國立臺灣大學資訊管理研究所碩士論文

指導教授：林永松　博士
顏宏旭　博士

具資料集縮能力之無線感測網路節能路由演算法

An Energy-Efficient Data-Centric Routing Algorithm in Wireless Sensor Networks

研究生：林書平　撰

中華民國九十四年七月

具資料集縮能力之無線感測網路
節能路由演算法

An Energy-Efficient Data-Centric Routing Algorithm in Wireless Sensor Networks

本論文係提交國立台灣大學

資訊管理研究所作為完成碩士

學位所需條件之一部份

研究生：林書平　撰

中華民國九十四年七月

謝　　誌

兩年研究所生涯隨著本篇論文的完成告一段落,研究生涯的結束也代表著人生的下一階段已在眼前向我招手，等著我大步邁進。回顧兩年的碩士生活，是我付出最多心力亦是收穫最大的階段,而論文的完成歸功於許多人對我的支持與鼓勵，對此心中更是充滿無限的感激與對自己的成就感。

首先我要感謝恩師林永松 博士，老師總是能循序漸進地啟發我思考問題，並適時引導我至正確的方向;每當我自滿於研究成果時，老師總能在我疏忽處提點我更嚴謹的研究方法，而老師認真嚴謹的治學態度與風度偏偏的待人處事，亦讓我學到學者應有的研究精神與做人處事的道理。此外我要感謝顏宏旭 博士兩年來辛勤的指導，顏老師每每在我陷入難題時適時助我跳脫問題的泥沼，指點我研究過程中下一步該如何走，顏老師認真的研究精神亦是我學習的楷模。感謝兩位恩師的指導，您的恩情學生銘記在心。感謝口試委員趙啟超 博士、林盈達 博士和呂俊賢 博士給予學生諸多寶貴的建議和指正,使本篇論文能更臻完備嚴謹。

另外要感謝博士班演福學長、旭成學長和政達學長對本篇論文的協助，在與學長討論過程中我總是能找到正確的研究方向，也感謝柏皓學長的鼓勵，讓我在困惑時能不迷失自我。當然也要謝謝陪我一起奮鬥的戰友們:明源、建宏、孝澤、琳智，在生活與課業上互相的扶持，以及研究所的同學們，謝謝你們給予我如此豐富的研究生活。也謝謝學弟妹們:中蓮、弘翕、文政、義倫與勇誠，謝謝你們在口試期間的協助，讓我可以專心準備口試，而你們的活潑也帶給我許多歡笑。

在此我要感謝我最親愛的家人，尤其是我的父母親:林振昌先生與鄭素珠女士，謝謝你們對我的照顧與關愛，在那迷惘的黑暗中沒有你們給予的勇氣我無法渡過這段艱辛，謝謝你們我最愛的家人，我永遠以你們為榮。

最後我要感謝所有我愛以及愛我的人，無論在生活及學業上，我的生命因你們的參與而更加充實、精彩;謝謝你們，我會永懷感恩之心繼續往前邁進。

林書平　謹識

于台大資訊管理研究所

中華民國九十四年七月

論文摘要

論文題目 ： 具資料集縮能力之無線感測網路節能路由演算法

作　　者 ： 林書平　　　　　　　　　　　　　　　　　民國九十四年七月

指導教授 ： 林永松　博士

　　　　　　顏宏旭　博士

近年來無線感測網路在諸多領域上具有高度應用價值，已被視為下一階段無線通訊技訊的殺手級應用。然而受限於硬體以及應用環境，使感測器對於能源消耗具有高度的限制性，因此在感測網路中降低感測器於運作過程所消耗的能源已成為一個重要的研究議題。結合資料集縮（data aggregation）能力於感測器中藉此減少資料傳輸量能有效降低感測器的能源消耗，而具備資料集縮能力之感測網路需有以資料為中心（data-centric）之路由演算法來充份利用此能力以達到節省能源消耗的目的。

本篇論文研究在感測器具有資料集縮能力之無線感測網路中，建立兼具最小能源消耗量及服務品質之資料集縮樹（data aggregation tree）的問題，在考量感測器之路由指定、傳輸半徑（communication radius）及資料重傳（data retransmission）限制下，透過集縮樹的方式盡可能減少資料的傳輸總量，使感測網路回報資料時所需消耗之總能量最小化，藉此提高感測網路的系統生存時間。此外，為了使問題能更符合應用環境，在考量資料集縮所帶來的集縮延遲時間限制下，我們將最大端對端延遲（maximum end-to-end delay）的概念考慮進來，並加以定義為感測器所消耗之集縮等待能源消耗，藉以更精確的描述感測網路的總能源消耗。

我們將整個問題數學模式化為一個嚴謹的混合式整數線性最佳化數學模

型，目標函式為最小化資料傳輸所需之總能源消耗，此數學問題在本質上是一個整數非線性規劃問題，問題本身具有高度的複雜性和困難度。本論文採用以拉格蘭日鬆弛法為基礎的方法來處理此一複雜問題，並根據所得到的結果改良演算法以得到較佳的資料集縮樹。實驗結果顯示，我們不但能有效率地求得此問題解，且在問題解的效能上亦比其它既有的演算法更為優越。

關鍵詞：資料集縮、資料中心路由、高效率節能路由、最佳化、拉格蘭日鬆弛法、混合式整數線性規劃、無線感測網路。

THESIS ABSTRACT

GRADUATE INSTITUTE OF INFORMATION MANAGEMENT

NATIONAL TAIWAN UNIVERSITY

NAME : SHU-PING LIN MONTH/YEAR : JULY/2005

ADVISOR : DR. YEONG-SUNG LIN

DR. HONG-HSU YEN

AN ENERGY-EFFICIENT DATA-CENTRIC ROUTING ALGORITHM IN WIRELESS SENSOR NETWORKS

Recently wireless sensor networks (WSNs) have attracted a great deal of attention due to their potential for numerous military, environmental detection, and civil applications. Sensor nodes in WSNs are highly energy-constrained, because of the limitations of hardware and the infeasibility of recharging sensor nodes under a severe environment. When a sensor is in operation, therefore, reducing energy consumption during the forwarding and sensing of data is a crucial issue in WSNs. Incorporating sensor nodes with data aggregation capabilities to transmit less data in WSNs could reduce total energy consumption. However, this calls for an efficient and effective data-centric routing algorithm to facilitate this potential advantage.

In this thesis, our work emphasizes on the construction of an energy-efficient data aggregation tree that possesses good QoS and minimizes the total energy consumption of sensor nodes simultaneously. In the first part of this thesis, we model the data-centric routing problem based on a rigorous mixed integer linear

mathematical formulation, where the objective function is to minimize the total transmission cost, subject to multicast tree and data aggregation constraints. With the advances in sensor network technology, sensor nodes with configurable transmission radius capability would further reduce energy consumption. Thus, the second part of this paper considers the transmission radius assignment of each sensor node and the data-centric routing assignment jointly. The objective function is to minimize the total power consumption together with consideration of construction of a data aggregation tree and sensor node transmission radius assignment. Finally, the effects of data retransmission and maximum end-to-end delay due to data aggregation delay are also considered in order to reflect the tradeoff between the advantages and the costs of data aggregation. We take the properties of QoS in wireless sensor networks as a new energy consumption metric that can not only maintain the traditional transmission delay, but also simultaneously reduce the energy consumption of sensor nodes operating in idle mode. How to construct a data aggregation tree that is energy-efficient and has QoS properties is a complicated problem that needs to be investigated. We conceive a rigorous mathematical formulation, where the objective function is to minimize the total energy consumption of data transmission subject to tree, data retransmission, and maximum end-to-end delay constraints.

The solution approach is based on Lagrangean relaxation in conjunction with novel optimization-based heuristics. With the exceptional properties of Lagrangean relaxation we are able to efficiently solve this complicated optimization problem, and derive an effective algorithm for routing assignments and construct a data aggregation tree simultaneously. Through our computational experiments, we show that the proposed algorithms calculate better solutions than other existing heuristics. When considering QoS routing in WSNs, the proposed algorithms can not only construct a better data aggregation tree in terms of energy consumption, but also maintain good maximum end-to-end delay.

Keywords—Data Aggregation, Data-Centric Routing, Energy-Efficient Routing, Optimization, Lagrangean Relaxation Method, Mixed Integer Linear Programming, Wireless Sensor Networks.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

1.1 Background

Wireless sensor networks (WSNs) are types of nascent technologies that probe and collect environmental information such as temperature, atmospheric pressure, and irradiation by providing ubiquitous sensing, computing, and communication capabilities. Each sensor in a WSN is capable of sensing ambient information in its vicinity and reporting the sensed data. Because of the rapid advances in microprocessor, memory, and radio techniques, the deployment of distributed networks comprising small, inexpensive sensor nodes capable of sensing and wireless communication will soon become reality.

In recent years, WSNs have attracted a great deal of attention for researchers due to their potential for numerous military, environmental detection, and civil applications. For example, on a battle field, a network of sensors could be used to track moving targets, detect chemical gases, or assist in surveillance missions. In some environmental research, sensor networks could gather various geologic parameters of interest. With such data, it is possible for scientists to make some forecasts such as the eruption of volcano or to detect harsh natural phenomena. In a disaster situation, such as the terror attack, if people were equipped with a small badge then sensor networks could locate the exact position of survivors and help rescuers to extricate them rapidly. In all the application scenarios described above, sensors would be typically scattered throughout an area of interest in unattended mode; thus recharging the batteries of a sensor would not be feasible. Energy aware management in WSNs, therefore, becomes an essential issue in order to prolong the lifetime of deployed sensors.

Wireless sensor networks are similar to mobile ad-hoc networks (MANETs) in that both involve multi-hop communications. However, there are two main differences between them. First, the typical communication mode in wireless sensor networks conveys information from multiple data sources to one data sink. This is a kind of reverse-multicast, rather than the communication between any two pair of nodes in MANETs. Second, since data are collected by multiple sensors there must be some redundancy in the data, being transmitted by numerous sources. Transmitting redundant data would rapidly deplete the energy of sensors and result in disconnected network. A data aggregation function, therefore, has been suggested as a particularly useful function for routing in terms of energy consumption in WSNs [2, 3, 8, 9, 12, 18].

A canonical system architecture of wireless sensor nodes is shown in Figure 1-1. A sensor node is composed of four major subsystems [13], namely:

1) **Computing subsystem**: the microcontroller unit (MCU) is responsible for executing the signal processing algorithm, data processing, and the communication protocol.

2) **Communication subsystem**: the leading component in a sensor node is radio frequency communication with which the sensor node can communicate with neighboring nodes and transmit data. This subsystem causes more energy consumption compared against any other component.

3) **Sensing subsystem**: the sensing subsystem is in charge of sensing real world phenomena and translating it into electronic signals. There are two types of

sensor, analog and digital, for measuring different environmental parameters.

4) **Power supply subsystem**: the power supply subsystem consists of a battery and a DC-DC converter. The latter is used to stabilize the voltage of the power provided by battery.

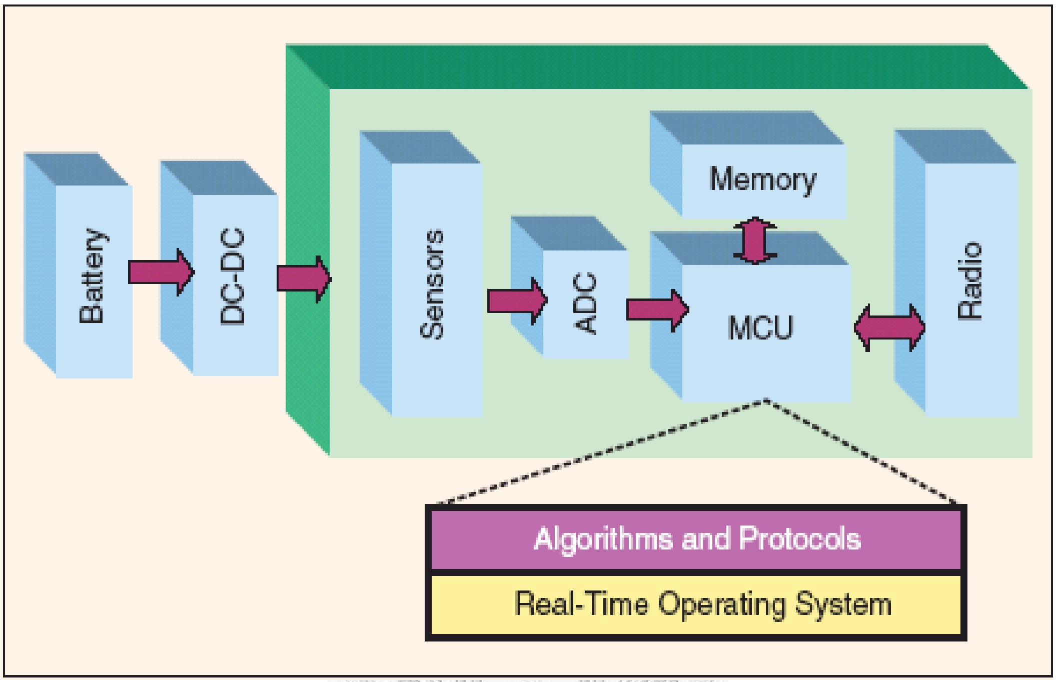


Figure 1-1. System architecture of a typical wireless sensor node [13]

Sensor nodes are highly energy-constrained due to the limitations of hardware and the infeasibility of recharging sensor nodes under severe environments. A great deal of research has focused on how to reduce energy consumption while a sensor is operating. From the perspective of a single sensor node, some conventional low-power design techniques and hardware architectures would be helpful for providing a mechanism where energy conservation of single sensor node is concerned. However, in addition to sensing and transmitting its own data to other nodes, a sensor node is also responsible for relaying and forwarding data from other nodes to the destination. Therefore, enormous numbers of sensor nodes are involved in data routing when sensors report data to the sink node. A low-power design technique for a single sensor node is inadequate for a highly energy-constrained WSN, since the total energy consumption of an entire network must be considered systematically. It is important to adopt a thorough approach that enables an energy-aware design from the underlying hardware platform to the application software and communication protocol. This approach would be useful for conserving not only the energy of an individual sensor, but also of the groups involved in data transmission as well in the whole sensor network. In this paper we consider the energy aware routing (EAR) with respect to operations of entire sensor network and try to minimize the total power consumption of a group of sensors involved in data transmission.

A representative wireless sensor network is shown in Figure 1-2. Sensor nodes are usually scattered in a *sensor field*. When any event occurs, such surging irradiation or temperature decline below a certain threshold, sensor nodes within a specific *sensing range* detect the event and collect data and transmit it to the sink node for further processing. We refer to the sink node as the *data sink* and each sensor node within the sensing range as the *data source* since data are generated from these sensors. The application scenario described above is called *event-driven* because sensors are assigned to detect some particular events. There are two other different applications of wireless sensor networks, namely *periodic* and *query-based*. In *periodic* scenario, sensors probe environmental information periodically and report their measurements back to the sink node. All sensors in this kind of network are necessitated to be synchronized such that they all sense information and report it simultaneously. The query-based scenario, on the other hand, is applied to user-oriented applications. Users can request information from certain area of sensors about subjects they are interested in.

