parents and then packets are forwarded until the sink. If we consider only the energy consumed for transmission, then the total energy is the sum of total energy consumed for generated packets ($E_{generatedPackets}$) and total energy consumed for forwarded packets ($E_{forwardedPackets}$):

$$E_{generatedPackets} = \sum_{k=0}^{L-1} \int_{kd}^{(k+1)d} \left[(2\pi x dx) \times E_{TX}(d) n\right] \tag{3.7}$$

$$E_{forwardedPackets} = E_{TX}(d) \times \left[\left(\sum_{k=1}^{L} \left(2k^2 - k\right) \times \pi d n^2\right) - \pi R^2 n\right] \tag{3.8}$$

Then the average energy per packet is computed by the sum of $E_{generatedPackets}$ and forwarded packets $E_{forwardedPackets}$ divided by the $totalNoOfPackets$.

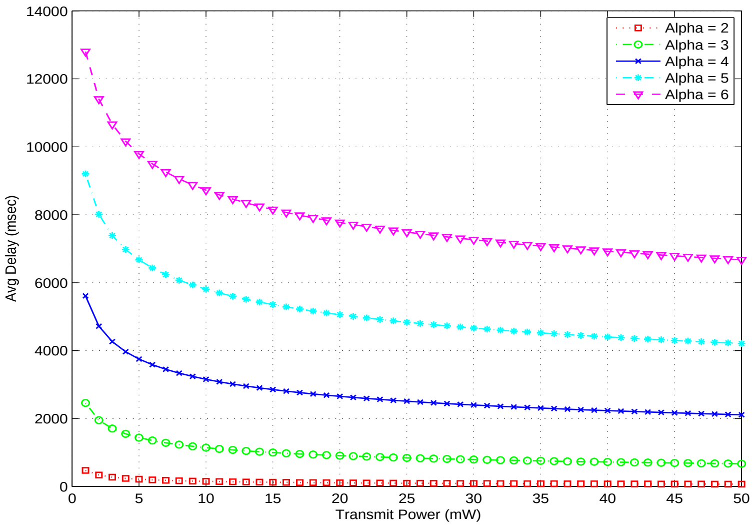


Figure 3.3: Transmit power vs. average delay when path loss exponent changes between 2 and 6.

In order to compute the average delay per packet, we first find the delay introduced in one hop and multiply it with the total hop count. Then we divide it by the total number of packets. We suppose that a packet experiences a delay t in one hop. Total number of hop counts for all packets is equal to the total number of point-to-point transmissions. Then the average delay per packet becomes:

$$avgDelay = t \times \left[\frac{(\pi d^2 n) \times \sum_{k=1}^{L} (2k^2 - k)}{\pi R^2 n} \right] \tag{3.9}$$

Figures 3.3 and 3.4 show the results of these analyses when the path loss exponent (α) changes between 2 and 6 with a step size 1 and the packet length (l) is 960 bits.

As predicted, when transmit power is increased, then the delay per packet reduces while energy consumption per packet increases. As the path loss exponent increases, the end-to-end delay also increases. This is because the path loss exponent causes the received power to decrease exponentially, however the transmit power increases linearly and cannot compensate for this decrease. As a consequence, the packets have to go through more number of hops to reach to

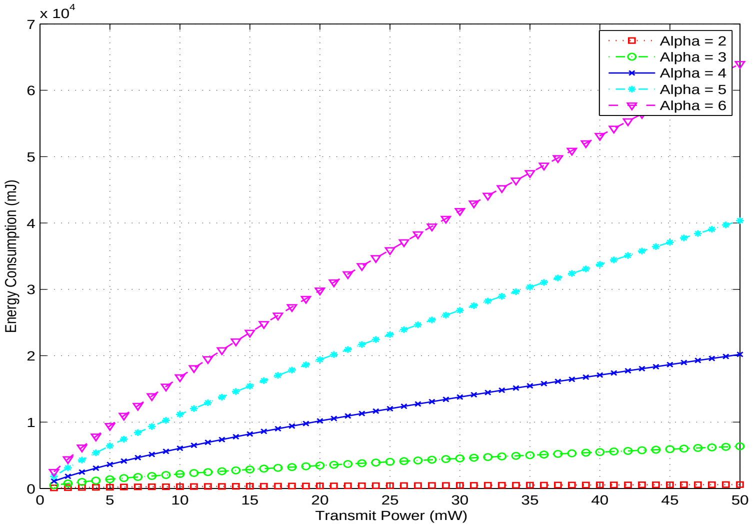


Figure 3.4: Transmit power vs. average energy consumption when path loss exponent changes between 2 and 6.

the sink. Moreover, the energy consumption increases drastically as the path loss exponent increases, which is expected (Figure 3.4). Therefore, as the path loss exponent increases, transmit power level becomes more significant for end-to-end delay and energy consumption.

Now we will explain the design of our routing protocol and describe the components of our approach in detail.

3.3 Routing Protocol Design

In this study, we assume that the sensor nodes are stationary and topology changes are only due to the failure of the nodes. We assume that sensor nodes do not have network-wide information such as topology and location. Also, we suppose that the sensor nodes are equipped with radio transceivers which can adjust transmit power, like the CC2420 radio component of TelosB motes. The CC2420 radio transceiver [9] can adjust its output power between -25 dBm and 0 dBm. The current consumption of the device also changes according to the output power as shown in Figure 2.2.

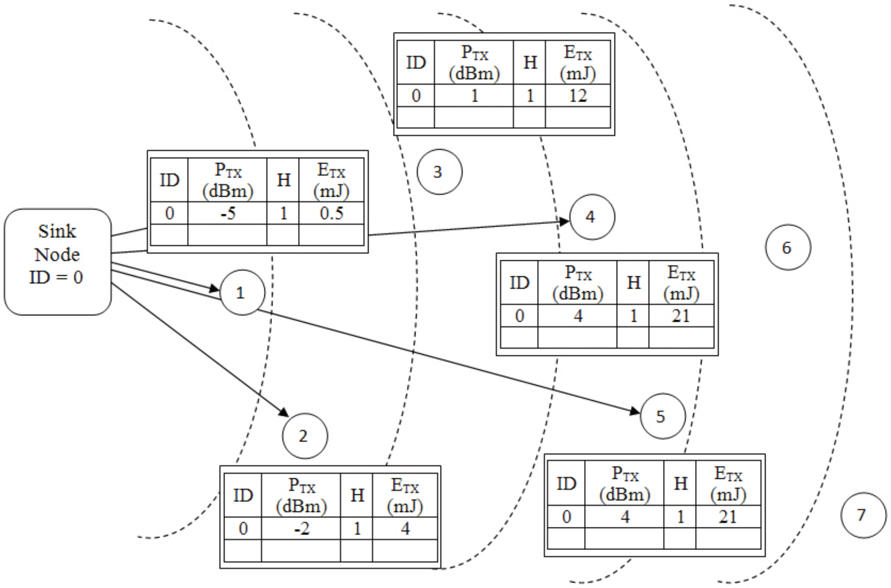
For our routing protocol, we considered only the many-to-one traffic flow. In our model, the sensor nodes send different types of real-time data towards the sink. We did not take into account the point-to-point communication between two sensor nodes.

We suppose that the real-time application assigns a delay bound to each packet when it is generated. We use this delay bound or *deadline* of the packet to determine its priority. Subsequently, the packet is forwarded to the next hop that can guarantee to deliver the packet before its *deadline*. Since the nodes do not have the coordinates of other nodes and the sink, the route should be established first. For this reason we initially employ a routing tree establishment phase. Each node finds its neighbors through broadcast messages and then selects its parents which are closer to sink in terms of number of hops.

Our routing protocol uses predetermined transmit power levels to adjust range and to change the next hop that the packet will be forwarded. For this purpose, first the routing trees for each power level are established. These trees are all rooted at the sink node. This way every node has one or more parents for each power level which can reach to the sink in different number of hops. Consequently, each node can select the appropriate parent according to how many hops it takes to reach the sink and the energy consumption of the route. We propose that, by determining the appropriate parent according to the *deadline* of the packets and energy cost, we can support real-time communications and use the resources of the nodes and the network efficiently.

3.3.1 Routing Tree Establishment

Before the nodes start disseminating packets with sensed data, they form the routing trees and determine their parents for each power level. Our protocol uses only local information for establishing routes. The nodes learn their one hop neighbors for each power level via message exchange.



ID	P_{TX} (dBm)	H	E_{TX} (mJ)
0	1	1	12

ID	P_{TX} (dBm)	H	E_{TX} (mJ)
0	-5	1	0.5

ID	P_{TX} (dBm)	H	E_{TX} (mJ)
0	4	1	21

ID	P_{TX} (dBm)	H	E_{TX} (mJ)
0	-2	1	4

ID	P_{TX} (dBm)	H	E_{TX} (mJ)
0	4	1	21

Figure 3.5: The sink broadcasts $TreeSetup$ messages.

First, all nodes should discover their neighbors in range and the required transmit power level to reach them. For this purpose, the nodes send $Hello$ messages in all available transmit power levels. A $Hello$ message contains the ID of sender and the transmit power level p of this message chosen by the sender. When a node receives a $Hello$ message it checks p of the message and if ID is not present in the table or if the power level of the previous record is greater than p, it records the ID to its $Neighbors$ table. The $Neighbors$ table of a node keeps the one hop neighbors which can reach to this node and the minimum transmit power level they use for reaching.

The links between sensor nodes tend to be asymmetric and transmit power adjustment also increases this tendency. In order to overcome the problems caused by asymmetrical links, all nodes keep the list of neighbors they hear in $Neighbors$ table and then share this information with their neighbors via the $TreeSetup$ messages.

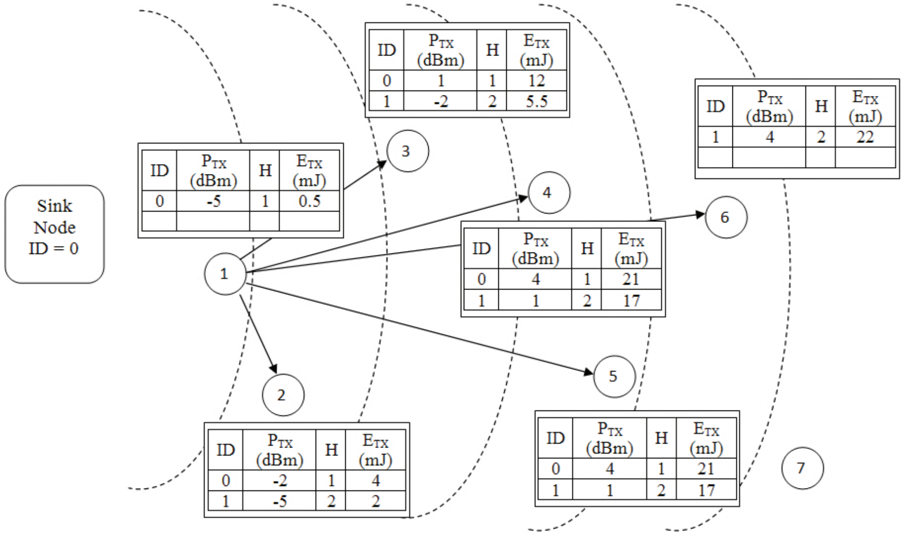


Figure 3.6: The node with ID = 1 reaches sink with minimum power, so it starts broadcasting *TreeSetup* messages first.