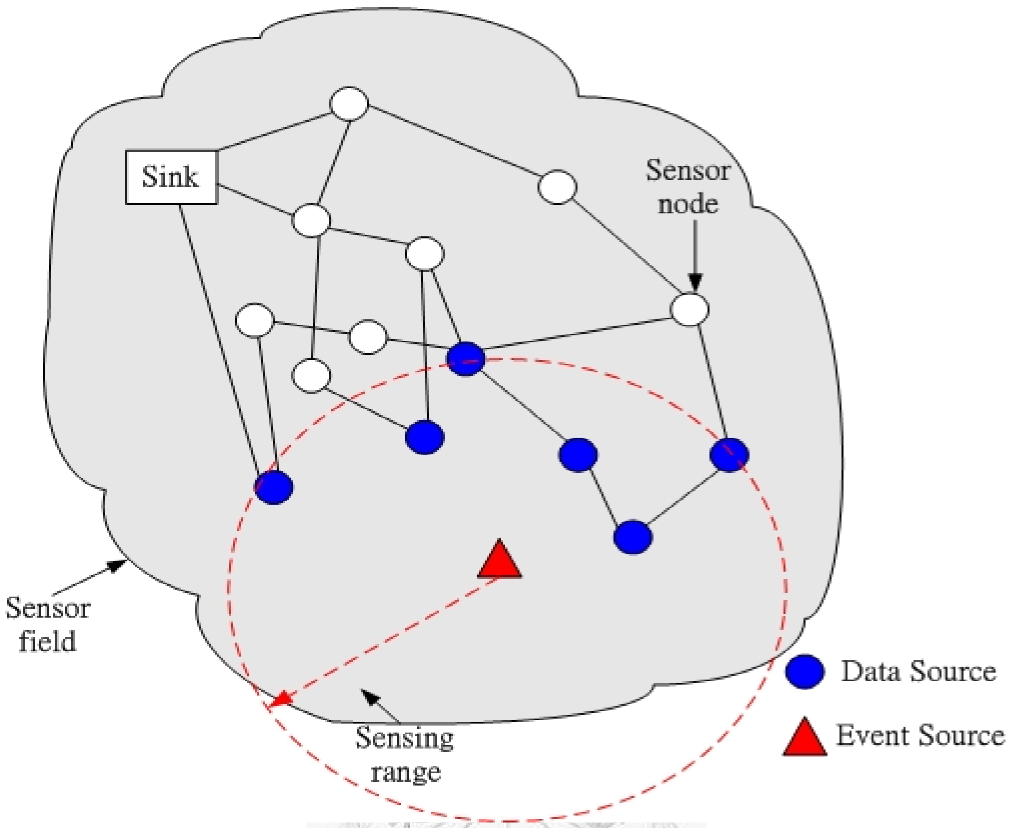


Figure 1-2. A typical wireless sensor network

In the event-driven model, as a specific event occurs raw data are collected and processed before conveying it to data sink. Redundant and useless data are discarded. The local raw data are first combined together in the data source and then the aggregated result is transmitted to the sink node. Interestingly, data are routed along a *reverse multicast tree* where multiple data sources transmit information back to the single data sink. Every non-leaf node on this reverse multicast tree can perform the data aggregation function to summarize the output from upstream data sources. This process is called *data-centric routing*. The operation described above can also be applied to periodic applications where all sensor nodes or parts of them are data sources and periodically report sensed data of interest to the data sink. If all sensor nodes in WSNs are designated as data sources, then the problem of minimum energy consumption for constructing a data aggregation tree becomes a minimum power consumption spanning tree problem.

Data aggregation is the key to the data-centric routing. By combining the data coming from different sources, redundant information can be eliminated; therefore, the total number of transmissions involved in data routing can be reduced significantly. Data aggregation achieves energy-efficient data transmission by processing data as it flows from the data source to the sink node. In addition to redundancy suppression, other possible aggregation functions could be MAX, MIN, and SUM. Data aggregation is application dependent, which means that according to the goal of the application the appropriate data aggregation function should be employed. For example, suppose that a controlled temperature environment is considered, the maximum temperature would need to be monitored. In this paper we assume that every sensor node possesses a data aggregation capability, which transmits a single aggregated packet if it receives multiple input packets to the same data sink. Figure 1-3 shows an illustrative example of data-centric routing, where the maximum temperature is reported to the data sink. The aggregation function is *MAX*. Label *x(y)* at each node represents the local temperature measurement which is *x* while the aggregated (maximum) value so far is *y*. For example, at node 4(7), the maximum temperature up to this node is 7 and its local temperature measurement is 4.

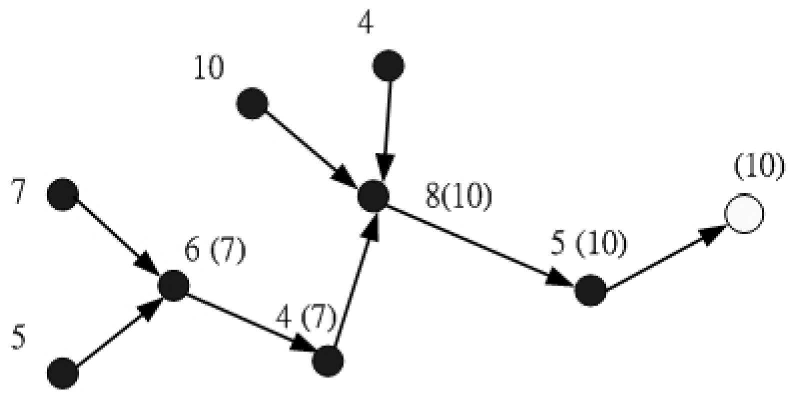


Figure 1-3. An illustrative example of data aggregation

1.2 Motivation

The construction of a data aggregation tree that would enable multiple data sources to transmit sensed data to a single sink node is a hard problem to solve. The tree adopted as the data aggregation tree significantly affects the total power consumption of sensor nodes. When constructing an aggregation tree, three major factors that consume the energy of a sensor node must be considered, namely: the *transmission radius* that each sensor activates; the *retransmission times* incurred on a link; and the *maximum end-to-end delay* between each data source and the data sink.

In WSNs, the transmission power (radius) is associated with the physical distance between the source and the destination. Thus, it is reasonable to assume that the transmission cost associated with each link is identical to the cost of transmission in the opposite direction. By this assumption, the total transmission cost in Figure 1-3 is identical to the multicast tree transmission cost where the root is node (10) and the other nodes are the destinations. In Chapter 2, we propose a *DCR model* that formulates this problem as a mixed integer linear programming (MILP) problem to optimally solve the minimum cost of the multicast tree transmission problem without considering retransmission and end-to-end delay effects. Construction of the minimum cost multicast tree is the well-known Steiner tree problem, which is proven to be the NP-complete [7].

In addition to the energy consumption of data transmission, data retransmission resulting from collisions and the hidden terminal problem in wireless communication also consumes a sensor's energy. The more data an intermediate node aggregate, the greater the number of collisions that will occur at intermediate node, which results in excess energy consumption. Besides the total amount of data that an intermediate node aggregates, many sensor nodes, called the neighbor nodes, whose transmission radii cover the intermediate node would also substantially influence data transmission times between children nodes and the intermediate node. By increasing the number of neighbor nodes, it is obvious that collisions would be very severe at the intermediate node during data transmission from children nodes, and consequently more energy consumption would be incurred at the children nodes.

End-to-end delay is another issue that we should consider. Sustaining satisfactory end-to-end delay implies two important factors about WSNs: energy consumption and quality of service (QoS). Each sensor node in a data aggregation tree should wait for data from the children nodes, and during the waiting time (the maximum end-to-end delay from the farthest leaf node, the sensor node would operate in idle mode). As shown in [13], the energy consumption for a sensor operating in idle mode is slightly less than that of operating in transmission mode. Therefore, the end-to-end delay in WSNs should implicitly be minimized in terms of energy consumption. Also, note that delay is a good QoS metric of WSNs when supporting reports of emergent events or real-time traffic is necessary. In this paper, we consider not only the construction of a data aggregation tree with minimum total power consumption, but also the retransmission effects and maximum end-to-end delay constraint on each sensor node. This is a constrained Steiner tree problem and requires an effective and efficient heuristic to solve it. In the following chapters, three MILP are proposed to formulate the constrained Steiner tree problem and some heuristic are derived to optimally solve it.

1.3 Literature Survey

In this section we survey the design problem of WSNs with data aggregation property. Algorithms for fixed and adjustable transmission radius and different kinds of QoS issues are studied.

1.3.1 Data Aggregation Tree without QoS

Although a great deal of existing research has been conducted to address the routing problem in wireless sensor networks, to the best of our knowledge, no prior work has investigated the handling of maximum end-to-end delay in conjunction with the MAC retransmission effects simultaneously. S. Singh [16] and C. Toh [19] show that by using new power-aware metrics, for example the energy consumed for transmitting per packet, or a shortest-cost routing algorithm based on these new power-aware metrics could reduce cost/packet of routing over shortest hop routing. This motivates us to construct power-aware metrics (a_l in DCR model), instead of hops which are used in [16], as the link cost. Even though the total power consumption is a critical metric for minimum power routing, it has a major disadvantage. Since it can reduce the total power consumption of the overall network, it does not directly reflect the lifetime of each sensor node. In other words, if the minimum power routes are via certain specific nodes, the energy of these nodes will be depleted quickly, so that these energy-drained sensors will be unable to continuously transmit data or sense environmental data. Therefore, the remaining energy of each sensor node is another useful metric to enhance the power-aware routing capability and thereby maximize system lifetime or total power consumption [6, 10, 11, 19].

Krishnamachari [12] devised three interesting suboptimal aggregation heuristics, namely, Shortest Paths Tree (SPT), Center at Nearest Source (CNS), and Greedy Incremental Tree (GIT), respectively. Figure 1-4 is a simple illustration of these three heuristics. Note that the transmission cost on each link is set to be 1. In the SPT scheme, each data source node finds the shortest path back to sink node. Figure 1-4(b) shows the tree generated by the SPT scheme, from which it is clear that SPT cannot find the optimal solution. CNS selects the node nearest to the sink node as the aggregation node and other data sources connect to this aggregation node using the shortest hop path. Figure 1-4(c) shows the final routing assignment by adopting CNS heuristic. In this case, CNS does not achieve the optimal solution, since nodes 2 and 3 can directly send data back to sink node rather than transmit via aggregation node 1.

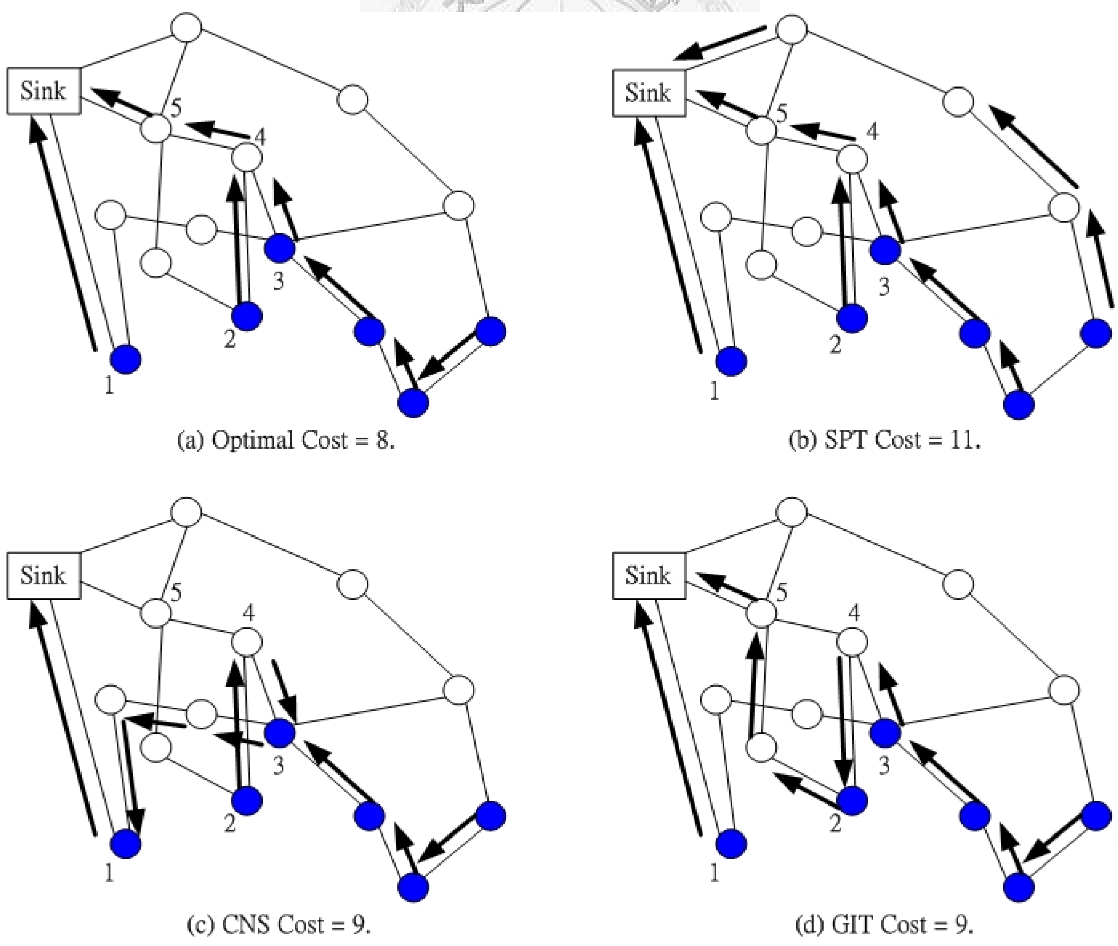


Figure 1-4. A simple illustration of SPT, CNS, and GIT

In GIT scheme, initially the only member of the current tree is only the sink node. Each data source finds the shortest hop path to this current tree and the data source with the minimum hop along with the intermediate nodes on this path are included in this tree. Then, the other source nodes find the shortest hop path to this new tree and the minimum hop source node along with the intermediate nodes on this path are included in the tree. This process is repeated until all source nodes are included in the tree. Although it seems that GIT might overcome the major weakness of CNS, it may not find the optimal solution. Note that how to properly select the path when there are two paths with the same hop distance to the tree will have significant impact on the solution quality of the GIT solution. In Figure 1-4(d), after the nearest node, node 1, connects to the sink node, nodes 2 and 3 are three hops away from the tree consisting of sink node and node 1. If node 2 selects the path through nodes 4 and 5 to reach the sink node then the resultant tree will be the optimal case. Moreover, this work does not deal with MAC retransmission effect and restriction on maximum end-to-end delay.

Fixed transmission radius data-centric routing problem in wireless sensor networks has been studied in existing research described above. The basic idea of fixed transmission radius algorithm is to save energy by reducing number of sensor nodes involved in data aggregation tree, while adjustable transmission radius algorithms tend to estimate power consumption from another point of view. If the transmission radius of sensor node could be configured, it is believed that energy consumption could be further reduced. The power consumption of transmitting data is measured as $r^{\alpha} + c$, where α is a signal attenuation constant (usually between 2 to 4) and c is a positive constant that represents signal processing and r is Euclidean distance between source node and destination node..

J. Carle [5] discusses the tradeoff between power consumption and coverage of transmission node. For long transmission radius (e.g. single-hop transmission), sensor nodes can cover other awake sensor nodes to relay data and also reduce the total number of sensor nodes involved in data transmission. However, with large signal attenuation constant (e.g. 4), long transmission radius incurs significant power consumption that would sacrifice the gain from reduced total number of transmissions. As we can see, the power consumption will increase exponentially as transmission radius increases.

1.3.2 Data Aggregation Tree with QoS

Younis [21] discusses the role that quality of service (QoS) traffic plays in WSNs. The paper presents a thorough discussion and the architectural and operational challenges of handling QoS traffic in sensor networks. Table 1-1 shows some architectural design issues and corresponding primary factors in WSNs.