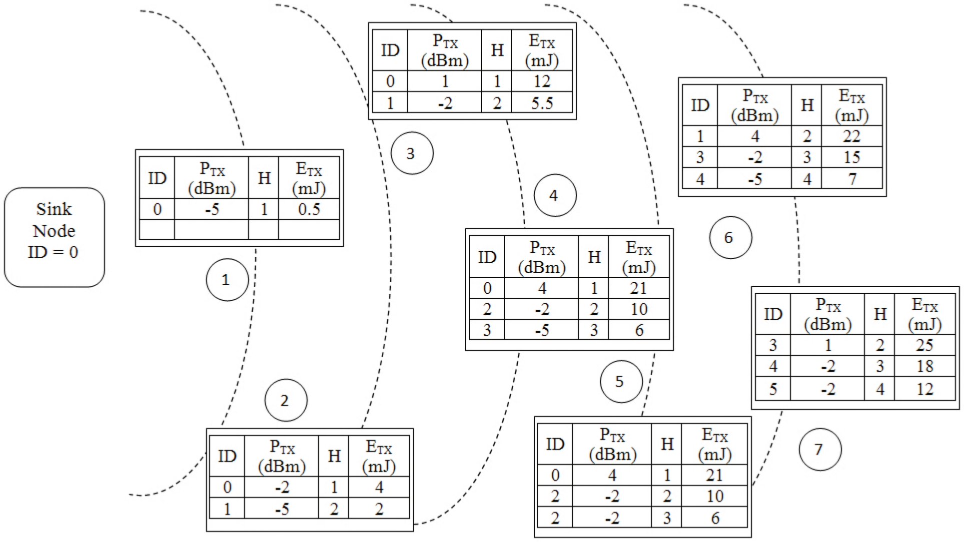


Figure 3.7: The *Parents* table for all nodes after the routing tree has been formed.

```
if node ≠ sink then
    foreach transmit power level p in powerLevels do
        Broadcast Hello messages with TTL 1;
    end
    if Hello message received then
        Record neighborID and neighborPowerLevel in Neighbors table;
        Wait for TreeSetup message;
    end
    if TreeSetup message received then
        Record sender of TreeSetup message as parent in Parents table;
        Wait for timeout according to $power_{parent}$ and $hopCount_{parent}$;
    end
    if timeout expired then
        Broadcast TreeSetup message including Neighbors table and
        Parents information;
    end
else
    if Hello message received then
        Record neighborID and neighborPowerLevel in Neighbors table;
        Wait for timeout;
    end
    if timeout expired then
        Broadcast TreeSetup message including Neighbors and tree setup
        information;
    end
end
```

Algorithm 1: Mechanism for building routing trees with different power levels all rooted at the sink.

The sink node also maintains a $Neighbors$ table and after this table is populated, it starts broadcasting $TreeSetup$ messages. The $TreeSetup$ message contains the ID and the neighbor information of the sender. Also, the information about the node's parents such as the number of hops to reach sink ($hopCount_{parent}$) and the estimated energy consumption for this route ($energyCost_{parent}$) is computed and sent with the $TreeSetup$ message. For the sink node, these values are: $hopCount = 0$ and $energyCost = 0$. The sink broadcasts this message with maximum transmit power level and with a time to live value of 1 ($TTL = 1$). When other nodes receive this $TreeSetup$ message, they first check if the sink can hear them by looking at the neighbor information in the message. If it can, the nodes record sink node and the transmit power level to reach it ($power_{parent}$) to the $Parents$ table and store $hopCount_{parent}$ and $energyCost_{parent}$ values for this parent. Following this, the nodes wait for other $TreeSetup$ messages for a $timeout$ value and record other parents. The $timeout$ is proportional to the minimum $hopCount$ value and minimum transmit power level for this $hopCount$ value. After the $timeout$ expires the nodes broadcast their $TreeSetup$ messages with maximum transmit power level. They include the $hopCount_{parent}$ and $energyCost_{parent}$ values from the $Parents$ table. This way, as the $TreeSetup$ messages propagate to the leaf nodes, the routing trees for different transmit power levels are established.

The algorithm for this mechanism is shown in Algorithm 1. Additionally, Figures 3.5, 3.6 and 3.7 illustrate an example of routing tree construction steps. The dissemination of $TreeSetup$ message is shown in 3.5, 3.6 and the resulting $Parents$ table of the nodes after tree is constructed is shown in 3.7. The $Parents$ table of the nodes are filled with example values of transmit power level for parent node (P_{TX}), $hopCount_{parent}$ (H) and $energyCost_{parent}$ (E_{TX}).

3.3.2 Packet Forwarding

Once the tree is established, each node will have the parent node IDs and corresponding transmit power levels to reach parents in its routing table. Each node will store a number of parents which provide different delay bounds and require minimum energy consumption among others. The maximum number of parents for each delay bound is specified by $maxParents$. We suppose that the number of hops to reach the sink node ($hopCount$) is directly proportional to the delay bound of the routes. Therefore, each node will have $maxParents$ parents for each $hopCount$ that is available.

When the application sends a packet, it sets the *deadline* value to a specific delay bound and our protocol uses this value to determine priority level of the packet. According to this priority, the packet is forwarded to the parent that can meet the *deadline* requirement by consuming the least energy of the network. The algorithm for selecting a parent to forward the packet is explained in Algorithm 2.

The crucial steps of the forwarding mechanism are the delay estimation for one hop, selection of the parent, and updating the *deadline* properly according to the progress the packet has made.

- Delay Estimation: We presume the determining factor for the end-to-end delay of a packet is the number of hops this packet traverses. The delay at intermediate nodes is caused by processing, queuing, contention, transmission and propagation delays. In order to estimate the total end-to-end delay, we estimate the delay at one hop and multiply it with the number of hops between source and destination. We assume that the propagation and processing delays at intermediate hops do not change considerably and initially the queuing and contention delays are very small since traffic load is light. We assume that the nodes are not synchronized with each other or with the sink. Therefore, we find the approximate delays from the round trip time of the packets. We initialize the one hop delay ($delay_{1hop}$) to an approximate value based on the transmissions in the routing tree establishment phase. Then the $delay_{1hop}$ value is updated by the round trip time of any packet sent and its acknowledgment.
- Parent Selection: When a packet is generated, the maximum number of hops that can support the packet's *deadline*, i.e., the required number of

hops ($hopCount_{req}$) is computed as: $hopCount_{req} = deadline/delay_{1hop}$. If the source node have parents that can provide a route with the computed $hopCount_{req}$, then one of them is selected as the forwarding parent.

In the routing tree establishment phase, the nodes establish routes for each number of hops available by evaluating the energy consumption of the routes. Therefore, the parents of a node are the ones that provide minimum energy routes. However, when the $maxParents$ is more than one, the forwarding parent must be selected among these parents. If persistently the parent with minimum energy consumption is selected, then this will drain the chosen node quickly. Since this may disturb the connectivity of the network, we try to balance the load on the nodes. Initially, the forwarding parent is selected randomly and as the node relays packets, some feedback is gathered from the transmissions. The ratio of successful transmissions is recorded, and information such as the remaining energy of parents, traffic load on the parents and the number of interfered nodes are obtained from the acknowledgment packets. Then the next hop is selected both according to this information and randomly. After the next hop is selected, the transmit power level is adjusted according to the required transmit power (tx_{req}) that can reach to this parent.

- Updating *deadline*: The *deadline* of a packet is updated on each hop according to the progress of the packet since the last hop. The time packet spent on this hop including the contention and queuing delays is subtracted from the *deadline* before it is transmitted. This is accomplished with the help of MAC layer support.

$$deadline = deadline - (d_{process} + d_{queue} + d_{contention} + d_{tx} + d_{propagation}) \quad (3.10)$$

Packets are re-examined by their *deadline* requirement on each intermediate hop, so the priority level of a packet may change on each hop. If the packet progressed at a speed higher than required, then the next hop shifts it to a lower speed by forwarding it on a more energy efficient route with more number of hops. This way dynamic compensation is employed to packets.

```
$elapsedTime = departureTime - arrivalTime + transmitDelay$;
$deadline = deadline - elapsedTime$;
$hopCount_{req} = deadline/delay_{1hop}$;
foreach parent in Parents table do
    if $hopCount_{req} \geq parent(h)$ then
        if $parent_{remainingEnergy} \geq forwardParent_{remainingEnergy}$ then
            if $parent_{interference} \geq forwardParent_{interference}$ then
                prevForwardParent = forwardParent;
                forwardParent = parent;
            end
        end
    end
end
$p$ = random();
if $p \geq 0.5$ then
    Send packet to forwardParent;
else
    Send packet to prevForwardParent;
end
```

Algorithm 2: Selecting parent according to deadline requirement of the packet.

We assume that urgent packets also have reliability requirements. If a packet has a tight *deadline* which cannot be satisfied by the available parents, then the node forwards it to the parent that provides the minimum delay bound. Hence the packet reaches the sink node as soon as possible. Similarly, if *deadline* of a packet expires, the node forwards it with maximum speed. If the reliability requirement of a packet is not strict, then it can be dropped when the *deadline* cannot be met.

In this chapter, we have explained the problem setting we are working on and our analysis on the subject. Then we described our routing protocol in depth, and in the next chapter we will continue with the discussion of performance results.

Chapter 4

Performance Evaluation

In this chapter, we analyze the performance of our approach by discussing the simulation results of our protocol with different settings. We examined the performance of our protocol in terms of delay, deadline meet ratio, transmit energy consumption, interference and network lifetime metrics. We will start with explaining our simulation model and then present the simulation results and observations.

4.1 Simulation Model

The routing algorithm is implemented in Prowler, a probabilistic wireless sensor networks simulator which runs under Matlab. Prowler provides a generic simulation environment, and in order to observe the performance of the routing protocol, the parameters of the simulator are configured according to a typical sensor mote in an ideal environment [33].

In our simulation, we used the common and simple path loss model (Equation 3.1) and we assumed the sensor nodes can adjust the transmission power to any level according to the desired range. Since we tested the effects of different path loss exponents (α), we assumed that the nodes support transmit power levels

Simulation Parameter	Current Setting
Wireless Channel Model	Ideal Wireless Channel and No Interference
Deployment Field	100 m x 100 m
Number of nodes	100
Neighbor RSSI Threshold	-43.01 dBm
Data Rate	40 Kbps
Packet Length	120 byte

Table 4.1: Simulation parameters and settings of our experiments.

that enable them to reach a maximum range of 30 m and minimum range of 7 m for grid deployment when α changes between 2 and 6. For example, in case of grid deployment, the minimum and maximum transmit power levels we used for $\alpha = 2$ is -27 dBm and -13 dBm respectively. For $\alpha = 6$ these values grow to be 7.7 dBm and 45.61 dBm.

We implemented a simple energy model in Prowler to evaluate the energy consumption of the nodes' transmissions. We assume that the radio spends E_{elect} = 50 nJ/bit for transmitter electronics and ϵ_{amp} = 100 p/bit/m^2 for the transmit amplifier [21]. Since we consider the path attenuation, the energy spent depends on the transmit distance (d). Then the transmission energy for a packet with length k bit becomes [21]:

$$E_{TX} = k \times (E_{elec} + \epsilon_{amp} \times d^{\alpha}) \tag{4.1}$$

In order to observe the performance of our protocol in an ideal environment, we assumed an ideal MAC layer and wireless channel which enables collision free communication. The common settings for the simulations are summarized in Table 4.1.

We consider the traffic flowing from sensor nodes towards the sink node. All the sensor nodes relay packets destined at the sink node and forward other nodes' packets towards the sink node. In our experiments, the sink node is always located on the bottom left corner of the area.

We simulated a real-time application which sends packets with different priorities and assigns deadline requirements accordingly. We configured the application to send 15 types of packets with deadlines changing from 12.5 ms to 187.5 ms with a step size of 12.5 ms. We chose these values to see the performance of our protocol in case of packets with both very strict deadlines and loose deadlines.

We used the following performance metrics for our protocol:

- Delay: End-to-end delay between the source and destination.
- Energy: Energy consumption per packet which is computed by the total transmission energy for all packets divided by the number of successfully delivered packets.
- Deadline meet ratio: The ratio of packets delivered before the deadline.
- Interference: Sum of the number of interfered nodes that are interfered by another transmission in all transmissions.
- Weighted interference: Sum of the number of affected nodes multiplied by the received signal strength in all transmissions.
- Network lifetime: The time interval until the first node in the network has a predetermined remaining energy.
- Average remaining energy: The average of the nodes' remaining energy values when the first node reached to a predetermined remaining energy.

We compare the performance results of our protocol with two protocols that use fixed transmit power. First protocol uses the maximum power available and has a range of 30 m in order to send packets with minimum delay. It establishes routing trees by broadcasting setup messages and selecting parents with lower levels in the tree. Second protocol chooses lower power levels which will maintain connectivity and selects energy efficient routes. It uses the transmit power levels 5 m or 20 m according to the deployment of the nodes. For the grid deployment, the connectivity in the network could be established with 7 m range since nodes are distributed with 7 m intervals. However, when the nodes are deployed randomly, 7 m range is not sufficient. Therefore, for the minimum fixed power scheme and in case of random deployment we specified the minimum range as 20 m, which is the minimum distance that preserves connectivity.

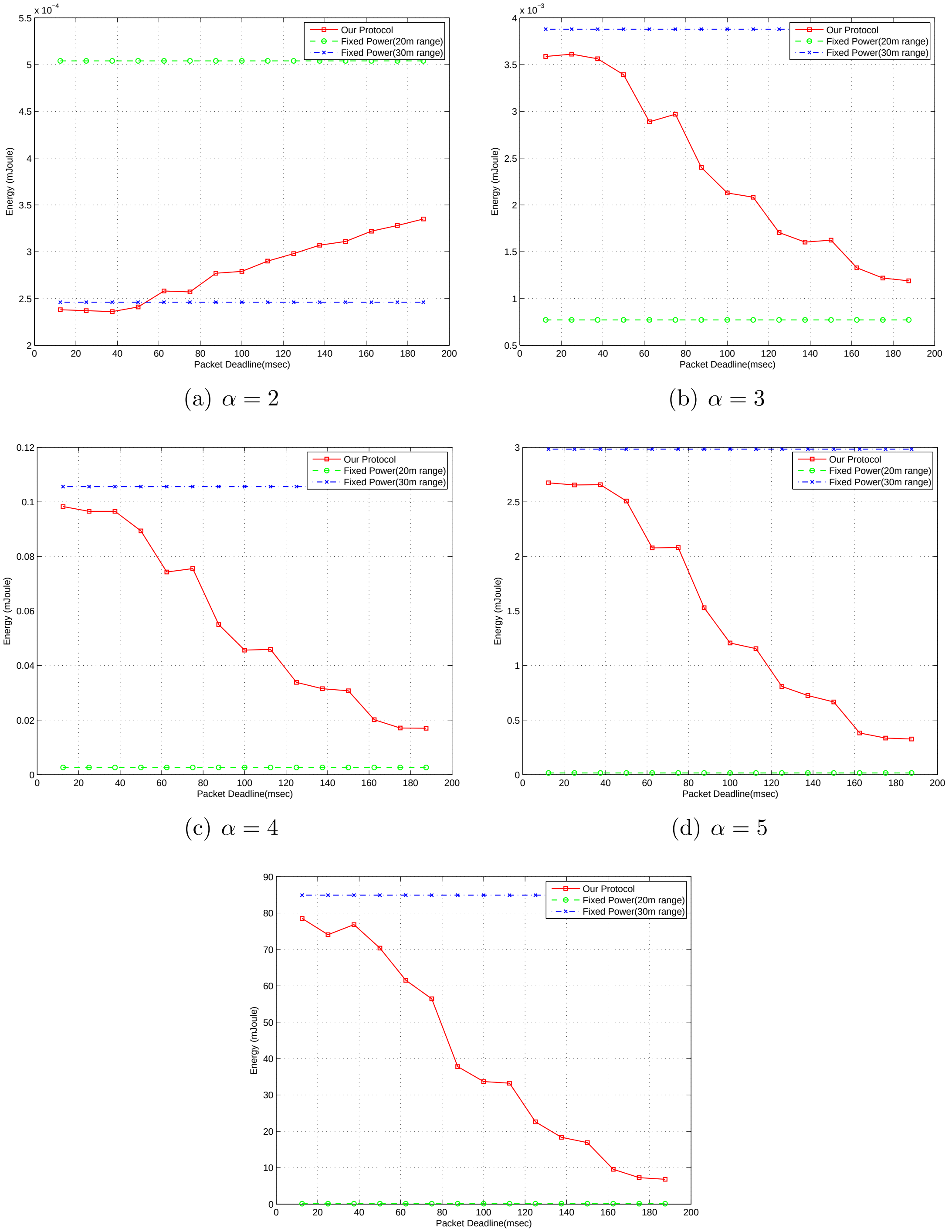
4.2 Experiments

4.2.1 Effects of Path Loss Exponent

We first conducted experiments to analyze the behavior of the protocol when the path loss exponent changes between $2 - 6$. The simulation was performed with a grid deployment of nodes and the routing trees were already constructed for $maxParents = 2$. All the nodes except sink relayed one packet per priority in a random order, with a rate of one packet per second. Figures 4.1, 4.2 and 4.3 show the resulting change in delay, energy and deadline meet ratio.

We included only the graphs for $\alpha = 2$ and $\alpha = 6$ for delay and deadline meet ratio metrics because no significant change occurs between these values. This is an expected result since range does not change when α changes in our setting, as explained in Section 4.1.

As can be seen in Figure 4.1, the energy consumption for our protocol changes drastically as α increases. For $\alpha = 2$, the energy consumption increases as the packet deadline increases, which means more transmission energy is spent for less urgent packets. In case of less urgent packets our protocol tries to divide long distances into hops with smaller distance. This way, the number of hops increases while the energy per hop decreases. If the reduction in energy per hop cannot compensate the increase due to the increase in number of hops, then transmission energy may increase as packet deadline increases. This is the reason why our protocol gives unexpected results for energy consumption when $\alpha = 2$



(a) $\alpha = 2$

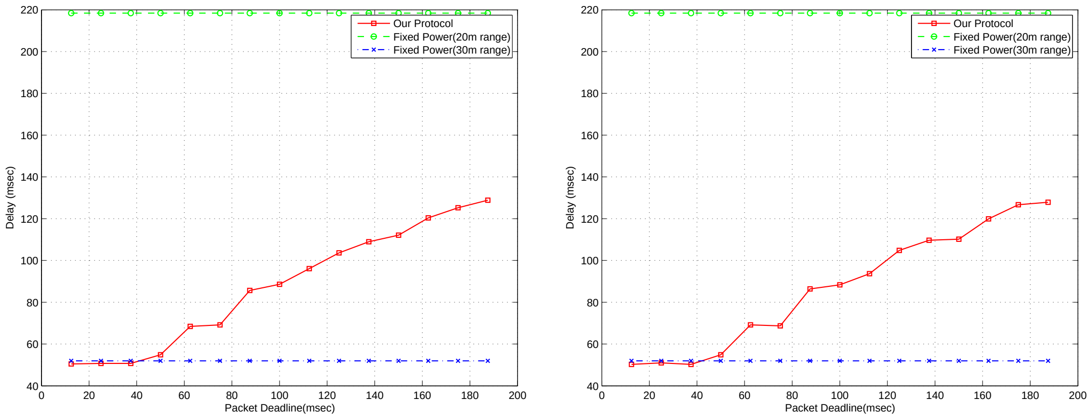
(b) $\alpha = 3$

(c) $\alpha = 4$

(d) $\alpha = 5$

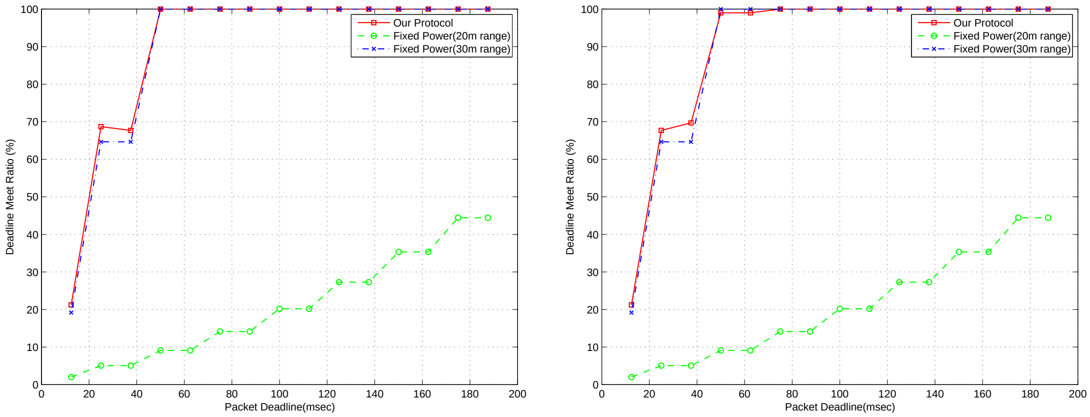
(e) $\alpha = 6$

Figure 4.1: Effect of α on transmission energy when packet deadline changes.



(a) $\alpha = 2$ (b) $\alpha = 6$

Figure 4.2: Effect of α on delay per packet when packet deadline changes.



(a) $\alpha = 2$ (b) $\alpha = 6$

Figure 4.3: Effect of α on deadline meet ratio when packet deadline changes.

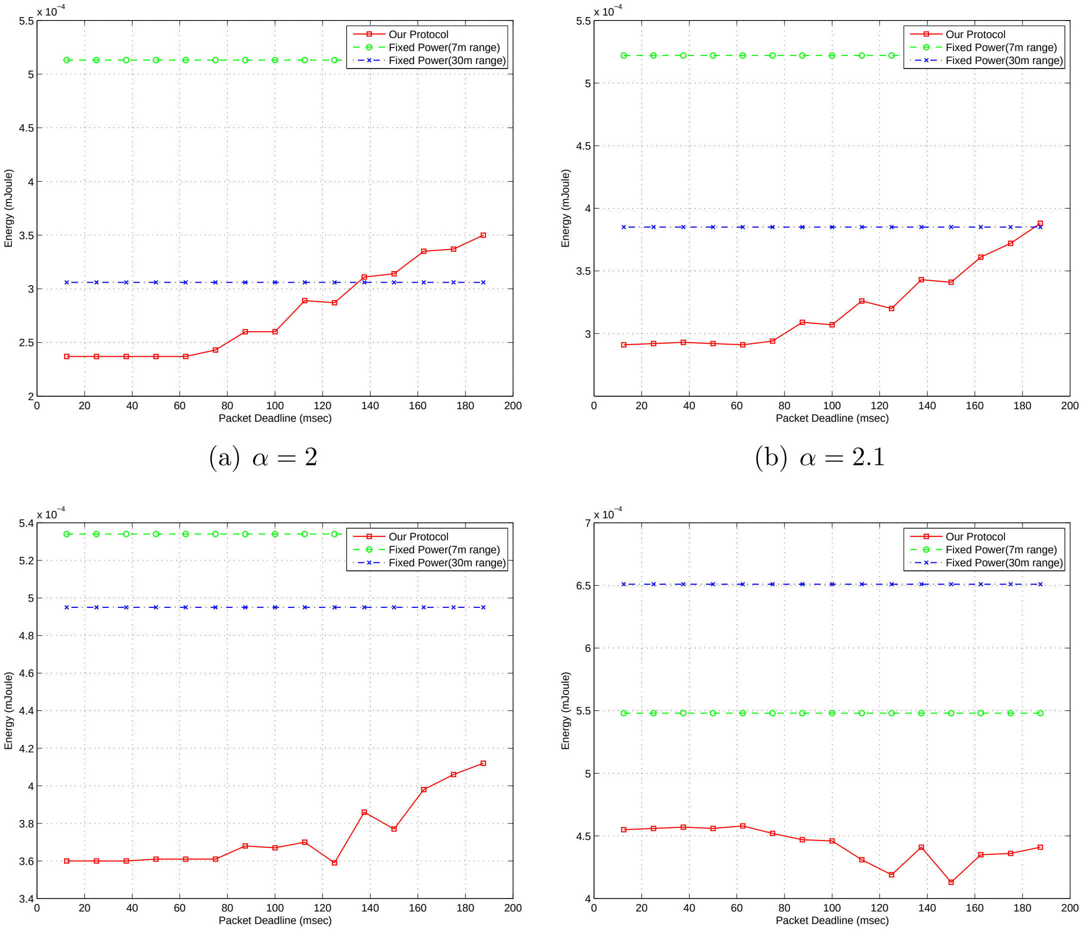
as shown in Figure 4.1(a). Starting from $\alpha = 3$, the energy consumption values rise as packet priority gets higher. The change in the energy consumption results occurs between $\alpha = 2$ and $\alpha = 3$. Therefore we will examine the behavior of our protocol for α values between 2 and 3 in the following experiments.