Table 1-1. Architectural Design Issues [21]

Design Issue	Primary Factors
Network Dynamics	Mobility of node, target, and sink
Node Deployment	Deterministic or Ad Hoc
Node Communications	Single-hop or multi-hop
Data Delivery Models	Continuous, event-driven, query-driven, or hybrid
Node Capabilities	Multi- or single function; homogeneous or heterogeneous capabilities
Data Aggregation/Fusion	In-network (partially or fully) or out-of-network

Youseef [22] augments power-aware routing algorithms by considering QoS in sensor networks, and proposes a power-aware routing algorithm whose objective is to minimize the total power consumption while a reasonable level of QoS is sustained. Youseef considers end-to-end delay from a source node to the sink node as a major QoS factor and takes end-to-end hop counts as its delay metric. The relations of end-to-end delay and energy consumption to the transmission distance are discussed in the experimental results. Although [22] considers delay in power-aware routing, it does not capture another cardinal factor, namely collisions resulting from contending media access control in a wireless environment. Collisions significantly affect total power consumption in terms of power-aware routing, because it consumes more energy of a sensor node in order to transmit data successfully. In this thesis, we take the collision effect into account.

Akkaya [2] differentiates between real-time and non-real-time traffic according to the latency-constrained requirement. A Weighted Fair Queueing (WFQ) mechanism is employed at each sensor node to perform service differentiation and guarantee the end-to-end delay bound. Flow aggregation of real-time traffic results in increasing queueing delay at an aggregation node. Therefore, an adjustment of the bandwidth sharing weight should be made in order to meet end-to-end delay requirements. Although this work tries to maintain end-to-end delay from each node on the tree to the sink node, it only uses a simple heuristic, namely Shortest Path Tree (SPT), to construct the aggregation tree. In this paper we compare the performance of our proposed algorithm with SPT. From the computational results it is clear that our approach outperforms SPT. Note that while wireless sensor networks are limited in bandwidth, the bandwidth reservation mechanism for supporting QoS constrained traffic is impractical unless the data flows are generated in the continuous mode. [2]

also observes that when a low data generation rate is concerned the queueing delay introduced by aggregation becomes negligible. In this paper we assume that sensor networks operate in periodic or event-driven mode, where data generation rate would be low. We, therefore, consider the end-to-end delay as transmission, propagation and retransmission delay. We do not consider the queueing delay.

Solis [17] discusses the impact of timing on data aggregation. Since data flows should be aggregated at intermediate nodes on the aggregation tree, a certain must be incurred. This kind of delay, namely, the maximum end-to-end delay, significantly influences data freshness (the interval between the time of data generation and that of sink node reception). [17] proposes a heuristic, called "*cascading timeouts*", to calculate how long an intermediate node should wait in order to achieve maximum freshness and accuracy of sensed data. The basic idea of cascading timeouts is that a node's timeout, i.e., the time interval it waits to receive data from its children before forwarding aggregated data, is based on its position on the data aggregation tree. Thus, the timeout of a node will occur immediately its parent's. The construction of the aggregation tree is not discussed in [17]. In this paper, the impact of timing in data aggregation tree is considered as a maximum end-to-end delay metric to reflect the interval that each node should wait before data aggregation. Moreover, the maximum end-to-end delay for each node should be minimized, because it is also regarded as a cost, which is the energy consumed by a sensor node operating in idle mode.

V. Annamalai [4] and S. Upadhyayula [20] propose the algorithm for solving the minimum energy convergecast problem which also tries to minimize data latency. These two algorithms construct the data aggregation tree based on greedy approach in which new nodes are iteratively added into the data aggregation tree such that the cost of adding new link to original tree is less. The algorithm then allocates DSSS or FHSS codes to each node on the tree. This paper actually does not tackle the problem of transmission radius assignment. Instead it only takes distance as its criteria to assign transmission radius to each node.

1.4 Proposed Approach

In this paper we proposed three models to precisely describe the data-centric routing problem in WSNs. We first propose an optimization-based heuristic to solve the *fixed transmission radius data-centric routing problems* (*DCR*) in wireless sensor networks. The problem is first formulated as a mixed integer and linear programming (MILP) problem, where the objective function is to minimize the total transmission cost for all multicast groups, subject to multicast tree and data aggregation constraints. In the extension model of DCR, besides routing assignment, we also study the transmission radius assignment of sensor nodes to further reduce total energy consumption. Hence, the *energy-efficient data-centric routing problem* (*EDCR*) in wireless sensor network could be formally defined as minimizing total power consumption subject to reverse-multicast tree, and configurable transmission radius.

Finally, the problem with consideration of data retransmission and the maximum end-to-end delay is then formulated as a MILP problem where the objective function is to minimize the total energy consumption of constructing the data aggregation tree, subject to aggregation tree, transmission radius, data retransmission times, and the maximum end-to-end delay constraints. We propose the Lagrangean relaxation scheme, in conjunction with the optimization-based heuristics to solve these two problems.

From the computational experiments, the proposed solution approaches are superior to the existing heuristics.

1.5 Thesis Organization

The remainder of this paper is organized as follows. In Chapter 2, MILP formulations of the DCR and EDCR problem are proposed. In Chapter 3, a MILP formulation of data-centric wireless sensor networks routing problem with QoS constrain and transmission radius assignment is proposed. In Chapter 4, solution approaches based on Lagrangean relaxation are presented. In Chapter 5, heuristics are developed for calculating good primal feasible solution of these problems. In Chapter 6, the computational results are reported. Finally, in Chapter 7 we present our conclusions and indicate the direction of the future works.

Chapter 2 Problem Formulation of DCR and EDCR

2.1 Problem Description

The problem to be solved is to decide how far the sensor node should turn on its transmission radius so that accordance with that corresponding topology the energy consumption cost of constructing a data aggregation tree can be minimized. The data aggregation tree would be formed after determining the transmission radius of each sensor. Each data source node should be assigned exactly one routing path in order to transmit sensed data to the sink node. The routing assignment for each data source should be carefully chosen, because after determining the routing assignment of all data sources, the union of all routing paths will form the data aggregation tree. Two factors that significantly affect total energy consumption are considered in DCR and EDCR, namely:

1) **Data aggregation capability:** as discussed in Chapter 1, data aggregation can substantially reduce the total energy consumption as it can incorporate many data packets into one single packet while its in-network processing ability is enabled. With the total number of aggregate flows increasing, the great benefit derived from transmitting fewer data packets would be that less transmission energy is needed to transmit data.

2) **Transmission radius assignment**: the power consumption function for transmitting data is defined as $r^{\alpha} + c$. We observe that without properly assigning a transmission radius to each sensor node on the data aggregation tree, more power will be dissipated.

2.2 Problem Formulation

2.2.1 DCR Problem

A data-centric wireless sensor network is modeled as a graph in which sensors are represented as nodes and the arc connecting two nodes indicates that one node is within the other's transmission radius. In WSNs, the transmission power (radius) is associated with the physical distance between the source and the destination. Thus, it is reasonable to assume that the transmission cost associated with each link is identical to the cost of transmission in the opposite direction. By this assumption, the total transmission cost of a data aggregation tree is identical to the multicast tree transmission cost, where the root is the sink node and the source nodes are destinations. In the DCR problem, we consider a topology in which the transmission radius of each sensor is fixed. The effects of data retransmission and end-to-end delay are not considered in the DCR model. The DCR model can be applied to a scenario where sensor nodes can not adjust their transmission radii, and there are few traffic flows to be sent such like event-driven applications; hence the effects of data retransmission and end-to-end delay can be neglected. The summary of problem description of the DCR model is given in Table 2-1.

Table 2-1. Problem description for DCR problem

Given:

- The set of all multicast groups
- The set of data source nodes for each multicast group
- The set of all links in the network
- The set of all candidate paths from the data source to the sink node

- Longest hops along shortest path to reach farthest data source for each multicast group
- Transmission cost of each link with respect to energy consumption

Objective:

To minimize total transmission cost of the data aggregation tree.

Subject to:

- Data aggregation constraint— several data items arriving at the same node will be aggregated and the intermediate node will transmit only one aggregated set of data.
- Routing constraint— each data source node should only select one routing path to send data back to the sink node.
- Tree constraint— the union of routing paths of each data source shall be a tree, namely, a data aggregation tree.

To determine:

- Routing path for each data source
- Total number of data items on each link
- Whether or not a link should be on the data aggregation tree

Table 2-2. Notations of given parameters for the DCR problem

Given parameters	
Notation	Description
G	The set of all multicast groups
D_g	The set of data source nodes for the multicast group g
L	The set of all links on the graph
P_{gd}	The set of candidate paths from the data source node d to the sink node of multicast group g
h_g	The longest distance along the shortest path to reach the farthest data source node for the multicast group g

a_l	Unit power aware transmission cost associated with the link *l*
δ_{pl}	The indicator function, which is 1 if link *l* is on path *p*; and 0 otherwise
n^-	The set of all outgoing links belonging to the node *n*

In this formulation, we generalize the formulation to consider multiple multicast groups, i.e., multiple events. P_{gd} considers all possible paths that source node *d* of multicast group *g* may use. We do not need to generate these paths in advance. In the algorithms proposed in Chapter 5, the arc weight, $(u_{gdl}^2 + u_{gl}^3)$, on each link *l* enables us to find the shortest path by using the Dijkstra's algorithm to identify the path used by data source *d* of multicast group *g*.

The decision variables for the wireless sensor networks routing problem are denoted as follows.

Table 2-3. Notations of decision variables for DCR problem

Decision Variables	
Notation	Description
C_l	Number of data units transmitted through the link *l*
y_{gl}	1 if the multicast group *g* uses the link *l* and 0 otherwise
x_{gpd}	1 if multicast group *g* uses path *p* to reach source node *d* and 0 otherwise

The data-centric routing problem in wireless sensor networks is then formulated as the following combinatorial optimization problem (IP1).

Objective function:

$$Z_{IP1} = \min \sum_{l \in L} a_l C_l \qquad \text{(IP1)}$$

subject to:

$$\sum_{g \in G} y_{gl} \le C_l \qquad \forall l \in L \qquad \text{(2-1)}$$

$$C_l \in \{0, 1, 2, 3, ..., |G|\} \qquad \forall l \in L \qquad \text{(2-2)}$$

$$\sum_{p \in P_{gd}} x_{gpd}\delta_{pl} \le y_{gl} \qquad \forall g \in G, l \in L, d \in D_g \qquad (2\text{-}3)$$

$$y_{gl} = 0 \text{ or } 1 \qquad \forall g \in G, l \in L \qquad (2\text{-}4)$$

$$\sum_{l \in L} y_{gl} \ge \max\left\{h_g, \left|D_g\right|\right\} \qquad \forall g \in G \qquad (2\text{-}5)$$

$$\sum_{d \in D_g}\sum_{p \in P_{gd}} x_{gpd}\delta_{pl} \le \left|D_g\right| y_{gl} \qquad \forall g \in G, l \in L \qquad (2\text{-}6)$$

$$\sum_{p \in P_{gd}} x_{gpd} = 1 \qquad \forall g \in G, d \in D_g \qquad (2\text{-}7)$$

$$x_{gpd} = 0 \text{ or } 1 \qquad \forall p \in P_{gd}, g \in G, d \in D_g \qquad (2\text{-}8)$$

$$\sum_{l \in n^-} y_{gl} \le 1 \qquad \forall g \in G, n \in N. \qquad (2\text{-}9)$$

The objective function of (IP1) is to minimize the total data transmission cost of data aggregation tree, which is equal to the total multicast routing cost. Constraint (2-1) requires that the number of multicast groups adopting link *l* as their multicast tree should be less then or equal to the number of data units transmitted through link *l*. Constraint (2-2) requires that number of data units on link *l* be at most the cardinality of *G*, registering sensor nodes can aggregate data belonging to the same multicast group. Constraint (2-3) requires that if one path is selected for the group *g* destined to the destination *d*, the path must also be on the tree adopted by the multicast group *g*.

Constraints (2-4) and (2-5) require that number of links on the multicast tree adopted by the multicast group *g* be at least the maximum of h_g and the cardinality of D_g. Note that both h_g and D_g are legitimate lower bounds of the number of links on the multicast tree adopted by the multicast group *g*. For example, if there are two destination nodes in multicast group *g*, then $|D_g|$ is equal to *2*. An illustrated example is given in Figure 2-1, where the multicast group source node is 1 and the destination nodes are 2 and 6. Obviously, h_g is 3, since the farthest destination node is node 6, which is three hops away from the source node. From the computational experiments, we find that introducing Constraint (2-5) significantly improve the solution quality. Note that $|D_g|$ and h_g could be calculated in advance. We propose a revised Dijkstra's algorithm, denoted as *Calculate_h*$_g$, to compute h_g for each multicast group g by setting each arc weight to be one.

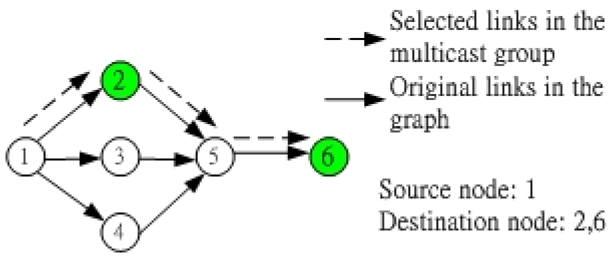


Figure 2-1. An illustrated example of Constraint (2-5)

Algorithm Calculate_h$_g$

begin

initialize all arc weight to be 1;

for g := 1 to $|G|$ do

begin

initialize $h_g[g]$:= 0;

for d := 1 to $|N|$ do

begin

if $d \in D_g$ then

run Dijkstra's shortest path algorithm to determine hops (hop[d]) between sink node and destination d.

if hop[d] > $h_g[g]$ **then** $h_g[g]$:= hop[d];

end;

end;

end;

The left-hand term of Constraint (2-6) calculates the number of paths destined for data source nodes that pass through link l for a multicast group. The right-hand term of Constraint (2-6) is at most $|D_g|$. If the union of the paths destined for the data source nodes does exist a cycle, and this cycle contains link l, then Constraint (2-6) would not be satisfied, since there would be too many paths to pass through this link. In other words, Constraint (2-6) is to restrict that the union of the paths destined for data source nodes does not contain a cycle. Constraints (2-7) and (2-8) require that any multicast group g selects exactly one routing path destined for its destination d. Constraint (2-9) is the outgoing constraint, which means that each node on the aggregation tree should only have one outgoing link. By enforcing Constraints (2-6), (2-7), (2-8), and (2-9) the union of the routing paths shall be a tree.

2.2.2 EDCR Problem

We first show the notations of EDCR model.

Table 2-4. Notations of given parameters for EDCR problem

Given parameters	
Notation	Descriptions
N	The set of all sensor nodes
P_{sq}	The set of all candidate paths that the data source node *s* connects to the sink node *q*
S	The set of all data source nodes
h	Longest hops of shortest path to reach farthest data source node
$\delta_{p(n,k)}$	The indicator function which is 1 if the link (*n*,*k*) is on the path *p* and otherwise 0
d_{nk}	Euclidean distance between the node *n* and the node *k*
q	The data sink node
R_n	The set of all possible transmission radii that the node *n* can adopt
$e_n(r_n)$	Energy consumption function of the node *n*, which is a function of node’s transmission radius

In EDCR model, we model the link *l* as the node pair (*n*,*k*). *n* is the origin node of the link *l* and *k* is the termination node of link *l*. As the node *k* is within transmission radius of the node *n*, link (*n*,*k*) will exist. The decision variables used in EDCR model are denoted as follows.

Table 2-5. Notations of decision variables for EDCR problem

Decision Variables	
Notation	Descriptions
r_n	Transmission radius of the node *n*
$y_{(n,k)}$	1 if the link (*n*,*k*) is used by the aggregation tree
x_{sp}	1 if the data source node *s* uses the path *p* to reach the sink node *q*

The EDCR is then formulated as the following combinatorial optimization problem (IP2).

Objective:

$$Z_{IP2} = \min \sum_{n \in N} e_n(r_n) \quad \text{(IP2)}$$

subject to:

$$\sum_{p \in P_{sq}} x_{sp} \delta_{p(n,k)} \le y_{(n,k)} \qquad \forall n,k \in N, s \in S \qquad \text{(2-10)}$$

$$\sum_{n \in N} \sum_{k \in N} y_{(n,k)} \ge \max\{h, |S|\} \qquad \text{(2-11)}$$

$$\sum_{s \in S} \sum_{p \in P_{sq}} x_{sp} \delta_{p(n,k)} \le |S| \cdot y_{(n,k)} \qquad \forall n,k \in N \qquad \text{(2-12)}$$

$$\sum_{k \in N} y_{(n,k)} \le 1 \qquad \forall n \in N \qquad \text{(2-13)}$$

$$y_{nk} d_{nk} \le r_n \qquad \forall n,k \in N \qquad \text{(2-14)}$$

$$r_n \in R_n \qquad \forall n \in N \qquad \text{(2-15)}$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n,k \in N \qquad \text{(2-16)}$$

$$\sum_{p \in P_{sq}} x_{sp} = 1 \qquad \forall s \in S \qquad \text{(2-17)}$$

$$x_{sp} = 0 \text{ or } 1 \qquad \forall s \in S,\ p \in P_{sq}. \qquad \text{(2-18)}$$

The objective function of (IP2) is to minimize total power consumption of the data aggregation tree for transmitting data to the sink node**.** Constraint (2-10) requires that if one path p is selected for the source node s to reach the sink node q, the path must also be on the tree. This constraint also enforces that if the link (n, k) is on the path p adopted by source node s to reach sink node, then $y_{(n,k)}$ must be 1. Constraint (2-11) and (2-16) require that total number of links on the aggregation tree be at least the maximum of h and the cardinality of S. Just as (IP1) in Section 2.2.1, introducing constraint (2-11) is to improve the solution quality.

Constraint (2-12) is to restrict that the union of the routing paths destined for data source nodes contains a cycle just as Constraint (2-6) in (IP1). Constraint (2-13) is the outgoing link constraint. All intermediate nodes on the aggregation tree should have only one outgoing link (e.g. node 4 has two incoming link and only one outgoing link in Fig. 1-4(a)). Constraint (2-17) and (2-18) require that any data source adopts only one path destined for sink node in aggregation tree. By enforcing Constraints (2-12), (2-13), (2-17), and (2-18) the union of the paths shall be a reverse multicast tree.

Constraint (2-14) is a transmission radius coverage constraint. This constraint enforces that if the link (n,k) is used by aggregation tree, the transmission radius of node n should be large enough in order to cover node k. Constraint (2-15) indicates the set of possible transmission radii for sensor node, which is a discrete and finite set. By enforcing Constraints (2-14) and (2-15), we ensure that every link on the aggregation tree is covered within the transmission radius of the origin node of the link.

After presenting the mathematical formulation of the EDCR model, we could summarize the major difference between the DCR and EDCR model. In the DCR model, after the maximum transmission radius is given, the topology of whole network can be constructed. Hence, the data centric aggregation algorithm developed for DCR model could also be applied to wired sensor network when a_l represents the link cost of physical link. On the other hand, in the EDCR model, the transmission radius of sensor node is also a decision variable. In other words, network topology needs to be determined by the transmission radius assignment of the sensor node. Such transmission radius assignment makes EDCR model more general than the DCR model.

Chapter 3 Data Aggregation Tree with QoS Routing

3.1 Problem Description

The problem to be solved is to decide how far the sensor node should turn on its transmission radius so that accordance with the corresponding topology the total energy consumption for constructing a data aggregation tree can be minimized. The data aggregation tree is formed after determining the transmission radius of each sensor. Each data source node should be assigned exactly one routing path in order to transmit sensed data to the sink node. The routing assignment for each data source should be carefully chosen, because after determining the routing path of all data sources, the union of all routing paths will form the data aggregation tree. Three factors that significantly affect total energy consumption are considered in this model, namely:

1) **Data aggregation capability:** as discussed in Chapter 1, data aggregation can substantially reduce the total energy consumption as it can incorporate many data packets into one single packet while its in-network processing ability is enabled. With the total number of aggregate flows increasing, the great benefit derived from transmitting fewer data packets would be that less transmission energy is needed to transmit data.

2) **MAC layer retransmission**: interference in wireless communication and the hidden terminal problem play important roles in packet retransmission times as well as increasing energy consumption. In wireless communication, data retransmission times are affected by the total number of sensor nodes whose transmission radius covers the receiver. The more flows that the intermediate

node on the aggregation tree are aggregated, the higher the probability that the sender will incur data retransmissions. Thus, it would be more appropriate to take the MAC layer retransmission times into account when deciding the transmission radius of a sensor and constructing an aggregation tree in wireless sensor networks.

3) **End-to-end delay from the leaf node to the sink node**: aggregation extends the delay at the relay nodes and can thus complicate the handling of latency-constrained data [20, 23]. We analyze the delay metric and apply it to this model. Moreover, we think of end-to-end delay as an energy consumption factor, since sensor nodes operating in idle mode also consume a lot of energy [13].

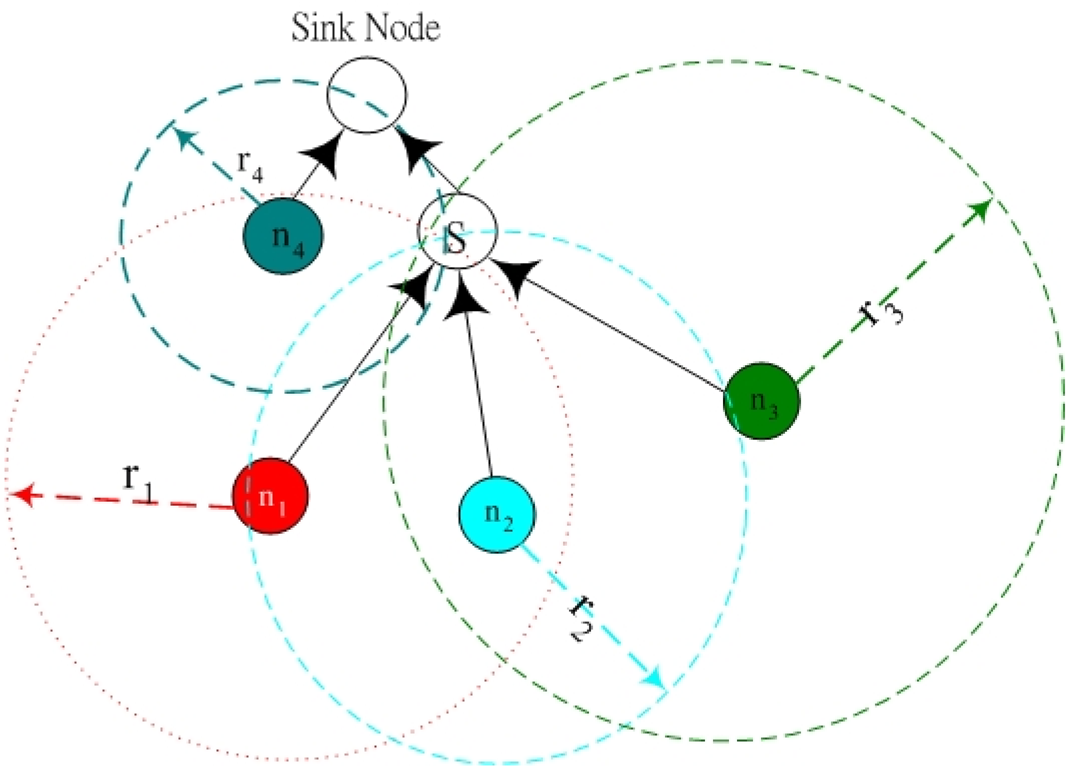


Figure 3-1. Tradeoff between data aggregation and data retransmission times

Figure 3-1 shows the tradeoff between the benefit of aggregating as many flows as possible and the cost of retransmitting data as the number children nodes on the aggregation tree increases. Obviously, when node *S*, which is the receiver of three children nodes, aggregates more flows sent from other nodes, each sender will suffer severe collisions resulting in more retransmission times in order to send data successfully to the receiver. Collisions at the receiver side are a characteristic of wireless transmissions and a cardinal issue that affects the energy consumption of sensor nodes when transmitting data to the sink node. In this paper, we discuss the impacts on retransmission of aggregating data, and achieve the balance between aggregating benefits and data retransmission costs.

The analysis of retransmission is conducted as follows. First of all, we assume that each sensor node is equipped with a CSMA-CA compatible transceiver. Based on the analysis in [15], we derive the mean retransmission time of a sender. Our perspective is that each transmission conforms to Geometric distribution and each sensor node generates data packets that follow Poisson distribution with a certain rate, λ. Successful transmission of data from a sender to a receiver is influenced by the number of sensor nodes whose transmission radius covers the receiver. As shown in Figure 3-1, we find that node *S* is covered by four sensor nodes where three of them are its children nodes and another is the neighbor node. By considering the receiver side collisions in terms of the communication radius of sensor nodes, the hidden terminal problem is also implicitly contemplated. We then derive the probability of successfully transmitting data from node *n* to node *k* is:

$$\textit{Average Retransmission Times}_{(n,k)} = \frac{1}{p_{success(n,k)}} = \frac{1}{e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}}. \quad (3\text{-}1)$$

The definitions of notations are given in Section 3.2. The meaning of (3-1) is the mean value of the Geometric distribution where the successful transmission probability, say $p_{success}$, is that no data transmission is occurring at any node whose transmission radius covers receiver node k within the interval of $RTS + SIFS + 2\theta$.

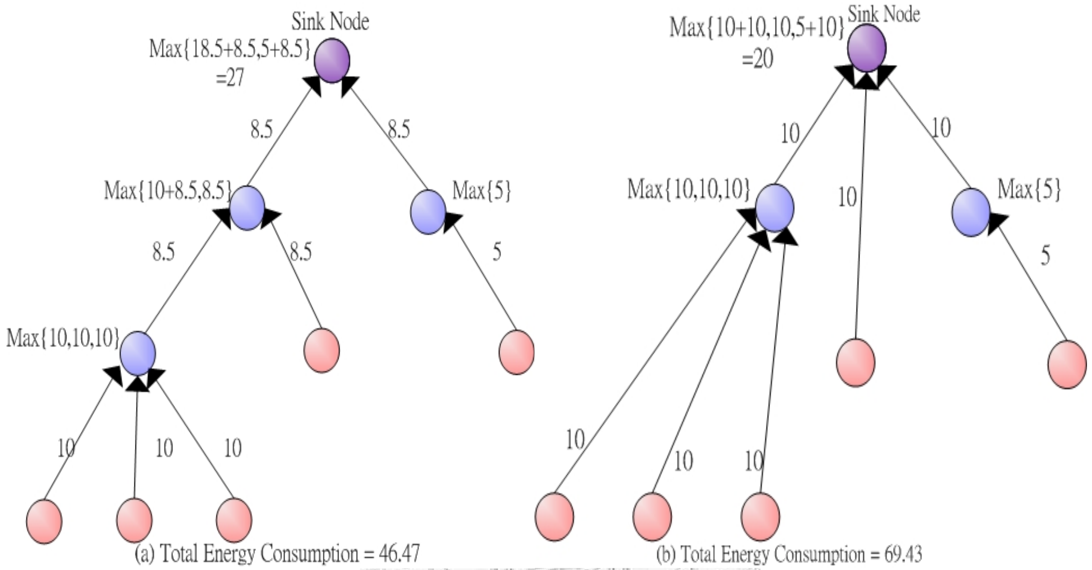


Figure 3-2. Tradeoff between maximum end-to-end delay and transmission radius

Figure 3-2 shows the tradeoff between maximum end-to-end delay and transmission radius in data aggregation tree. In Figure 3-2 (a) we can observe that the maximum end-to-end delay is higher than (b), but the transmission radius of most sensors in (a) is smaller than (b). On the other hand, the maximum end-to-end delay is low in (b), but the average of transmission radius is large. As the total energy consumption of data aggregation tree is sum of transmitting energy and idle energy, we should construct the data aggregation tree that balances the energy consumption caused by maximum end-to-end delay and data transmission.

3.2 Problem Formulation

In this section, we consider a topology that transmission radius of each sensor should be determined and effects of retransmission and end-to-end delay are considered. This model can be applied to periodic application scenario where each sensor periodically reports information to the sink node. Besides the energy consumption of data retransmissions, end-to-end delay for each sensor should be considered as another energy consumption factor under periodic application. Before aggregating data coming from children nodes, each sensor nodes must defer its data transmission until all data are received. During the waiting time sensor operates in idle mode in which considerable energy would be consumed. Therefore, it is reasonable to take end-to-end delay as an energy consumption factor. In this problem total energy consumption, including data transmission, retransmission, and operating energy in idle mode, is minimized. The summary of problem description is listed as below.

Table 3-1. Problem description for the model with QoS routing

Given:

- The set of all sensor nodes
- The set of all candidate paths for each data source to reach sink node
- The set of all data sources
- Longest hops along shortest path from sink node to reach farthest data source
- An arbitrary large number M
- The maximum end-to-end delay B
- The maximum number of retransmission times T on each link
- Distance between each sensor node

- Transmission time for transmitting one data packet
- Transmission time for RTS, CTS, ACK frame
- Waiting time for SIFS, DIFS
- Average packet arrival rate for each sensor node on data aggregation tree
- Maximum propagation delay for transmission data packet
- The sink node *q*
- The set of all possible transmission radii that a sensor node can adopt
- Energy consumption function whish is a function of sensor node's transmission radius

Objective:

To minimize total energy consumption on data aggregation tree

Subject to:

- Routing constraint — each data source node should only select one routing path to send data back to the sink node.
- Tree constraint — the combination of routing path of each data source shall be a tree, namely data aggregation tree.
- Retransmission constraint — for data transmission on each link, there would be a certain retransmissions in order to transmit data successfully.
- Maximum end-to-end delay constraint — the maximum end-to-end delay of each sensor nodes on data aggregation tree should be minimized in order to conserve energy consumption while sensor operates in idle mode.
- Number of neighbors constraint — the total number of sensor nodes whose transmission radius covers a sensor node should be considered.