4.2.2 Performance Under Light Traffic

In this part, we will discuss the performance results of our approach when each node in the network generates one packet per second and sends 15 packets, i.e., one packet per deadline as in the previous experiment. This time we test our protocol using both random and grid deployments. The protocol first establishes the routing tree and we set $maxParents$ to 2, so all nodes find 2 parents. In order to observe the behavior of energy consumption when α changes between 2 and 3, we test our approach for all the values between 2 and 3 with a step size of 0.1.

Figures 4.4 and 4.5 show the change in energy consumption as α changes between 2 and 3 for grid deployment and random deployment, respectively. According to these results, starting from $\alpha = 2.3$ the energy consumption of our protocol decreases as the packet deadline increases, and behaves as expected. For path loss exponent values smaller than this, the energy values increase as packet priority decrease, due to the reasons explained in the previous section.

As the simulation results indicate, our protocol becomes clearly advantageous starting from $\alpha = 2.3$ since it consumes much less energy than the protocol that uses a fixed power with a range of 30 m. Moreover, for packets with looser deadlines the energy consumption falls to the same levels with fixed power with the minimum range. While α increases, the difference between the energy consumption of fixed power protocol with range 30m and our protocol increases even more. For $\alpha = 3$, the difference in energy consumed per packet for grid and random deployments are respectively 2.5 mJ and 3.5 mJ for top priority packets and 4.5 mJ and 6 mJ for least priority packets.



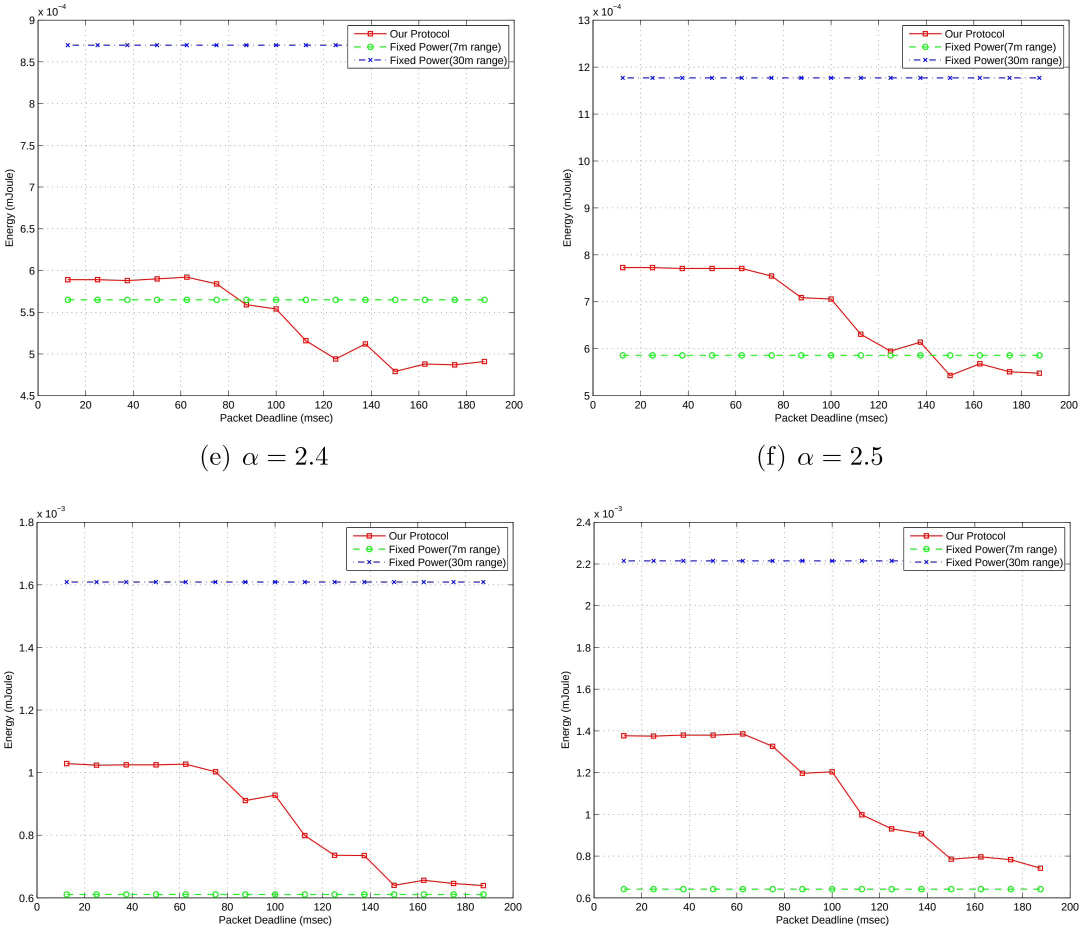
(a) $\alpha = 2$

(b) $\alpha = 2.1$

(c) $\alpha = 2.2$

(d) $\alpha = 2.3$

Figure 4.4: Packet deadline vs. average energy for grid deployment.



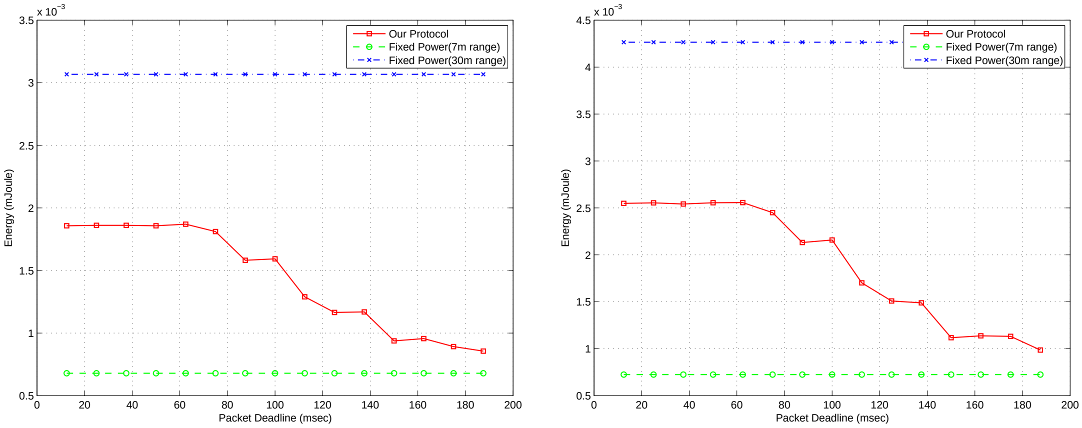
(e) $\alpha = 2.4$

(f) $\alpha = 2.5$

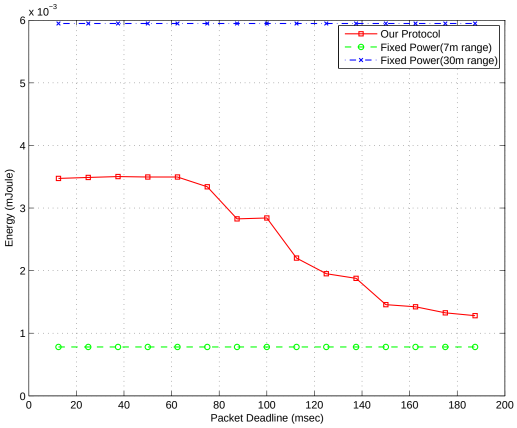
(g) $\alpha = 2.6$

(h) $\alpha = 2.7$

Figure 4.4: Packet deadline vs. average energy for grid deployment (cont.).

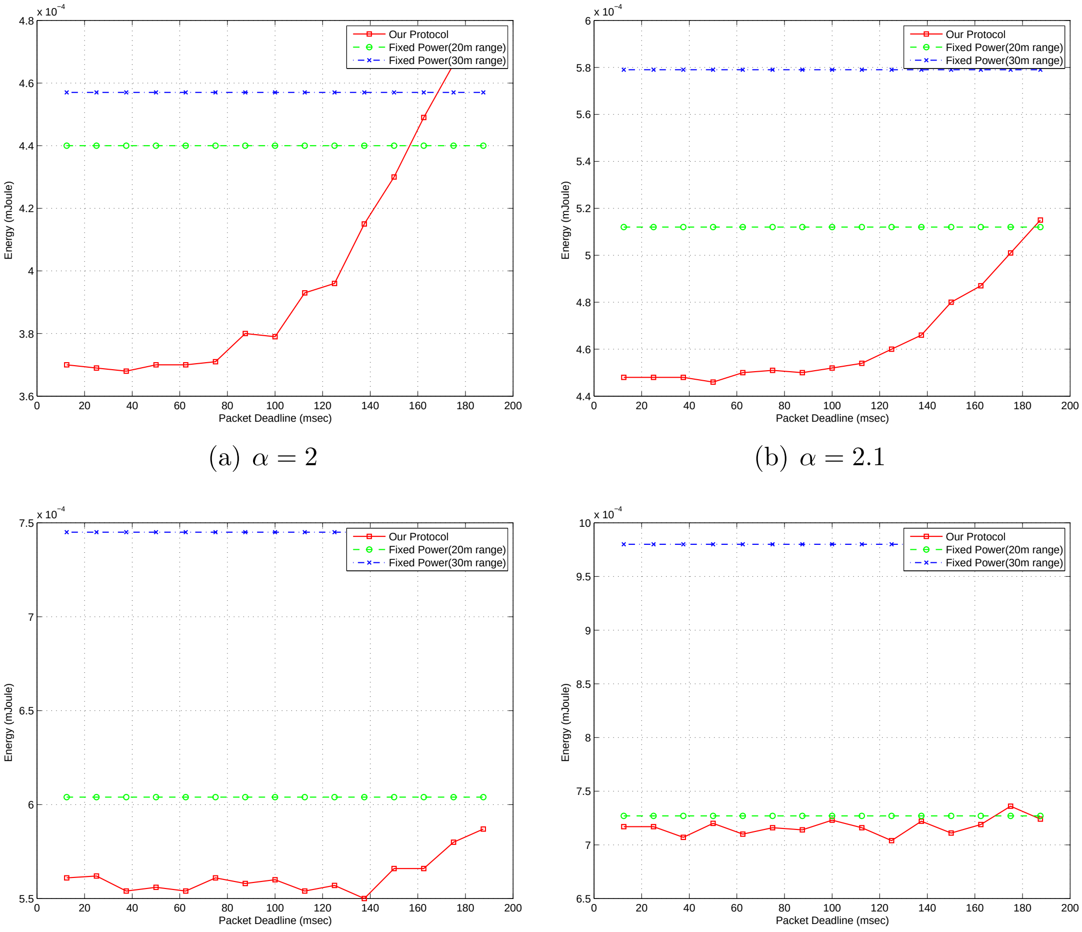


(i) $\alpha = 2.8$ (j) $\alpha = 2.9$



(k) $\alpha = 3$

Figure 4.4: Packet deadline vs. average energy for grid deployment (cont.).



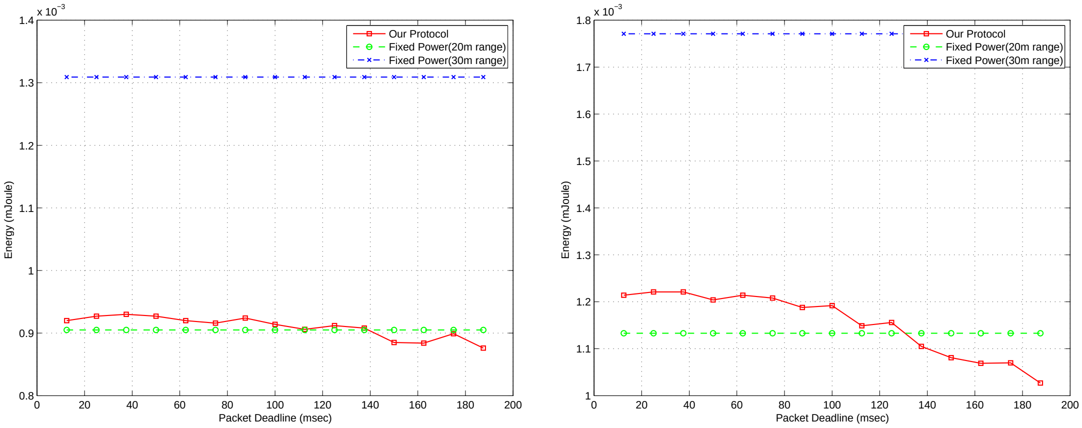
(a) $\alpha = 2$

(b) $\alpha = 2.1$

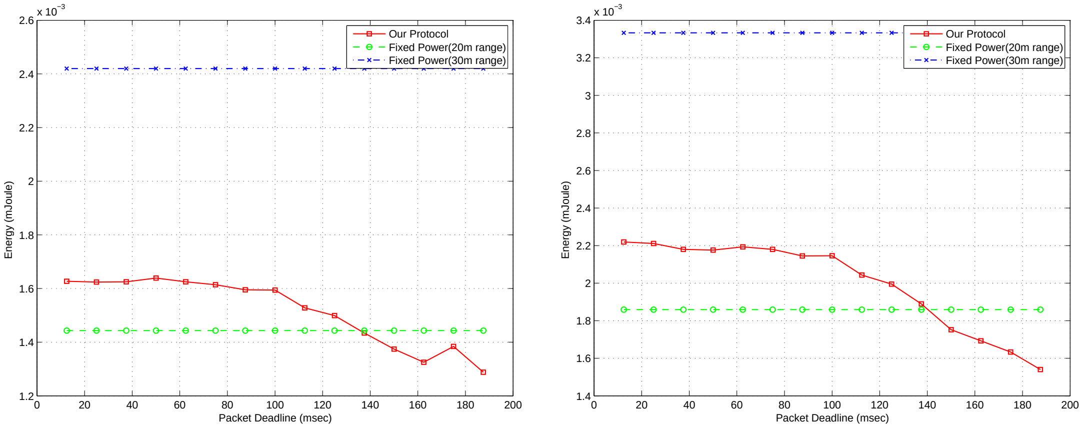
(c) $\alpha = 2.2$

(d) $\alpha = 2.3$

Figure 4.5: Packet deadline vs. average energy for random deployment.

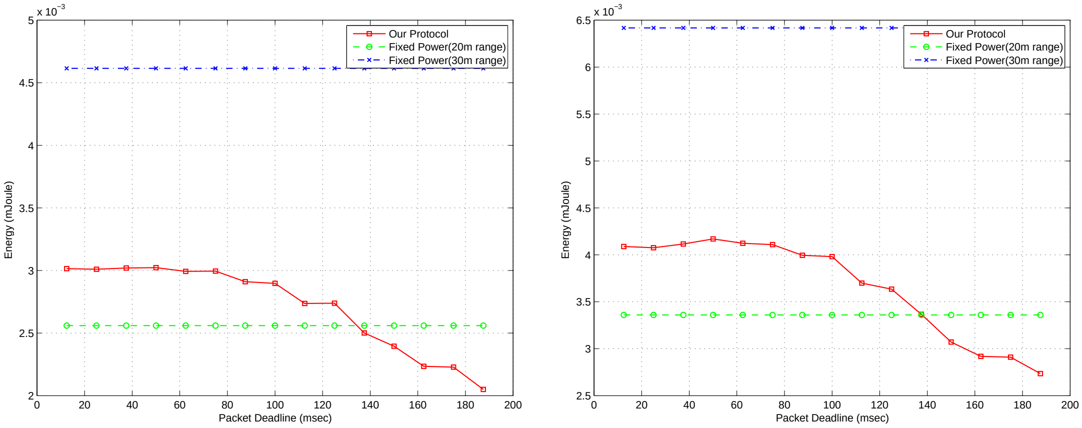


(e) $\alpha = 2.4$ (f) $\alpha = 2.5$

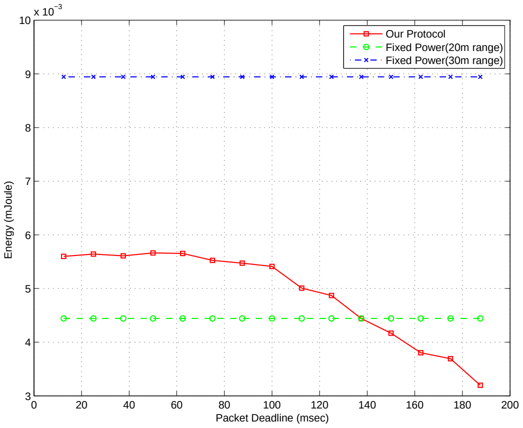


(g) $\alpha = 2.6$ (h) $\alpha = 2.7$

Figure 4.5: Packet deadline vs. average energy for random deployment (cont.).

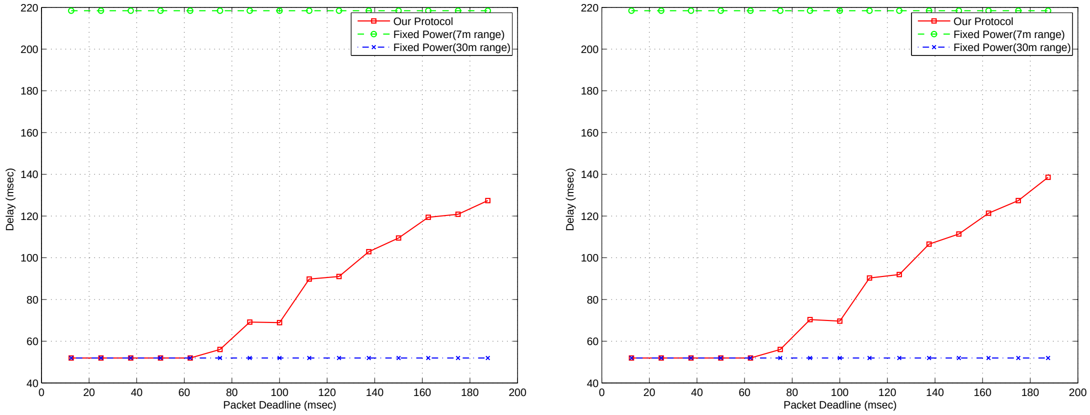


(i) $\alpha = 2.8$ (j) $\alpha = 2.9$

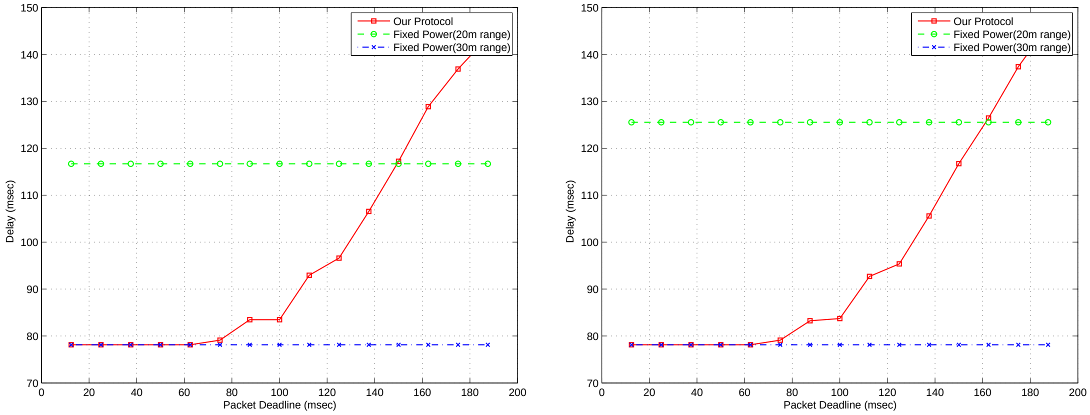


(k) $\alpha = 3$

Figure 4.5: Packet deadline vs. average energy for random deployment (cont.).



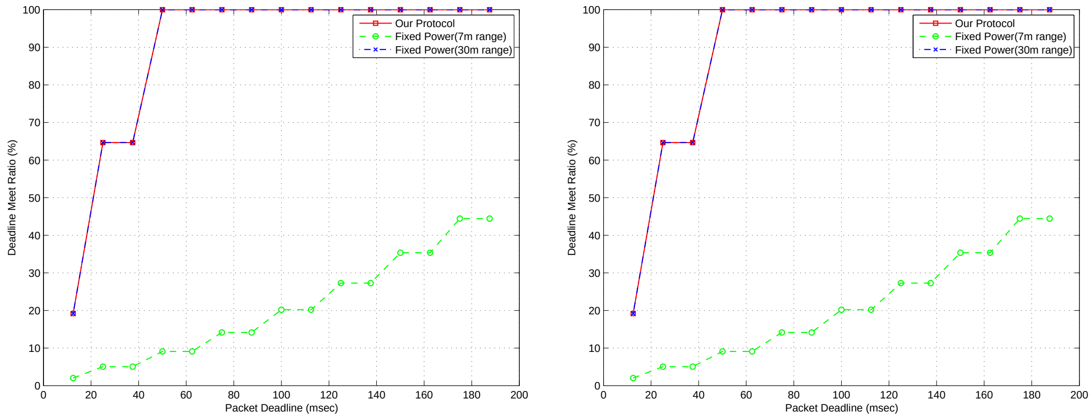
(a) $\alpha = 2$ for grid deployment (b) $\alpha = 3$ for grid deployment



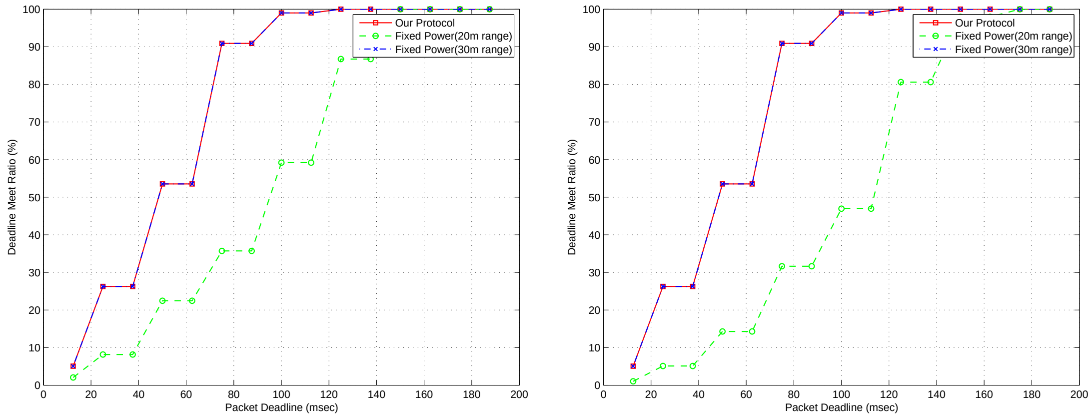
(c) $\alpha = 2$ for random deployment (d) $\alpha = 3$ for random deployment

Figure 4.6: Packet deadline vs. delay for grid and random deployments.