To determine:

- Routing path for each data source
- Transmission radius for each sensor node

- Whether a link should be on the data aggregation tree - Maximum end-to-end delay for each sensor node on data aggregation tree - The data aggregation tree

Table 3-2. Notations of given parameters for model with QoS routing

Given Parameters	
Notation	Description
N	The set of all sensor nodes
P_{sq}	The set of all candidate paths that the data source node s connect to the sink node q
S	The set of all data source nodes
h	Longest distance of shortest path to reach farthest data source node
M	An arbitrary large number, $M \geq 1$
A	Maximum link delay
B	Maximum end-to-end delay
$\delta_{p(n,k)}$	The indicator function which is 1 if the link (n, k) is on the path p and 0 otherwise
d_{nk}	Euclidean distance between the node n and the node k
t_{data}	Transmission time for transmitting a data packet
RTS	Transmission time for RTS frame
CTS	Transmission time for CTS frame
$SIFS$	Short inter-frame space time
$DIFS$	Distributed inter-frame space time
Θ	Maximum propagation delay for transmitting data packet
λ	Packet arrival rate
q	The sink node
R_n	The set of all possible transmission radii that the node n can adopt, this is a discrete set
$e_n(r_n)$	Energy consumption function of the node n, which is a function of sensor's transmission radius
E_{idle}	Energy consumption when sensor nodes are operating in idle mode
$\overline{B}$	Average random backoff time
$\overline{N}$	Average network allocation vector (NAV)

Table 3-3. Notations of decision variables for model with QoS routing

Decision Variables	
Notation	Descriptions
x_{sp}	1 if the data source node s uses the path p to reach the sink node q
$y_{(n,k)}$	1 if the link (n, k) is on the tree
r_n	Transmission radius of the node n
$l_{(n,k)}$	Data transmission delay from the node n to the node k
m_n	Maximum end-to-end delay from leaf nodes to the node n on data aggregation tree
z_{nk}	1 if the node k is covered within transmission radius of the node n
c_{nk}	Retransmission times of the node n to transmit data to the node k

Objective function:

$$Z_{IP3} = \min \sum_{n \in N} \left(\mathrm{t}_{\mathrm{data}} + (RTS \sum_{k \in N} c_{nk}) \right) \cdot e_n(r_n) + m_n \cdot E_{idle} \tag{IP3}$$

subject to:

$$\sum_{p \in P_{sq}} x_{sp} \delta_{p(n,k)} \le y_{(n,k)} \qquad \forall n, k \in N,\ s \in S \tag{3-2}$$

$$\sum_{n \in N} \sum_{k \in N} y_{(n,k)} \ge \max\{h, |\mathrm{S}|\} \tag{3-3}$$

$$\sum_{s \in S} \sum_{p \in P_{sq}} x_{sp} \delta_{p(n,k)} \le |S| \cdot y_{(n,k)} \qquad \forall n, k \in N \tag{3-4}$$

$$\sum_{p \in P_{sq}} x_{sp} = 1 \qquad \forall s \in S \tag{3-5}$$

$$\sum_{k \in N} y_{(n,k)} \le 1 \qquad \forall n \in N \tag{3-6}$$

$$\frac{r_n - d_{nk}}{M} \le z_{nk} \qquad \forall n, k \in N \tag{3-7}$$

$$z_{nk} d_{nk} \le r_n \qquad \forall n, k \in N \tag{3-8}$$

$$y_{(n,k)} \le z_{nk} \qquad \forall n, k \in N \tag{3-9}$$

$$(m_k + l_{(k,n)}) - A(1 - y_{(k,n)}) \le m_n \qquad \forall n,k \in N \qquad (3\text{-}10)$$

$$l_{(n,k)} = \frac{(e^{-\lambda(DIFS)\sum_{j\in N} z_{jn}}(RTS + SIFS + CTS + \overline{B}) + DIFS + \overline{N})}{e^{-\lambda(DIFS)\sum_{j\in N} z_{jn}} e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}} - \overline{N} \qquad \forall n,k \in N \quad (3\text{-}11)$$

$$c_{nk} \ge \frac{e^{-(1-y_{(n,k)})M}}{e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}} \qquad \forall n,\, k \in N \qquad (3\text{-}12)$$

$$x_{sp} = 0 \text{ or } 1 \qquad \forall s \in S,\ p \in P_{sq} \qquad (3\text{-}13)$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n,\, k \in N \qquad (3\text{-}14)$$

$$z_{nk} = 0 \text{ or } 1 \qquad \forall n,\, k \in N \qquad (3\text{-}15)$$

$$r_n \in R_n \qquad \forall n \in N \qquad (3\text{-}16)$$

$$r_n \ne 0 \qquad \forall n \in S \qquad (3\text{-}17)$$

$$c_{nk} \in \{0,1,2,....,T\} \qquad \forall n,\, k \in N \qquad (3\text{-}18)$$

$$B \ge m_n \ge 0 \qquad \forall n \in N \qquad (3\text{-}19)$$

$$A \ge l_{(n,k)} \ge RTS + SIFS + CTS + \overline{B} + DIFS \qquad \forall n,\, k \in N\,. \qquad (3\text{-}20)$$

The objective function of (IP3) is to minimize total energy consumption of the data aggregation tree for transmitting data to sink node and waiting for data aggregation**.** Constraint (3-2) requires that if the path *p* is selected for the source node *s* to reach the sink node *q*, the path must be on the tree. This constraint also enforces that if the link (*n*, *k*) is on the path *p* adopted by the source node *s* to reach the sink node, then $y_{(n,k)}$ must be 1. Constraint (3-3) and (3-14) require that total number of links on the aggregation tree is at least the maximum of *h* and the cardinality of *S*. Note that both *h* and |*S*| are legitimate lower bound on the total number of links on a aggregation tree. Introducing constraint (3-3) will significantly improve the solution quality. |*S*| and *h* could be calculated in advance. The explanation of this legitimate lower bound is shown in Figure 2-1.

The left-hand term of constraint (3-4) calculates the number of paths, which are destined for the sink node and passing through the link l on aggregation tree. The right-hand term of constraint (3-4) is at most $|S|$. When the union of the paths destined for the sink node does exist a cycle, and this cycle contains link l, then constraint (3-4) would not be satisfied since there would be many paths pass through this link. Constraint (3-5) and (3-13) require that any data source adopts only one routing path destined for sink node. Constraint (3-6) is the outgoing link constraint. All intermediate nodes on the aggregation tree should have only one outgoing link. The illustrative example of constraint (3-6) is shown in Figure 3-3. Constraint (3-4), (3-5), (3-6), and (3-13) enforce that the union of all routing paths would be a tree.

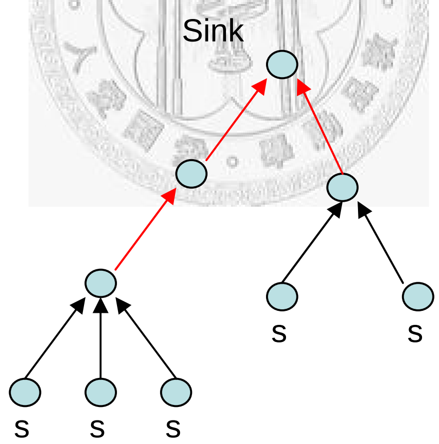


Figure 3-3. An illustrative example of constraint (3-6).

Constraint (3-7) and (3-8) are the number of neighbors constraints. If $r_n \geq d_{kn}$, z_{nk} should be equal to 1 and 0 otherwise. Using z_{nk} we can calculate the total number of sensor nodes whose transmission covers sensor node k, or the total number of sensor nodes covered by transmission radius of sensor node n. These two constraints are complementary as shown in Table 3-4. By jointly enforcing constraint (3-7) and (3-8) we can model the relationship described above through decision variable z_{nk}.

Table 3-4. Explanation of constraint (3-7) and (3-8).

Constraint / Condition	Constraint (3-7)	Constraint (3-8)
If $r_n \geq d_{kn}$	$z_{nk} = 1$	$z_{nk} = 0$ or 1
If $r_n < d_{kn}$	$z_{nk} = 0$ or 1	$z_{nk} = 0$

Constraint (3-9) is a necessary constraint that relates decision variable $y_{(n,k)}$ to z_{nk}. If $y_{(n,k)}$ equals to 1 then z_{nk} also must be 1. Also note that constraint (3-9) implicitly obligates that if sensor nodes were not on the data aggregation tree, they would not choose any link emanating from them as a link used by data aggregation tree, since this behavior would increase the cost of objective function (IP3). Constraint (3-10) is the maximum end-to-end delay from leaf nodes of the aggregation tree to intermediate node *n*. Data aggregation schema in WSNs has a major characteristic that intermediate node on tree should wait a suitable delay time for aggregating all data items coming from children which may have subtree rooted at itself. This is a recursive relation between an intermediate node and its children nodes. Therefore, the maximum end-to-end delay of intermediate node *n* on aggregation tree is the maximum delay of its children plus link delay. This recursive relation then goes down along the tree until leaf nodes are reached. The illustration of maximum end-to-end delay is shown in Figure 3-4.

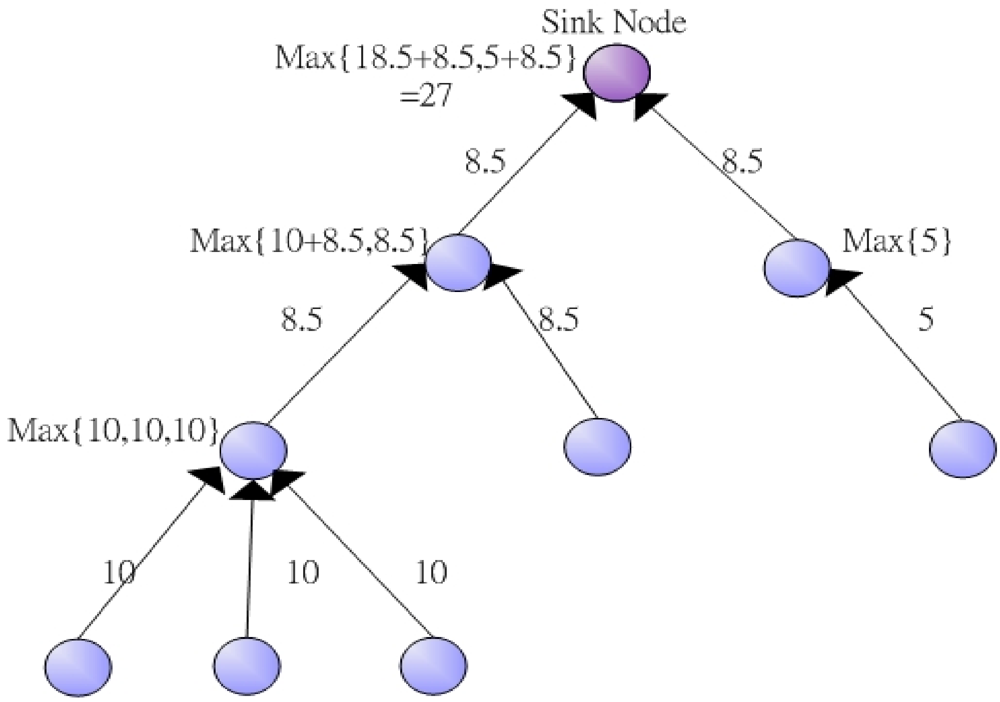


Figure 3-4. Illustrative example of Constraint (3-10).

Constraint (3-11) is the calculation function of link transmission delay. Constraint (3-12) is the calculation function of link retransmission times. Constraint (3-16) restricts that the set of possible transmission radii that node *n* can adopt is discrete and finite set. Constraint (3-17) enforces that each data source node should turn on its transmission radius. The transmission radius of each source node can not be 0. Constraint (3-18) is the bounding constraint of retransmission times. The bounding value is related to maximum end-to-end delay or can be obtain according to specification of standard. Constraint (3-19) is the lower bound and upper bound of maximum end-to-end delay. Constraint (3-20) enforces the bounding constraint on the transmission delay of each link.

For convenience of applying our solution approach to this model, we make some transformations on constraint (3-11) and (3-12) in order to make (IP3) solvable.

Constraint (3-11) can be approximated by:

$$
\begin{aligned}
l_{(n,k)} &= \frac{(e^{-\lambda(DIFS)\sum_{j\in N} z_{jn}}(RTS+SIFS+CTS+\overline{B})+DIFS+\overline{N})}{e^{-\lambda(DIFS)\sum_{j\in N} z_{jn}}\, e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}} - \overline{N} \\
&\cong \frac{e^{0.115+(0.017-\lambda(DIFS))\sum_{j\in N} z_{jn}}(RTS+SIFS+CTS+330)}{e^{-\lambda(DIFS)\sum_{j\in N} z_{jn}}\, e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}} \\
&= \frac{e^{0.115+0.017\sum_{j\in N} z_{jn}}(RTS+SIFS+CTS+330)}{e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}}
\end{aligned}
$$

By adopt this function to approximate original delay function, we can guarantee that the error under five percent can be achieved. In addition to small error, the approximation function overestimates the link delay which means a good approximation from the perspective of engineering. The comparison of approximate and original function is depicted in Appendix.

We then take natural logarithm on both sides in order to make this function solvable.

$$
\begin{aligned}
\ln(l_{(n,k)}) &= \ln\left(RTS+SIFS+CTS+330\right)+0.115+0.017\sum_{j\in N} z_{jn} \\
&\quad + \lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}.
\end{aligned} \tag{3-21}
$$

For constraint (3-12), we take natural logarithm on both side:

$$
\ln(c_{nk}) \geq \ln\left(\frac{e^{-(1-y_{(n,k)})M}}{e^{-\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}}}\right) \Rightarrow \ln(c_{nk}) \geq \lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk} - M + My_{(n,k)}. \tag{3-22}
$$

3.3 Extension to the Model without Aggregation

In some application scenarios of wireless sensor networks, the data sink node needs all sensed data of interest. Therefore, data is not necessary to be aggregated during data transmission. The model of data aggregation with QoS routing can be easily extended to the model without aggregation capabilities, where data will not be aggregated in intermediate nodes. It only requires additional flow conservation constraints in order to transform original model into extended model. The flow conservation constraint is as follows:

$$\sum_{s\in S}\sum_{p\in P_{sq}}\sum_{k\in N} x_{sp}\delta_{p(k,n)} = \sum_{s\in S}\sum_{p\in P_{sq}}\sum_{j\in N} x_{sp}\delta_{p(n,j)} \qquad \forall n \in N \qquad (3\text{-}23)$$

Constraint (3-22) is flow conservation constraint. The meaning of constraint (3-23) is that the total incoming flows of each node should equal to the total outgoing flows. With constraint (3-23) we can restrict that each sensor node needs to send all received data to its parent node without any aggregation operation. In addition to flow conservation constraint, the objective function should be modified slightly in order to reflect more energy consumption due to no aggregation during data transmission. The modified objective function is as follows:

$$\min \sum_{n\in N}\left(H_n + (RTS\sum_{k\in N} c_{nk}) \right)\cdot e_n(r_n) + m_n \cdot E_{idle}$$

where $H_n = \sum_{s\in S}\sum_{p\in P_{sq}}\sum_{j\in N} x_{sp}\delta_{p(n,j)} \qquad \forall n \in N$.

By adding above constraints and modifications, we can transform the original model into the model without aggregation capabilities.

Chapter 4 Solution Approach

4.1 Lagrangean Relaxation Method

One of the most computationally useful ideas in the 1970s is the observation that many hard integer programming problems can be viewed as easy problems complicated by a relatively small set of side constraints. This technique is so-called decomposition which is an invaluable methodology to conquer many complicated problems in the field of computer science, industrial engineering, and operation research. Lagrangean relaxation method is one of the decomposition techniques. By dualizing the side, or complicated, constraints we form a Lagrangean relaxation problem that is easy to solve and whose optimal value is a lower bound (for the minimization problem) on the optimal value of the original problem. To obtain the best lower bound, we need to choose the appropriate Lagrangean multiplier for Lagrangean multiplier problem so that the optimal value of the Lagrangean subproblem is as large as possible. We can solve the Lagrangean multiplier problem in a variety of ways. The subgradient optimization technique is possibly the most popular technique for solving the Lagrangean multipliers problem.