When we take a look at the results in Figure 4.6, i.e., for packets with different deadlines, we see that for our approach delay increases as packet deadline increases. As predicted, the protocol that uses fixed power for 30 m range has low delay values for all packet types and our protocol has approximately the same delay values for urgent packets but larger values for less urgent packets. The protocol that uses minimum fixed power causes large delays, again as expected. For deadline meet ratios shown in Figure 4.7, we see that our protocol reaches to the same ratio with the maximum fixed power protocol while minimum fixed power has significantly low ratios. Also, in case of random deployment, delay values of our protocol exceed the delay values of minimum fixed power scheme for less urgent packets, but still meet the deadline (Figures 4.6 and 4.7).

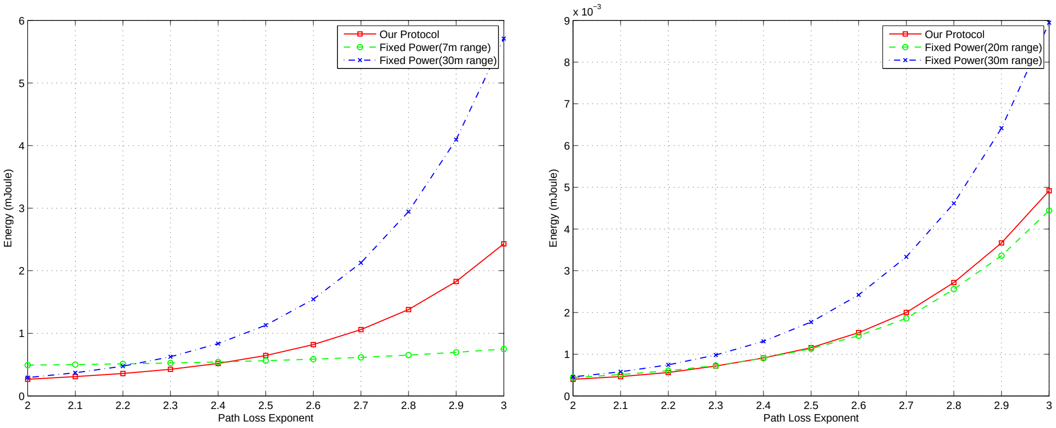


(a) $\alpha = 2$ for grid deployment (b) $\alpha = 3$ for grid deployment



(c) $\alpha = 2$ for random deployment (d) $\alpha = 3$ for random deployment

Figure 4.7: Packet deadline vs. average deadline meet ratio for grid and random deployments.



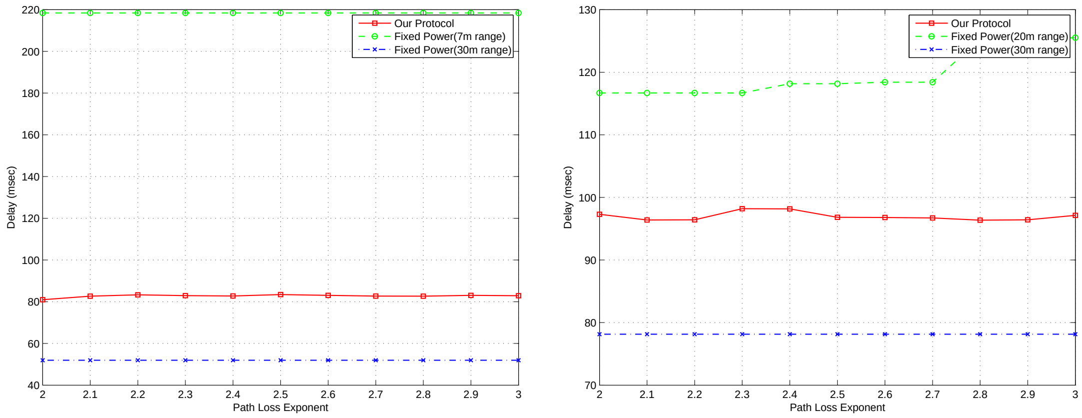
(a) Grid deployment (b) Random deployment

Figure 4.8: Path loss exponent (α) vs. average energy spent per packet for grid and random deployments.

Figures 4.8 and 4.9 show the *average* energy and delay per packet as α changes between 2 and 3. As predicted, the energy values increase in greater-than-linear fashion and our protocol has lower average energy consumption than fixed power protocol with 30 m range. The average delay values indicate that our approach yields results close to the maximum fixed power protocol.

In this experiment, we also examined the interference caused by our protocol and compared it with fixed power protocols. Figure 4.10 shows the total number of nodes that are affected by other transmissions in one run. According to this figure, maximum fixed power protocol affects an excessive number of nodes while our protocol has a much less interference effect, close to the minimum fixed power protocol. According to these results, interference does not increase with α. As explained in section 4.1, in our experiment we adapt the transmission power according to α so that desired range is achieved. Since the range does not vary according to α, the number of nodes in the range does not change. Hence, the interference values, which depend on the number of nodes affected by other transmissions, do not increase with respect to α.

In Figure 4.11, the number of nodes multiplied by how much they are affected in terms of received signal is presented. This time as predicted, the affect of interference also grows when α increases, especially for maximum fixed power



(a) Grid deployment (b) Random deployment

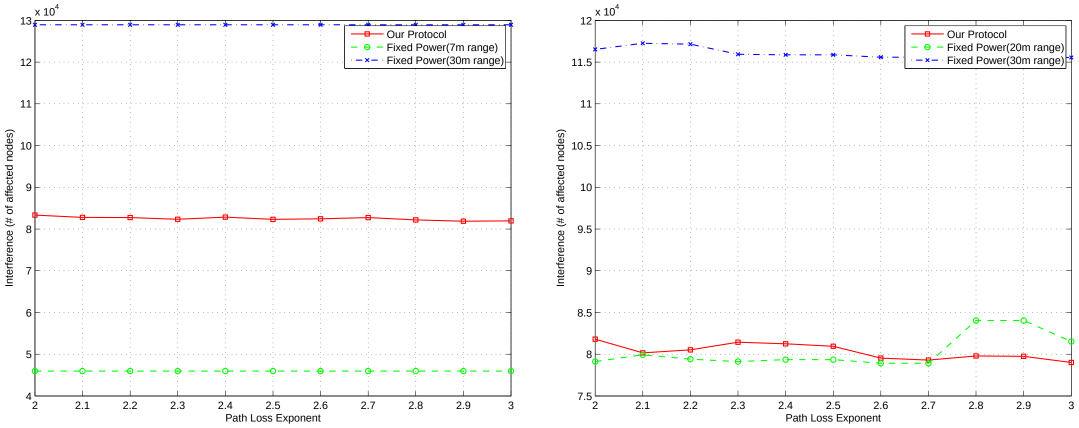
Figure 4.9: Path loss exponent (α) vs. average delay per packet for grid and random deployments.

protocol. Our approach causes only a limited increase in the weighted interference value.

4.2.3 Network Lifetime

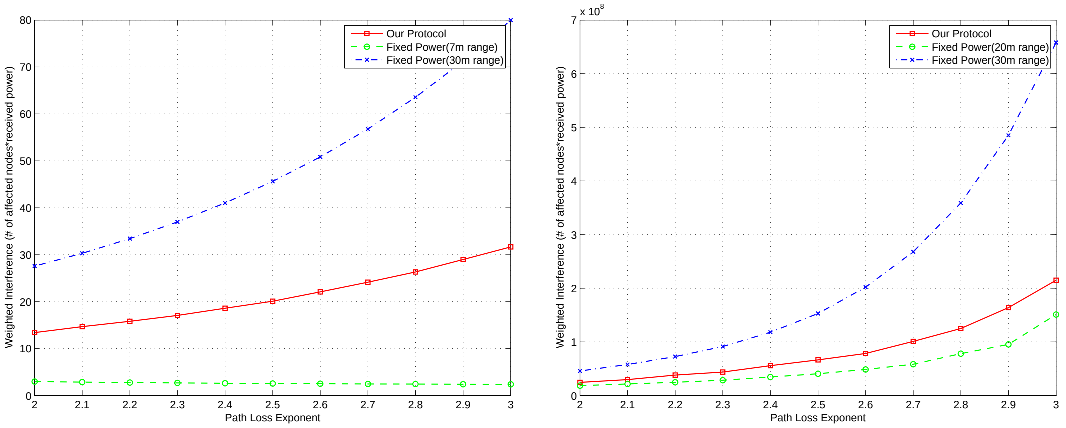
In order to evaluate the performance of our protocol on network lifetime, we conducted experiments by considering the remaining energy in nodes. We assumed the nodes have an initial energy of 5000 mJ and considered both transmission and receiving energy consumptions. The nodes generate packets each second with random priorities, until one of the nodes have a remaining energy of 4750 mJ. We consider *network lifetime* as the time between the initialization of the network and the time when the first node in the network has a remaining energy of 4750 mJ. When the first node reaches the remaining energy threshold, simulation stops. The energy consumed in routing tree establishment is not considered for neither of the protocols.

Additionally, in the routing tree establishment, *maxParents* parameter is changed between 1 and 4. For different α values, the total energy consumed for transmission and receiving per packet type as the number of parents change is



(a) Grid deployment (b) Random deployment

Figure 4.10: Path loss exponent (α) vs. interference for grid and random deployments.



(a) Grid deployment (b) Random deployment

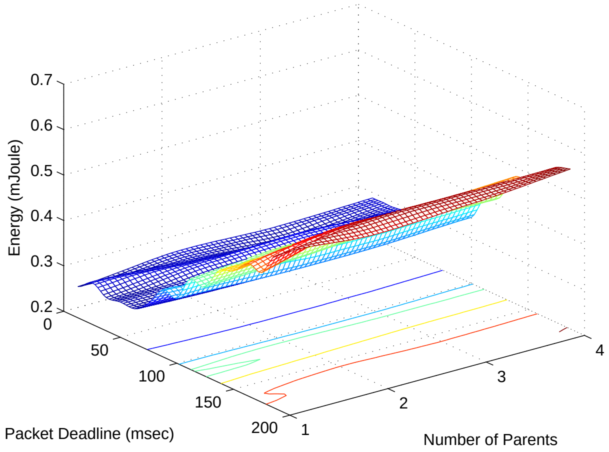
Figure 4.11: Path loss exponent (α) vs. weighted interference for grid and random deployments.

shown in Figure 4.12 for grid deployment. The delay and deadline meet ratio values are also shown in Figures 4.13 and 4.14, respectively. As the change in α does not have a significant effect on delay and deadline meet ratios, only $\alpha = 2$ and $\alpha = 3$ values are shown. According to the results, the number of parents does not cause significant changes in energy, delay and deadline meet ratio values.

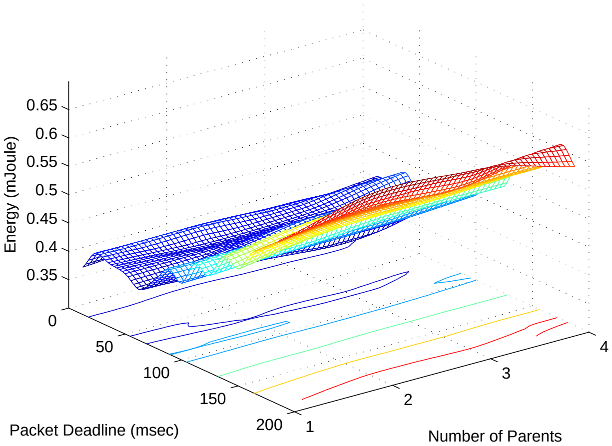
Figure 4.15 depicts the time passed until one of the nodes had a remaining energy of 4750 mJ, for both grid and random deployments. As predicted, network lifetime decreases as α grows because when we assign higher values to α, the energy consumption grows larger and the nodes reach the threshold remaining energy faster. According to the results shown, it can be inferred that our protocol leads to a longer lifetime than maximum fixed power protocol for all α values. In our approach, the nodes have more than one parent because of transmit power adjustment and hence the load of one node is distributed among many parents. The maximum lifetime is achieved when $maxParent = 1$ for grid deployment and when $maxParent = 2$ for random deployment. We can infer from the results that there is no significant change with respect to $maxParents$. Additionally, $maxParents$ setting affects the performance differently depending on the topology.