By decomposing the original problem into several easily solvable subproblems, Lagrangean relaxation can solve the subproblems that we have decomposed as stand-alone problems. In decomposed problems, Lagrangean relaxation solves core subproblems as stand-alone models. This solution approach permits us to exploit any well-know efficient algorithm for solving the subproblems. The Lagrangean relaxation method, therefore, can be used to solve optimization problems such as integer programming, non-linear programming, mixed integer linear programming, and combinatorial optimization problems. This approach has led to dramatically improve algorithms for a large number of critical and difficult problems in the areas of routing assignment, location, scheduling assignment and set covering.

Figure 4-1 illustrates the procedure of Lagrangean Relaxation, while Figure 4-2 shows the detailed procedure of Lagrangean relaxation.

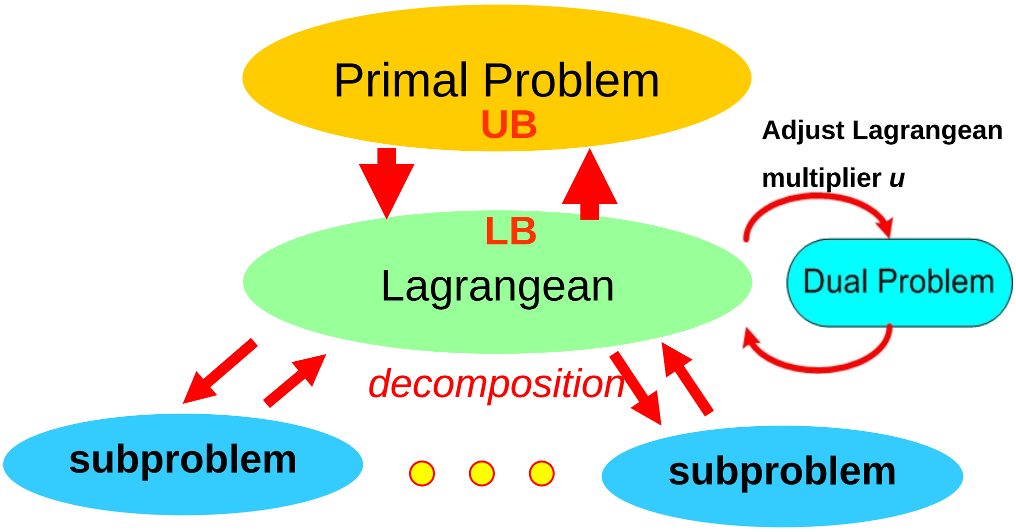


Figure 4-1. An illustration of Lagrangean relaxation

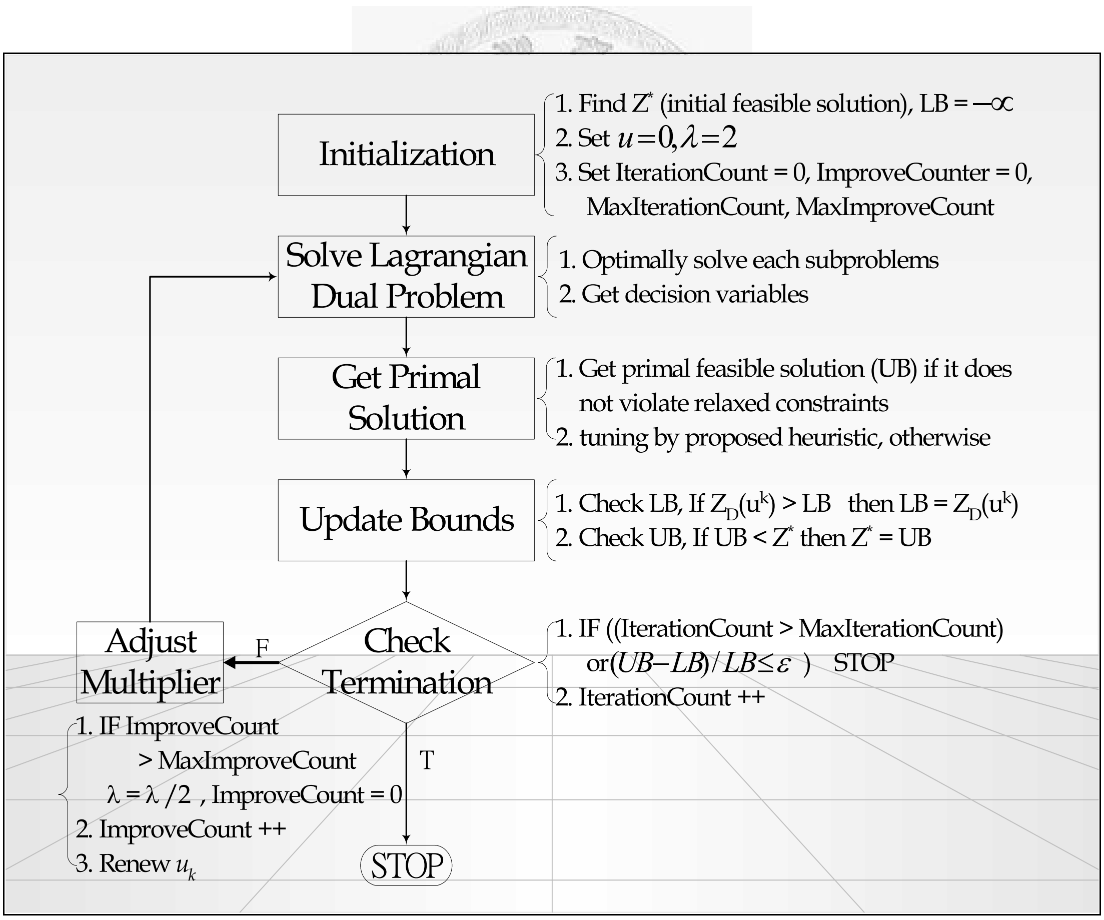


Figure 4-2. Detailed Lagrangean relaxation procedure

4.2 DCR Problem

4.2.1 Solution Approach

The algorithm development is based upon Lagrangean relaxation. In (IP1), by introducing Lagrangean multiplier vector u^1,u^2,u^3, we dualize Constraints (2-1), (2-3) , and (2-6) to obtain the following Lagrangean relaxation problem (LR1).

4.2.2 Lagrangean Relaxation

$$Z_{D1}(u^1,u^2,u^3) = \min \sum_{l\in L} a_l C_l + \sum_{l\in L} u_l^1 (\sum_{g\in G} y_{gl} - C_l) + \sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L} u_{gdl}^2 (\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} - y_{gl})$$

$$+ \sum_{g\in G}\sum_{l\in L} u_{gl}^3 (\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} - |D_g| y_{gl}) \qquad \text{(LR1)}$$

subject to:

$$C_l \in \{0,1,2,3,....,|G|\} \qquad \forall l \in L \qquad (4\text{-}1)$$

$$y_{gl} = 0 \ or \ 1 \qquad \forall g \in G, l \in L \qquad (4\text{-}2)$$

$$\sum_{l\in L} y_{gl} \geq \max\{h_g, |D_g|\} \qquad \forall g \in G \qquad (4\text{-}3)$$

$$\sum_{p\in P_{gd}} x_{gpd} = 1 \qquad \forall g \in G, d \in D_g \qquad (4\text{-}4)$$

$$x_{gpd} = 0 \ or \ 1 \qquad \forall g \in G, d \in D_g, p \in P_{gd} \qquad (4\text{-}5)$$

$$\sum_{l\in n^-} y_{gl} \leq 1 \qquad \forall g \in G, n \in N. \qquad (4\text{-}6)$$

We can decompose (LR1) into three independent subproblems.

Subproblem 4-1 (related to decision variable C_l)

$$\min \sum_{l\in L}(a_l - u_l^1)\, C_l \qquad \text{(SUB4-1)}$$

subject to:

$$C_l \in \{0,1,2,3,....,|G|\} \qquad \forall l \in L. \qquad (4\text{-}1)$$

(SUB4-1) can be further decomposed into |*L*| independent subproblems. For each link *l*,

$$\min \ (a_l - u_l^1)C_l \qquad \text{(SUB4-1-1)}$$

subject to:

$$C_l \in \{0,1,2,3,....,|G|\}.$$

If coefficient of link l $(a_l - u_l^1)$ is negative then set C_l to be |*G*| otherwise 0. The computational complexity of (SUB4-1) is $O(1)$ for each link *l*.

Subproblem 4-2 (related to decision variable y_{gl})

$$\min \sum_{g\in G}\sum_{l\in L}(u_l^1 - u_{gl}^3 \,|\, D_g\,|)\, y_{gl} - \sum_{g\in G}\sum_{l\in L}\sum_{d\in D_g} u_{gdl}^2\, y_{gl} \qquad \text{(SUB4-2)}$$

subject to:

$$y_{gl} = 0 \ \ or \ \ 1 \qquad \forall g \in G, l \in L \qquad (4\text{-}2)$$

$$\sum_{l\in L} y_{gl} \ge \max\{h_g, |D_g|\} \qquad \forall g \in G \qquad (4\text{-}3)$$

$$\sum_{l\in n^-} y_{gl} \le 1 \qquad \forall g \in G, n \in N. \qquad (4\text{-}6)$$

(SUB4-2) can be further decomposed into |*G*| independent subproblems. For each multicast group *g*,

$$\min \sum_{l\in L}\left(u_l^1 - u_{gl}^3 \,|\, D_g\,| - \sum_{d\in D_g} u_{gdl}^2\right) y_{gl} \qquad \text{(SUB4-2-1)}$$

subject to:

$y_{gl} = 0 \ or \ 1 \qquad \forall l \in L$

$\sum_{l \in L} y_{gl} \geq \max\left\{h_g, \left|D_g\right|\right\}$

$\sum_{l \in n^-} y_{gl} \leq 1 \qquad n \in N.$

The proposed algorithm for solving (SUB4-2) is described as follows:

Step 1. Compute the number of negative coefficients for all links where the coefficient for each link l is $u_l^1 - u_{gl}^3 \mid D_g \mid - \sum_{d \in D_g} u_{gdl}^2$.

Step 2. If the number of negative coefficients is greater than $max\{h_g, |D_g|\}$ for each multicast group g, then set each y_{gl} whose corresponding coefficient is negative to 1 otherwise 0.

Step 3. If the number of negative coefficients, say n, is less then $max\{h_g, |D_g|\}$ for multicast group g, then first let each y_{gl} whose corresponding coefficient is negative be 1. Second, assign the ($max\{h_g, |D_g|\} - n$) number of y_{gl} to be 1 whose corresponding coefficients is the smallest positive values. Third, let the remaining y_{gl} be 0.

Step 4. For each sensor node n, check that only one outgoing link y_{gl} can be set to 1. If there are more than one outgoing link set to 1, choose the link with smaller coefficient. After investigating the outgoing link constraint if the total number of y_{gl} set to 1,say k, are smaller than $max\{h_g, |D_g|\}$, assign the ($max\{h_g, |D_g|\} - k$) number of y_{gl} to be 1 whose corresponding coefficients is the smallest values and has not been set to 1 before. Continue step 4 until constraint (4-3) is satisfied.

The computational complexity of above algorithm is $O(|L|(|D_g| + \log|L|))$ for each multicast group g.

Subproblem 4-3 (related to decision variable x_{gdp})

$$\min \sum_{g\in G}\sum_{d\in D_g}\sum_{l\in L}\sum_{p\in P_{gd}}(u^2_{gdl}+u^3_{gl})x_{gpd} \quad \text{(SUB4-3)}$$

subject to:

$$\sum_{p\in P_{gd}} x_{gpd} = 1 \qquad \forall g\in G, d\in D_g \quad \text{(4-4)}$$

$$x_{gpd} = 0 \;\; or \;\; 1 \qquad \forall g\in G, d\in D_g, p\in P_{gd}. \quad \text{(4-5)}$$

(SUB4-3) can be further decomposed into $\sum_{g\in G}|D_g|$ independent shortest path problems with nonnegative arc weight. For each shortest path problem it can be effectively solved by Dijkstra's algorithm. The computational complexity of Dijkstra's algorithm is $O(|N|^2)$ for each destination of the multicast group.

4.2.3 The Dual Problem and the Subgradient Method

According to the algorithms proposed above, we could effectively solve the Lagrangean relaxation problem optimally. Based on the weak Lagrangean duality theorem (for any given set of nonnegative multipliers, the optimal objective function value of the corresponding Lagrangean relaxation problem is a lower bound on the optimal objective function value of the primal problem [1]), $Z_{D1}(u^1,u^2,u^3)$ is a lower bound on Z_{IP1}. We construct the following dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method [1].

Dual Problem (D1):

$$Z_{D1} = \max Z_{D1}\left(u^1,u^2,u^3\right) \quad \text{(D1)}$$

subject to: $u^1,u^2,u^3 \geq 0$.

Let the vector S be a subgradient of $Z_{D1}(u^1,u^2,u^3)$ at (u^1,u^2,u^3). In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (u^{1k},u^{2k},u^{3k})$ is updated by $m^{k+1} = m^k + \alpha^k S^k$, where $S^k(u^1,u^2,u^3) = (\sum_{g\in G} y_{gl} - C_l,$ $\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} - y_{gl},$ $\sum_{d\in D_g}\sum_{p\in P_{gd}} x_{gpd}\delta_{pl} - |D_g|\, y_{gl})$.

The step size α^k is determined by $\delta\dfrac{Z_{IP1}{}^k - Z_{D1}(m^k)}{\|S^k\|^2}$, where $Z_{IP}{}^k$ is the best primal objective function value found by iteration k (an upper bound on the optimal primal objective function value), and δ is a constant ($0 \le \delta \le 2$).

4.3 EDCR Problem

4.3.1 Solution Approach

The algorithm development is based upon Lagrangean relaxation. In (IP2), by introducing Lagrangean multiplier vector v^1,v^2,v^3, we dualize Constraints (2-10), (2-12) ,and (2-14) to obtain the following Lagrangean relaxation problem (LR2).

4.3.2 Lagrangean Relaxation

$$
\begin{aligned}
Z_{D2}(v^1,v^2,v^3) = \min \ & \sum_{n\in N} e_n(r_n) + \sum_{n\in N}\sum_{k\in N}\sum_{s\in S} v^1_{(n,k)s}\Big(\sum_{p\in P_{sq}} x_{sp}\delta_{p(n,k)} - y_{(n,k)}\Big) \\
& + \sum_{n\in N}\sum_{k\in N} v^2_{(n,k)}\Big(\sum_{s\in S}\sum_{p\in P_{sq}} x_{sp}\delta_{p(n,k)} - |S|\cdot y_{(n,k)}\Big) \\
& + \sum_{n\in N}\sum_{k\in N} v^3_{(n,k)}\big(y_{(n,k)} d_{nk} - r_n\big) \qquad \text{(LR2)}
\end{aligned}
$$

subject to:

$$\sum_{n\in N}\sum_{k\in N} y_{(n,k)} \geq \max\{h,\ |S|\} \qquad (4\text{-}7)$$

$$\sum_{k\in N} y_{(n,k)} \leq 1 \qquad \forall n \in N \qquad (4\text{-}8)$$

$$r_n \in R_n \qquad \forall n \in N \qquad (4\text{-}9)$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n,\ k \in N \qquad (4\text{-}10)$$

$$\sum_{p\in P_{sq}} x_{sp} = 1 \qquad \forall s \in S \qquad (4\text{-}11)$$

$$x_{sp} = 0 \text{ or } 1 \qquad \forall s \in S,\ p \in P_{sq}. \qquad (4\text{-}12)$$

We can decompose (LR2) into three independent subproblems.