In Figure 4.16, we can see the average remaining energy per node for grid and random deployments when the simulation ended. The average remaining energy depends on network lifetime and when lifetime is longer the remaining energy values are smaller. Consequently, our protocol has smaller average remaining energy values than maximum fixed power scheme, especially for large α values.

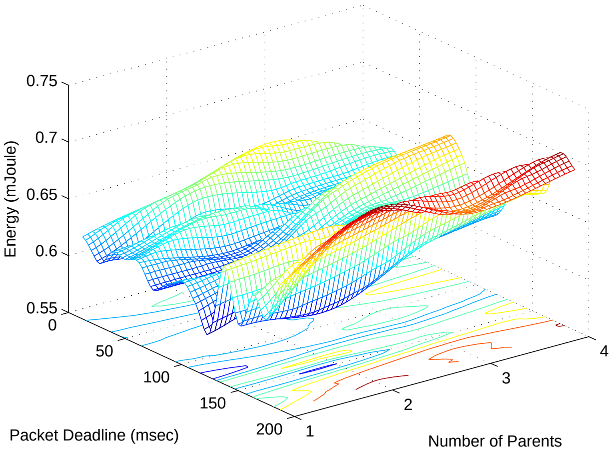
In this chapter, we have presented the experimental results for our protocol and discussed its performance for a number of metrics. In order to evaluate the real-time communication support of our approach, we considered the metrics energy consumption, delay, and deadline meet ratio per packet type. We can infer from the simulation results that our protocol can provide real-time guarantees for different types of traffic when α is greater than 2.2. The deadline meet ratios per packet type are as high as the maximum fixed power scheme for all α values while there is a significant gain in energy consumption for transmission where $\alpha \geq 2.3$.



(a) $\alpha = 2$

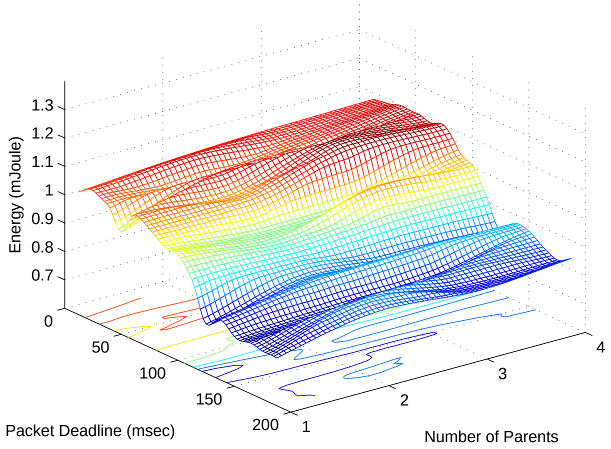


(b) $\alpha = 2.2$

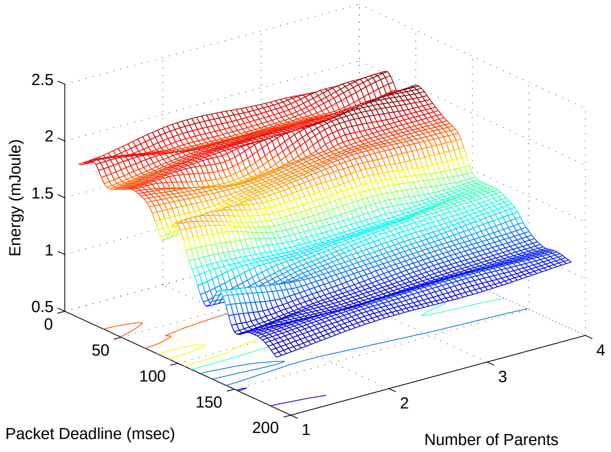


(c) $\alpha = 2.4$

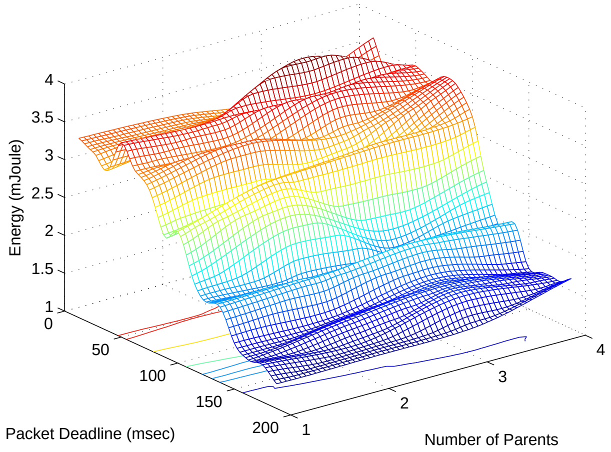
Figure 4.12: Packet deadline vs. energy as parent number ($maxParents$) and α changes.



(d) $\alpha = 2.6$

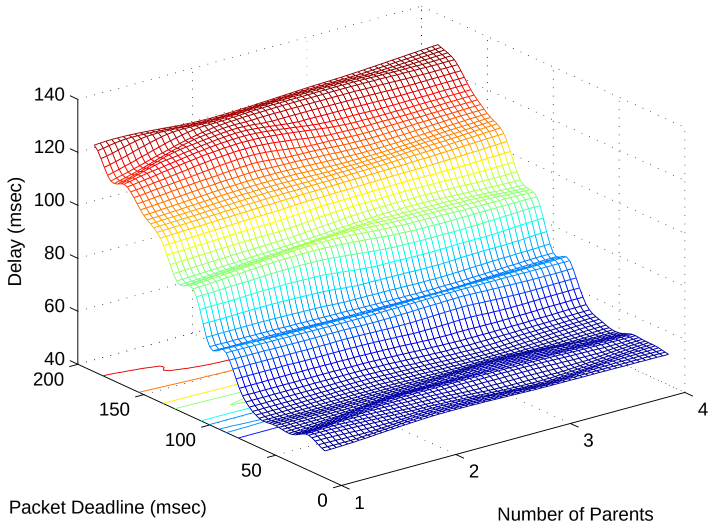


(e) $\alpha = 2.8$

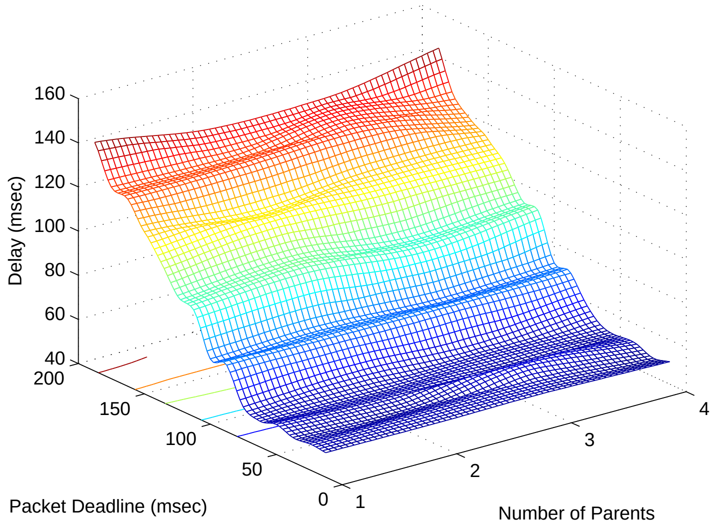


(f) $\alpha = 3$

Figure 4.12: Packet deadline vs. energy as parent number ($maxParents$) and α changes.

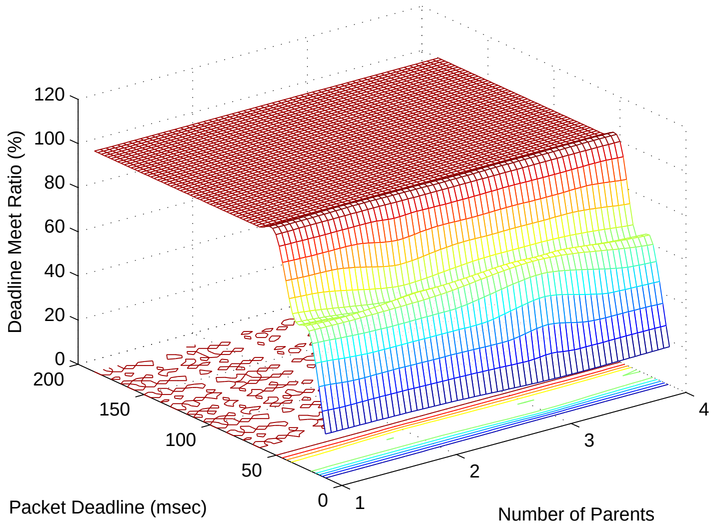


(a) $\alpha = 2$

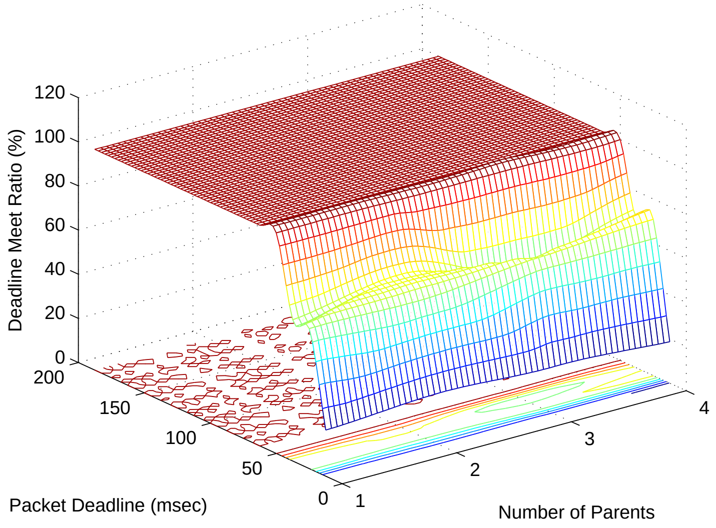


(b) $\alpha = 3$

Figure 4.13: Packet deadline vs. delay as parent number ($maxParents$) and α changes.