Subproblem 4-4 (related to decision variable r_n)

$$\min \sum_{n\in N}(e_n(r_n) - r_n \sum_{k\in N} v^3_{(n,k)}) \qquad \text{(SUB4-4)}$$

subject to:

$$r_n \in R_n \qquad \forall n \in N. \qquad (4\text{-}9)$$

(SUB4-4) can be further decomposed into $|N|$ independent subproblems. For each node *n*,

$$\min\ e_n(r_n) - r_n \sum_{k\in N} v^3_{(n,k)} \qquad \text{(SUB4-4-1)}$$

subject to:

$r_n \in R_n$.

Since R_n is a finite and discrete set, we can examine all possible transmission radii of node *n* to identify the smallest value of the objective function. The computational complexity of (SUB4-4) is $O(|R_n|)$ for each node *n*.

Subproblem 4-5 (related to decision variable $y_{(n,k)}$)

$$\min \sum_{n\in N}\sum_{k\in N}(v^3_{(n,k)}d_{nk} - u^2_{(n,k)}|S| - \sum_{s\in S} v^1_{(n,k)s})y_{(n,k)} \qquad \text{(SUB4-5)}$$

subject to:

$$\sum_{n\in N}\sum_{k\in N} y_{(n,k)} \ge \max\{h, |S|\} \qquad (4\text{-}7)$$

$$\sum_{k\in N} y_{(n,k)} \le 1 \qquad \forall n \in N \qquad (4\text{-}8)$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n, k \in N. \qquad (4\text{-}10)$$

The proposed algorithm for solving (SUB4-5) is described as follows:

Step1. For each link (n,k) compute the coefficient $v^3_{(n,k)}d_{nk} - v^2_{(n,k)}|S| - \sum_{s\in S} v^1_{(n,k)s}$ for each $y_{(n,k)}$.

Step2. For all outgoing links of node n, find the smallest coefficient. If the smallest coefficient is negative then set the corresponding $y_{(n,k)}$ to be 1 and the other outgoing links $y_{(n,k)}$ to be 0, otherwise set all outgoing link $y_{(n,k)}$ to be 0. Repeat step 2 for all nodes.

Step3. If the total number of $y_{(n,k)}$ whose value is 1 (denote as T) are smaller than $max\{h, |S|\}$, then identify the nodes that have all its outgoing links $y_{(n,k)} = 0$. From these identified nodes, selected ($max\{h, |S|\}-T$) number of these identified nodes whose corresponding smallest coefficients are the smallest. Then, assign the outgoing link $y_{(n,k)} = 1$ with the smallest coefficient for each of these selected nodes.

The computational complexity of above algorithm is $O(|N|^2)$.

Subproblem 4-6 (related to decision variable x_{sp})

$$\min \sum_{n\in N}\sum_{k\in N}\sum_{s\in S}\sum_{p\in P_{sq}}(v^1_{(n,k)s}+v^2_{(n,k)})x_{sp}\delta_{p(n,k)} \qquad \text{(SUB4-6)}$$

subject to:

$$\sum_{p\in P_{sq}} x_{sp} = 1 \qquad \forall s\in S \qquad \text{(4-11)}$$

$$x_{sp} = 0 \text{ or } 1 \qquad \forall s\in S,\ p\in P_{sq}. \qquad \text{(4-12)}$$

(SUB4-6) can be further decomposed into |S| independent shortest path problems with nonnegative arc weight whose value is $v^1_{(n,k)s}+v^2_{(n,k)}$. For each shortest path problem it can be effectively solved by Dijkstra's algorithm. The computational complexity of Dijkstra's algorithm is $O(|N|^2)$ for each source node.

4.3.3 The Dual Problem and the Subgradient Method

According to the algorithms proposed above, we could effectively solve the Lagrangean relaxation problem optimally. Based on the weak Lagrangean duality theorem (for any given set of nonnegative multipliers, the optimal objective function value of the corresponding Lagrangean relaxation problem is a lower bound on the optimal objective function value of the primal problem [1]), $Z_{D2}(v^1,v^2,v^3)$ is a lower bound on Z_{IP2}. We construct the following dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method [1].

Dual Problem (D2):

$$Z_{D2} = \max Z_{D2}\left(v^1,v^2,v^3\right) \qquad \text{(D2)}$$

subject to: $v^1, v^2, v^3 \ge 0$.

Let the vector S be a subgradient of $Z_{D2}(v^1,v^2,v^3)$ at (v^1,v^2,v^3). In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (v^{1k},v^{2k},v^{3k})$ is updated by $m^{k+1} = m^k + \alpha^k S^k$, where $S^k(v^1,v^2,v^3) = (\sum_{p \in P_{sq}} x_{sp}\delta_{p(n,k)} - y_{(n,k)}$, $\sum_{s \in S}\sum_{p \in P_{sq}} x_{sp}\delta_{p(n,k)} - |S| \cdot y_{(n,k)}$, $y_{(n,k)}d_{nk} - r_n)$. The step size α^k is determined by $\delta \frac{Z_{IP2}{}^k - Z_{D2}(m^k)}{\| S^k \|^2}$, where $Z_{IP}{}^k$ is the best primal objective function value found by iteration k (an upper bound on the optimal primal objective function value), and δ is a constant ($0 \le \delta \le 2$)

4.4 Data Aggregation Tree with QoS Routing

4.4.1 Solution Approach

Before applying Lagrangean relaxation to this model, we replace constraint (3-11) and (3-12) by (3-21) and (3-22), respectively. The development of algorithm is based upon Lagrangean relaxation. In (IP3), by introducing Lagrangean multiplier vector u^1, u^2, u^3, u^4, u^5, u^6, u^7, and u^8, we dualize Constraints (3-2), (3-4), (3-7), (3-8), (3-9), (3-10), (3-11), and (3-12) to obtain the following Lagrangean relaxation problem (LR3).

4.4.2 Lagrangean Relaxation

$$Z_{D3}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8) = \min \sum_{n\in N}\left(\mathrm{t}_{\mathrm{data}} + (RTS\cdot\sum_{k\in N} c_{nk})\right)\cdot e_n(r_n) + \sum_{n\in N} m_n \cdot E_{idle} \; +$$

$$\sum_{n\in N}\sum_{k\in N}\sum_{s\in S} u^1_{nks}\left(\sum_{p\in P_{sq}} x_{sp}\delta_{p(n,k)} - y_{(n,k)}\right) + \sum_{n\in N}\sum_{k\in N} u^2_{nk}\left(\sum_{s\in S}\sum_{p\in P_{sq}} x_{sp}\delta_{p(n,k)} - |S|\, y_{(n,k)}\right) +$$

$$\sum_{n\in N}\sum_{k\in N} u^3_{nk}(r_n - d_{nk} - Mz_{nk}) + \sum_{n\in N}\sum_{k\in N} u^4_{nk}(z_{nk}d_{nk} - r_n) + \sum_{n\in N}\sum_{k\in N} u^5_{nk}(y_{(n,k)} - z_{nk}) +$$

$$\sum_{n\in N}\sum_{k\in N} u^6_{kn}\left(m_k + l_{(k,n)} - A(1-y_{(k,n)}) - m_n\right) +$$

$$\sum_{n\in N}\sum_{k\in N} u^7_{nk}\left(\begin{array}{l}\ln\left(RTS + SIFS + CTS + 330\right) + 0.115 + 0.017\sum_{j\in N} z_{jn} \\ + \lambda(RTS + SIFS + 2\theta)\sum_{j\in N} z_{jk} - \ln(l_{(n,k)})\end{array}\right) +$$

$$\sum_{n\in N}\sum_{k\in N} u^8_{nk}\left(\lambda(RTS + SIFS + 2\theta)\sum_{j\in N} z_{jk} - M(1-y_{(n,k)}) - \ln(c_{nk})\right) \qquad \text{(LR3)}$$

subject to:

$$\sum_{n\in N}\sum_{k\in N} y_{(n,k)} \ge \max\{h, |\mathrm{S}|\} \qquad \text{(4-13)}$$

$$\sum_{p \in P_{sq}} x_{sp} = 1 \qquad \forall s \in S \qquad (4\text{-}14)$$

$$\sum_{k \in N} y_{(n,k)} \leq 1 \qquad \forall n \in N \qquad (4\text{-}15)$$

$$x_{sp} = 0 \text{ or } 1 \qquad \forall s \in S,\ p \in P_{sq} \qquad (4\text{-}16)$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n, k \in N \qquad (4\text{-}17)$$

$$z_{nk} = 0 \text{ or } 1 \qquad \forall n, k \in N \qquad (4\text{-}18)$$

$$r_n \in R_n \qquad \forall n \in N \qquad (4\text{-}19)$$

$$r_n \neq 0 \qquad \forall n \in S \qquad (4\text{-}20)$$

$$c_{nk} \in \{0, 1, 2, ..., T\} \qquad \forall n, k \in N \qquad (4\text{-}21)$$

$$B \geq m_n \geq 0 \qquad \forall n \in N \qquad (4\text{-}22)$$

$$A \geq l_{(n,k)} \geq e^{0.115} \cdot (RTS + SIFS + CTS + 330) \qquad \forall n, k \in N. \qquad (4\text{-}23)$$

We can decompose (LR3) into six independent subproblems.

Subproblem 4-7 (related to decision variable m_n)

$$\min \sum_{n\in N} m_n \cdot E_{idle} + \sum_{n\in N}\sum_{k\in N} u_{kn}^6 (m_k - m_n) \qquad \text{(SUB4-7)}$$

subject to:

$$B \ge m_n \ge 0 \qquad \forall n \in N. \qquad \text{(4-22)}$$

We rewrite the objective function of (SUB4-7) into another form so that this subproblem can be efficiently solved.

Transformation:

$$\begin{aligned}
&\sum_{n\in N} m_n \cdot E_{idle} + \sum_{n\in N}\sum_{k\in N} u_{kn}^6 (m_k - m_n) \\
&= \sum_{n\in N} m_n \cdot E_{idle} + \sum_{n\in N}\sum_{k\in N} u_{kn}^6 m_k - \sum_{n\in N}\sum_{k\in N} u_{kn}^6 m_n \\
&= \sum_{n\in N} m_n \cdot E_{idle} + \sum_{n\in N}\left(\sum_{k\in N} u_{nk}^6 - \sum_{k\in N} u_{kn}^6\right) m_n \\
&= \sum_{n\in N}\left(E_{idle} + \sum_{k\in N} u_{nk}^6 - \sum_{k\in N} u_{kn}^6\right) m_n .
\end{aligned}$$

After transforming the objective function, we can now decompose (SUB-4-7) into $|N|$ independent subproblems. For each node n,

$$\min \left(E_{idle} + \sum_{k\in N} u_{nk}^6 - \sum_{k\in N} u_{kn}^6\right) m_n \qquad \text{(SUB4-7-1)}$$

subject:

$$B \ge m_n \ge 0$$

For each (SUB4-7-1) subproblem, we check the coefficient $E_{idle} + \sum_{k\in N} u_{nk}^6 - \sum_{k\in N} u_{kn}^6$ of each node n. If the coefficient of node n is negative then set m_n to be B, otherwise 0. The computational complexity of (SUB4-7) is $O(1)$ for each node n.

Subproblem 4-8 (related to decision variable $y_{(n,k)}$)

$$\min \sum_{n \in N} \sum_{k \in N} (u_{nk}^5 + u_{nk}^6 A + u_{nk}^8 M - u_{nk}^2 \mid S \mid - \sum_{s \in S} u_{nks}^1) y_{(n,k)} \qquad \text{(SUB4-8)}$$

subject to:

$$\sum_{n \in N} \sum_{k \in N} y_{(n,k)} \ge \max\{h, |\mathrm{S}|\} \qquad \text{(4-13)}$$

$$\sum_{k \in N} y_{(n,k)} \le 1 \qquad \forall n \in N \qquad \text{(4-15)}$$

$$y_{(n,k)} = 0 \text{ or } 1 \qquad \forall n, k \in N. \qquad \text{(4-17)}$$

The proposed algorithm for solving (SUB4-8) is described as follows.

Step1. For each link (n,k) compute the coefficient $u_{nk}^5 + u_{nk}^6 U + u_{nk}^8 M - u_{nk}^2 \mid S \mid - \sum_{s \in S} u_{nk}^1$ for each $y_{(n,k)}$.

Step2. For all outgoing links of node n, find the smallest coefficient. If the smallest coefficient is negative then set the corresponding $y_{(n,k)}$ to be 1 and the other outgoing links $y_{(n,k)}$ to be 0, otherwise set all outgoing link $y_{(n,k)}$ to be 0. Repeat step 2 for all nodes.

Step3. If the total number of $y_{(n,k)}$ whose value is 1 (denoted as T) are smaller than $max\{h, |\mathrm{S}|\}$, then identify the nodes that have all its outgoing links $y_{(n,k)} = 0$. From these identified nodes, selected ($max\{h, |\mathrm{S}|\} - T$) number of these identified nodes whose corresponding smallest coefficients are the smallest. Then, assign the outgoing link $y_{(n,k)} = 1$ with the smallest coefficient for each of these selected nodes.

The computational complexity of above algorithm is $O(|N|^2)$.

Subproblem 4-9 (related to decision variable x_{sp})

$$\min \sum_{n\in N}\sum_{k\in N}\sum_{s\in S}\sum_{p\in P_{sq}}(u^1_{nks}+u^2_{nk})x_{sp}\delta_{p(n,k)} \tag{SUB4-9}$$

subject to:

$$\sum_{p\in P_{sq}} x_{sp}=1 \qquad \forall s\in S \tag{4-14}$$

$$x_{sp}=0 \text{ or } 1 \qquad \forall s\in S,\ p\in P_{sq}. \tag{4-16}$$

(SUB4-9) can be further decomposed into |*S*| independent shortest path problems with nonnegative arc weight whose value is $u^1_{nks}+u^2_{nk}$. For each shortest path problem it can be effectively solved by Dijkstra's algorithm. The computational complexity of Dijkstra's algorithm is $O(|N|^2)$ for each source node.

Subproblem 4-10 (related to decision variable r_n and c_{nk})

$$\min \sum_{n\in N} e_n(r_n)\cdot t_{data} + RTS\sum_{n\in N}\sum_{k\in N} e_n(r_n)\cdot c_{nk} + \sum_{n\in N}\sum_{k\in N}(u^3_{nk}-u^4_{nk})r_n - \sum_{n\in N}\sum_{k\in N} u^8_{nk}\ln(c_{nk}) \tag{SUB4-10}$$

subject to:

$$r_n\in R_n \qquad \forall n\in N \tag{4-19}$$

$$r_n\neq 0 \qquad \forall n\in S \tag{4-20}$$

$$c_{nk}\in\{0,1,2,....,T\} \qquad \forall n,k\in N. \tag{4-21}$$

(SUB4-10) can be further decomposed into $|N|$ independent subproblems. For each node *n*,

$$\min\ e_n(r_n)\cdot t_{data}+RTS\cdot e_n(r_n)\sum_{k\in N}c_{nk}+r_n\sum_{k\in N}(u^3_{nk}-u^4_{nk})-\sum_{k\in N}u^8_{nk}\ln(c_{nk}) \tag{SUB4-10-1}$$

subject to:

$r_n \in R_n$

$r_n \neq 0 \qquad\qquad n \in S$

$c_{nk} \in \{0, 1, 2, ..., T\} \qquad\qquad \forall k \in N$.

Each (SUB4-10-1) subproblem can be optimally solved by exhaustively searching the combination of radius r_n and c_{nk}. The computational complexity of (SUB4-10) is $O(|R_n| \times |T|)$ for each node n.