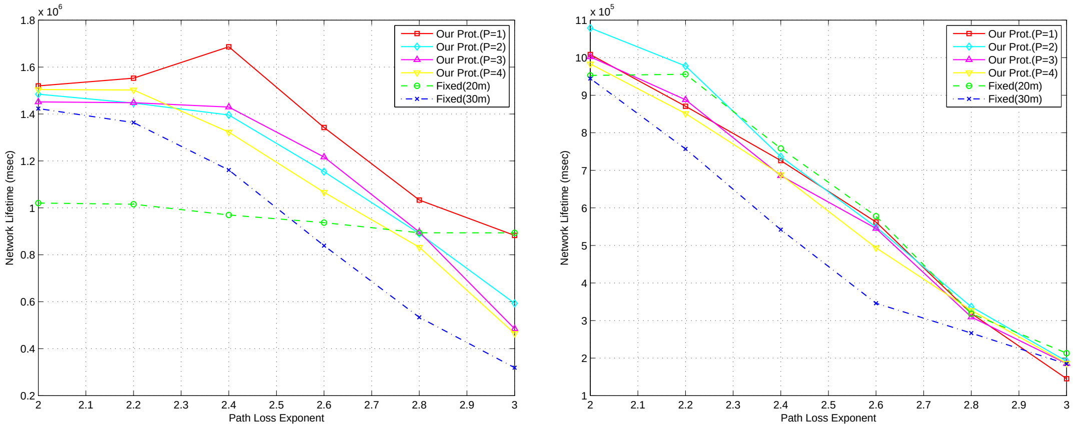


(a) $\alpha = 2$



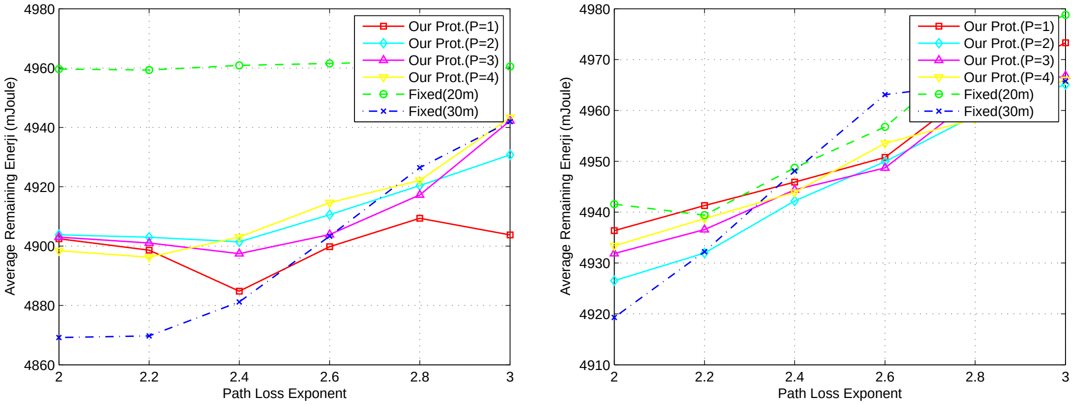
(b) $\alpha = 3$

Figure 4.14: Packet deadline vs. deadline meet ratio as parent number ($maxParents$) and α changes.



(a) Grid deployment (b) Random deployment

Figure 4.15: Path loss exponent (α) vs. lifetime for grid and random deployments.



(a) Grid deployment (b) Random deployment

Figure 4.16: Path loss exponent (α) vs. average remaining energy for grid and random deployments.

We also inspected the effects of our protocol on network capacity by evaluating the resulting interference on the network. The results indicate that our protocol leads lower number of interfered nodes. Additionally, the weighted interference caused by our protocol, which is the number of affected nodes multiplied by received power, is smaller than the amount caused by maximum fixed power scheme.

Our protocol has also performed well in network lifetime experiment. It yielded a longer network lifetime when compared with maximum fixed power scheme. However, we saw that the change in the parent number does not affect overall performance significantly.

Chapter 5

Conclusion and Future Work

Wireless sensor networks facilitate development of a wide range of applications for monitoring and surveillance purposes. Many of these applications involve real-time communication which imposes timely delivery of urgent packets and these packets might have a variety of deadline requirements depending on the application. In this thesis, we propose a routing protocol which aims to support real-time traffic with varying deadline requirements. We utilize transmit power adjustment to differentiate packets according to priority and reduce end-to-end delay. Moreover, we want to lower the energy dissipated in transmissions while also reducing the interference caused by transmissions.

Studies in literature that support real-time communication and transmit power control generally specialize on geographical routing which has some drawbacks. One disadvantage is that localization services necessary for geographical routing may not be suitable for indoor environments and also they consume high levels of energy. Moreover, most of the related works do not differentiate packets according to their QoS requirements and do not consider the increased interference.

In order to analyze the effects of transmit power on delay, we conducted some experiments on our sensor motes and observed that increasing transmit power can reduce end-to-end delay considerably, as predicted. Furthermore, we modeled a simple network to examine the resulting delay and energy consumption values when transmit power is varied. The outcome of this analysis also confirmed that different levels of transmit power causing different ranges shape delay and transmit energy consumption. Also transmit power adjustment has a deeper effect as the path loss exponent increases.

Our protocol, Real-time Routing with Priority Scheduling and Power Adjustment, uses transmit power adjustment to meet strict delay obligations by decreasing the number of hops between sender and receiver. In case of less urgent packets, we reduce the transmit power so that the node can send packets to a next hop which provides longer routes in terms of hop count but energy efficient at the same time. Our protocol undergoes a routing tree setup phase first, for the nodes to learn their one hop neighbors and to find eligible parent nodes that provide routes with desired properties. Since we assume the traffic flows from the nodes towards the sink node, each node maintains next hop information that supports energy efficient routes to sink with low delay values. After the routing tree is formed, the nodes make packet forwarding decisions based on the packet priority in order to differentiate packets. In order to deliver urgent packets on time, nodes increase transmit power and forward packets to parent nodes that provide routes with smaller number of hops. For less urgent packets, the parent nodes that provide energy efficient routes with larger number of hops are selected.

We tested our approach in a simulation environment with idealistic settings and observed the effects of path loss exponent, different deployment schemes and different number of parents per hop count in comparison with fixed power protocols. Simulation results show that our routing protocol increases the deadline meet ratio per packet type and reduces average delay per packet type. While the deadline meet ratio values are approximately the same with maximum fixed power scheme, the energy consumption values for our protocol are significantly lower when the path loss exponent is greater than 2.2. We see that for this setting our protocol has energy consumption values closer to minimum fixed power scheme yet the delay values for this scheme are significantly higher.

We also examined the number of nodes that are interfered from other nearby transmissions and how much they are affected. We compared the total number of interfered nodes and also sum of the power they received. The results indicate that our approach yields less interference than maximum power scheme in terms of both number of nodes and the level they are affected.

Additionally, we compared the network lifetime and average remaining energy values of nodes for our approach and fixed power protocols. The lifetime until the first node reaches the threshold of the remaining energy level, and the remaining energy values of the nodes are evaluated and the resulting values imply that our protocol is advantageous.

Although the simulation results for ideal settings show that we can achieve our objectives with our proposed approach, we can extend it to achieve better results. First of all, the delay estimation could be improved since for now we derive the end-to-end delay from one hop delay. Also, the protocol should update the routing tree based on changing network conditions. We can attain this goal by getting feedback from the transmissions. Moreover, we can test our protocol with more realistic settings as a future work. For example wireless channel conditions can cause unexpected network conditions and affect the performance. Besides, the energy consumption values depend heavily on radio characteristics. Therefore our protocol can be tested with more realistic wireless channel and radio models. In addition, we can compare our protocol with different real-time routing protocols.

Bibliography

[1] K. Akkaya and M. Younis. An energy-aware qos routing protocol for wireless sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems*, page 710. IEEE Computer Society, 2003.

[2] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey on wireless multimedia sensor networks. *Computer Networks*, 51(4):921–960, 2007.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, 38(4):393–422, 2002.

[4] R. M. Anthony Rowe and R. Rajkumar. FireFly: A time synchronized real-time sensor networking platform. `http://www.andrew.cmu.edu/\~agr/pubpg/firefly-06.pdf`.

[5] J. Arias, J. Lázaro, A. Zuloaga, J. Jiménez, and A. Astarloa. GPS-less location algorithm for wireless sensor networks. *Computing Community*, 30(14-15):2904–2916, 2007.

[6] A. Barberis, L. Barboni, and M. Valle. Evaluating energy consumption in wireless sensor networks applications. In *Proceedings of the 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*, pages 455–462. IEEE Computer Society, 2007.

[7] E. H. Callaway. *Wireless Sensor Networks: Architectures and Protocols.* CRC Press, Inc., Boca Raton, FL, USA, 2003.

[8] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. *Quality*

of Service, 2006. IWQoS 2006. 14th IEEE International Workshop on, pages 83–92, 2006.

[9] Chipcon. CC2420 datasheet. `http://enaweb.eng.yale.edu/drupal/system/files/CC2420\_Data\_Sheet\_1\_4.pdf`.

[10] C. D. M. Condeiro and D. P. Agrawal. *Ad Hoc & Sensor Networks: Theory And Applications*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2006.

[11] Crossbow. MICA2 datasheet. `http://www.xbow.com/products/Product\_pdf\_files/Wireless\_pdf/MICA2\_Datasheet.pdf`.

[12] Crossbow. MICAz datasheet. `http://www.xbow.com/products/product\_pdf\_files/wireless\_pdf/6020-0060-01\_a\_micaz.pdf`.

[13] Crossbow. TelosB datasheet. `http://www.xbow.com/Products/Product\_pdf\_files/Wireless\_pdf/TelosB\_Datasheet.pdf`.

[14] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. K. Dao. Power management for throughput enhancement in wireless ad-hoc networks. In *IEEE International Conference on Communications*, pages 1506–1513, 2000.

[15] S. C. Ergen and P. Varaiya. Energy efficient routing with delay guarantee for sensor networks. *Wireless Networks*, 13(5):679–690, 2007.

[16] E. Felemban, C.-G. Lee, and E. Ekici. MMSPEED: Multipath multi-speed protocol for qos guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing*, 5(6):738–754, 2006.

[17] Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall, and X. H. Peng. Radio range adjustment for energy efficient wireless sensor networks. *Ad Hoc Networks*, 4(1):75–82, 2006.

[18] P. Gupta and P. Kumar. Capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[19] T. He, J. A. Stankovic, C. Lu, and T. F. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. *23rd International Conference on Distributed Computing Systems*, pages 46–55, 2003.

[20] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Communications*, 1(4):660–670, 2002.

[21] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 8, page 8020. IEEE Computer Society, 2000.

[22] B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins. *Global Positioning System: Theory and Practice*. Springer; 5th edition, 2004.

[23] J. Jeong, D. Culler, and J.-H. Oh. Empirical analysis of transmission power control algorithms for wireless sensor networks. *Fourth International Conference on Networked Sensing Systems*, pages 27–34, 2007.

[24] V. Kawadia and P. Kumar. Principles and protocols for power control in wireless ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(1):76–88, 2005.

[25] M. Kubisch, H. Karl, A. Wolisz, L. Zhong, and J. Rabaey. Distributed algorithms for transmission power control in wireless sensor networks. *Wireless Communications and Networking*, 1:558–563, 2003.

[26] Y. Li, C. S. Chen, Y.-Q. Song, and Z. Wang. Real-time qos support in wireless sensor networks: A survey. In *7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems*, 2007.

[27] S. Lin, J. Zhang, G. Zhou, L. Gu, J. A. Stankovic, and T. He. ATPC: adaptive transmission power control for wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 223–236, 2006.

[28] C. Lu, B. M. Blum, T. F. Abdelzaher, J. A. Stankovic, and T. He. RAP: A real-time communication architecture for large-scale wireless sensor networks. In *IEEE Real Time Technology and Applications Symposium*, pages 55–66. IEEE Computer Society, 2002.

[29] S. Misra, M. Reisslein, and G. Xue. A survey of multimedia streaming in wireless sensor networks. *IEEE Communications Surveys and Tutorials,10(3)*, 2008.

[30] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad hoc networks: Theory, architecture, algorithm, and implementation of the COMPOW protocol. *European Wireless Conference*, pages 156–162, 2002.

[31] P. Pothuri, V. Sarangan, and J. Thomas. Delay-constrained, energy-efficient routing in wireless sensor networks through topology control. *Networking, Sensing and Control, 2006. ICNSC '06. Proceedings of the 2006 IEEE International Conference on*, pages 35–41, 2006.

[32] R. Ramanathan and R. Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 404–413, 2000.

[33] G. Simon, P. Vlgyesi, M. Marti, and kos Ldeczi. Simulation-based optimization of communication protocols for large-scale wireless sensor networks. *IEEE Aerospace Conference*, 3:1339–1346, 2003.

[34] D. Son, B. Krishnamachari, and J. Heidemann. Experimental study of the effects of transmission power control and blacklisting in wireless sensor networks. In *Proceedings of the First IEEE Conference on Sensor and Adhoc Communication and Networks*, pages 289–298, 2004.