Subproblem 4-11 (related to decision variable z_{nk})

$$\min \begin{array}{l} \sum_{n\in N}\sum_{k\in N}\left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5\right) z_{nk} + \lambda(RTS + CTS + 2\theta) \\ \sum_{n\in N}\sum_{k\in N}\sum_{j\in N}(u_{nk}^7 + u_{nk}^8) z_{jk} + 0.017\sum_{n\in N}\sum_{k\in N}\sum_{j\in N} u_{nk}^7 z_{jn} \end{array} \qquad \text{(SUB4-11)}$$

subject to:

$$z_{nk} = 0 \text{ or } 1 \qquad \forall n, k \in N . \qquad \text{(4-18)}$$

We can transform the objective function of (SUB4-11) into the following form in order to effectively solve this subproblem:

$$\sum_{n\in N}\sum_{k\in N}\left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5\right) z_{nk} + \lambda(RTS + CTS + 2\theta)\sum_{n\in N}\sum_{k\in N}\sum_{j\in N}(u_{nk}^7 + u_{nk}^8) z_{jk}$$
$$+ 0.017\sum_{n\in N}\sum_{k\in N}\sum_{j\in N} u_{nk}^7 z_{jn}$$
$$= \sum_{n\in N}\sum_{k\in N}\left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5\right) z_{nk} + \lambda(RTS + CTS + 2\theta)\sum_{n\in N}\sum_{k\in N}\left(\sum_{j\in N} u_{jk}^7 + u_{jk}^8\right) z_{nk}$$
$$+ 0.017\sum_{n\in N}\sum_{k\in N}\left(\sum_{j\in N} u_{kj}^7\right) z_{nk}$$
$$= \sum_{n\in N}\sum_{k\in N}\left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5 + \lambda(RTS + CTS + 2\theta)\sum_{j\in N}\left(u_{jk}^7 + u_{jk}^8\right) + 0.017\sum_{j\in N} u_{kj}^7\right) z_{nk}$$

Thereafter, (SUB4-11) can be decomposed into $|N \times N|$ independent subproblems. For each link (n,k),

$$\min \left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5 - \lambda(RTS + CTS + 2\theta) \sum_{j \in N} \left(u_{jk}^7 + u_{jk}^8 \right) + 0.017 \sum_{j \in N} u_{kj}^7 \right) z_{nk} \quad \text{(SUB4-11-1)}$$

subject to:

$z_{nk} = 0$ or 1

(SUB4-11-1) is an easy problem to be solved. If the corresponding coefficient $\left(u_{nk}^4 d_{nk} - u_{nk}^3 M - u_{nk}^5 - 2\lambda(RTS + CTS + 2\theta) \sum_{j \in N} \left(u_{jk}^7 + u_{jk}^8 \right) + 0.017 \sum_{j \in N} u_{kj}^7 \right) z_{nk}$ of link (n,k) is negative then set z_{nk} to be 1, otherwise 0. The computational complexity of (SUB4-11) is $O(1)$ for each link (n,k).

Subproblem 4-12 (related to decision variable $l_{(n,k)}$)

$$\min \sum_{n \in N} \sum_{k \in N} u_{nk}^6 l_{(n,k)} - u_{nk}^7 \ln(l_{(n,k)}) \quad \text{(SUB4-12)}$$

subject to:

$$A \geq l_{(n,k)} \geq e^{0.115} \cdot (RTS + SIFS + CTS + 330) \qquad \forall n, k \in N. \quad \text{(4-23)}$$

We can further decompose (SUB4-12) into $|N \times N|$ independent subproblems. For each link (n,k),

$$\min \quad u_{nk}^6 l_{(n,k)} - u_{nk}^7 \ln(l_{(n,k)}) \quad \text{(SUB4-12-1)}$$

subject to:

$A \geq l_{(n,k)} \geq e^{0.115} \cdot (RTS + SIFS + CTS + 330)$.

For each (SUB4-12-1) subproblem, if u_{nk}^7 is negative then set $l_{(n,k)}$ to be $e^{0.115} \cdot (RTS + SIFS + CTS + 330)$. If u_{nk}^7 is positive then we can get the value of $l_{(n,k)}$ that makes (SUB4-12-1) minimal by following procedure.

Apply first derivative of $l_{(n,k)}$ on the objective function of (SUB4-12-1) and let it equal 0, the optimal value of $l_{(n,k)}$ is:

$$\frac{\partial(u_{nk}^{6} l_{(n,k)} - u_{nk}^{7} \ln(l_{(n,k)}))}{\partial l_{(n,k)}} = u_{nk}^{6} - \frac{u_{nk}^{7}}{l_{(n,k)}}$$
$$\Rightarrow u_{nk}^{6} - \frac{u_{nk}^{7}}{l_{(n,k)}} = 0$$
$$\Rightarrow l_{(n,k)} = \frac{u_{nk}^{7}}{u_{nk}^{6}}.$$

The second derivative of (SUB4-12-1) is larger than zero:

$$\frac{\partial(u_{nk}^{6} - \frac{u_{nk}^{7}}{l_{(n,k)}})}{\partial l_{(n,k)}} = \frac{u_{nk}^{7}}{(l_{(n,k)})^{2}} \geq 0, \text{since } u_{nk}^{7} \text{ is positive.}$$

The computational complexity of (SUB4-12) is $O(1)$ for each link (n,k).

4.4.3 The Dual Problem and the Subgradient Method

According to the algorithms proposed above, we could effectively solve the Lagrangean relaxation problem optimally. Based on the weak Lagrangean duality theorem (for any given set of nonnegative multipliers, the optimal objective function value of the corresponding Lagrangean relaxation problem is a lower bound on the optimal objective function value of the primal problem [1]), $Z_{D3}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8)$ is a lower bound on Z_{D3}. We construct the following dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method [1].

Dual Problem (D3):

$$Z_{D3} = \max Z_{D3}\left(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8\right) \tag{D3}$$

subject to: $u^1,u^2,u^3,u^4,u^5,u^6,u^8 \geq 0.\ \ u^7 \geq -\infty$.

Let the vector S be a subgradient of $Z_{D3}(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8)$ at $(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8)$. In iteration k of the subgradient optimization procedure, the multiplier vector $m^k = (u^{1k},u^{2k},u^{3k},u^{4k},u^{5k},u^{6k},u^{7k},u^{8k})$ is updated by $m^{k+1} = m^k + \alpha^k S^k$, where

$$S^k(u^1,u^2,u^3,u^4,u^5,u^6,u^7,u^8)=\Bigg(\sum_{p\in P_{sq}} x_{sp}\delta_{p(n,k)}-y_{(n,k)},\ \sum_{s\in S}\sum_{p\in P_{sq}}(x_{sp}\delta_{p(n,k)}-|S|\,y_{(n,k)}),$$

$$r_n-d_{nk}-Mz_{nk},\ z_{nk}d_{nk}-r_n,\ y_{(n,k)}-z_{nk},\ m_k+l_{(k,n)}-A(1-y_{(k,n)})-m_n,$$

$$\ln\big(RTS+SIFS+CTS+330\big)+0.115+0.017\sum_{j\in N} z_{jn}$$

$$+\lambda(RTS+SIFS+2\theta)\sum_{j\in N} z_{jk}-\ln(l_{(n,k)}),$$

$$\lambda(RTS+SIFS+2\theta)\sum_{j\in n} z_{jk}-M(1-y_{(n,k)})-\ln(c_{nk})\Bigg).$$

The step size α^k is determined by $\delta\dfrac{Z_{IP3}{}^k-Z_{D3}(m^k)}{\|S^k\|^2}$, where $Z_{IP}{}^k$ is the best primal objective function value found by iteration k (an upper bound on the optimal primal objective function value), and δ is a constant ($0\le\delta\le 2$).

Chapter 5 Getting Primal Feasible Solutions

By using Lagrangean relaxation and the subgradient method, we can get a theoretical lower bound of the primal problem. In addition, the results obtained from the procedure of Lagrangean relaxation will provide some good hints to help us get the primal feasible solutions. The solutions to the Lagrangean relaxation problem and Lagrangean multipliers are both good hints. If the solutions of Lagrangean relaxation satisfy all constraints in the primal problem, a primal feasible solution is found. Otherwise, we need to make some modifications to transform the infeasible solution into a feasible one. Lagrangean multipliers can also be used on some existing heuristics to adjust the original heuristic to a Lagrangean-based modified heuristic.

5.1 Getting Primal Feasible Solutions of DCR

To obtain primal feasible solutions for data-centric wireless sensor routing problems, solutions to the Lagrangean Relaxation (LR1) are considered. We propose the following two heuristics for getting primal feasible solutions.

The first heuristic constructs a shortest path tree based on the solutions in (SUB4-3). However, in (SUB4-3), the union of the shortest path for each data source node may not be a tree, since the arc weight of link l is $u^2_{gdl} + u^3_{gl}$. The multiplier u^2_{gdl} is associated with each data source node d. In other words, each data source node may have a different arc weight on link l, which results in the possibility of having a cycle for the union of the shortest paths. Therefore, we set the arc weight of link l to be

$\frac{\sum_{d \in D_g} u_{gdl}^2}{|D_g|} + u_{gl}^3 + a_l$, so that the arc weight for link l is the same as each data source node d of multicast group g. This ensures that the union of the shortest paths destined to every data source in a multicast group shall be a tree. In order to take account of the transmission cost, we also introduce a_l on the arc weight. The computational complexity for first heuristic is $O(|G||N|^2)$.

The basic idea of the second heuristic is GIT, which according to [12], is a better heuristic than shortest path tree heuristics. By leveraging the solutions to the dual problem (LR1), we set the arc weight for link l as $a_l + u_{gl}^3$. The first term, a_l, reflects the transmission cost and the second term, u_{gl}^3, reflects the penalty cost for link l to be a link in a cycle. By incorporating $a_l + u_{gl}^3$ as the arc weight, we try to achieve the minimum transmission cost and gain from the data-centric routing (tree) at the same time.

In addition to the link arc weight setting, we have developed an efficient method to implement the GIT algorithm. In the traditional GIT algorithm for tree construction, if there are three nodes in a tree and two unvisited data source nodes, we have to perform Dijkstra's algorithm six times to determine the minimum distance to one unvisited data source node. By adding two pseudo nodes, we only need to perform Dijkstra's algorithm once to identify the minimum route to the closest unvisited data source node. The first pseudo node is used for all nodes in the current multicast tree and the other is used for all source nodes not contained in the current multicast tree. For each node, n, in the current multicast tree we add a pseudo link whose arc weight is 0 from pseudo node 1 to n. For each unvisited source node, s, we add a pseudo link whose arc weight is 0 from s to pseudo node 2. After the new topology is constructed, we perform Dijkstra's algorithm to find the shortest path between pseudo nodes 1 and 2. Along this shortest path, the closest unvisited data source node will be visited. The example in Figure 5-1 illustrates the concept. The computational complexity of the second heuristic is $O(|N|^2 \sum_{g \in G} |D_g|)$.

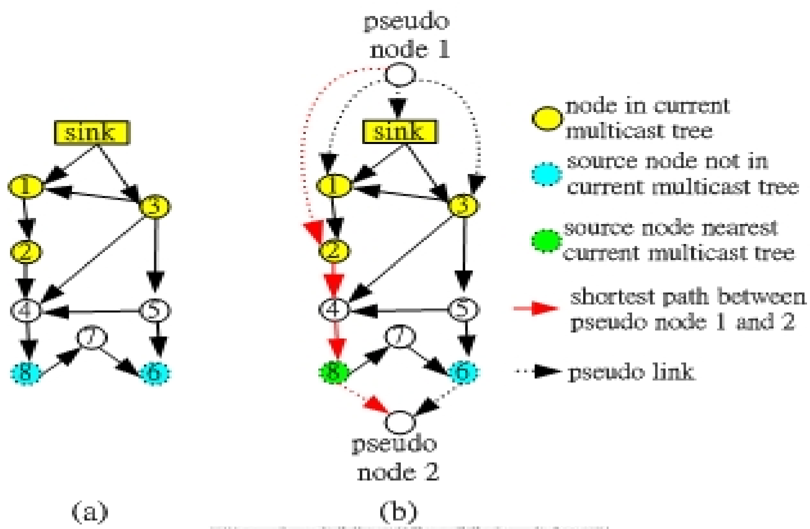


Figure 5-1. (a) Initial topology with a current multicast tree and two unvisited sources. (b) New topology with the shortest path between pseudo nodes 1 and 2.

In the following, we show the complete algorithm (denoted as LGR) for solving (IP1).

Algorithm LGR

begin

Initialize the Lagrangean multiplier vector (u^1, u^2, u^3) to be all zero vectors;

run **Calculate_h_g**;

UB := very large number; *LB* := 0;

improve_counter := 0; *step_size_coefficient* := 2;

for *iteration* := 1 **to** *Max_Iteration_Number* **do**

begin

run subproblem (SUB4-1);

run subproblem (SUB4-2);

run subproblem (SUB4-3);

calculate Z_{D1};

if $Z_{D1} > LB$ **then**

$LB := Z_{D1}$; *improve_counter* := 0;

else *improve_counter* := *improve_counter* + 1;

if *improve_counter* = *Improve_Threshold* **then**

improve_counter := 0; $\delta := \delta / 2$;

run Primal_Heuristic_Algorithm of DCR;

if $ub < UB$ **then** $UB := ub$; /* *ub* is the newly computed upper bound. */

run update-step-size;

run update-Lagrangean-multiplier;

end;

end;

The computational complexity of LGR is $O(|N|^2 \times \sum_{g \in G} |D_g| + |L||G| \log |L|)$ for each iteration.

5.2 Getting Primal Feasible Solutions of EDCR

To obtain the primal feasible solutions to the extension of data-centric wireless sensor routing problem, solutions to the Lagrangean Relaxation (LR2) are considered. We propose the following two heuristics to get the primal feasible solutions.

The first heuristic constructs the shortest path tree based on the solutions in (SUB4-6). However, in (SUB4-6), the union of the shortest path for each data source node may not be a tree, since the multiplier, $v^1_{(n,k)s}$, is associated with each data source node *s*. In other words, each data source node may have a different arc weight on link (*n,k*), which results in the possibility of having a cycle for the union of the shortest paths. Therefore, we set the arc weight of link (*n*,*k*) to be $\frac{\sum_{s\in S} v^1_{(n,k)s}}{|S|} + v^2_{(n,k)} + (\frac{d_{nk}}{4})^2$, so that the arc weight for link (*n*,*k*) is the same for all data source nodes. This ensures that the union of the shortest paths destined to every data source shall be a tree. In order to take account of the transmission energy consumption, we also introduce transmission distance d_{nk} between nodes *n* and *k* on the arc weight. After the aggregation tree is determined, the minimum power to cover each link on the tree can be determined. The computational complexity for first heuristic is $O(|N|^2)$.

The principle idea of the second heuristic is also based on leveraging GIT. We set the arc weight for link (*n*,*k*) as $v^2_{(n,k)} + \left(\frac{d_{nk}}{4}\right)^2$ and then run the GIT algorithm. The idea of dividing d_{nk} by 4 is for normalization purposes such that the arc weight will not be dominated by d_{nk}. The first term, $v_{(n,k)}^2$, reflects the penalty cost for link (n,k) to be a link in a cycle. The second term, $\left(\frac{d_{nk}}{4}\right)^2$, is used to reflect the transmission power consumption. By incorporating $v_{(n,k)}^2 + \left(\frac{d_{nk}}{4}\right)^2$ as the arc weight, we try to achieve minimum transmission energy and gain from the data-centric routing (tree) at the same time. After the aggregation tree is determined, the minimum power to cover each link on the tree can be determined. The computational complexity of second heuristic is $O(|N|^2|S|)$.

In the following, we show the complete algorithm (denoted as LGR2) for solving (IP2).

Algorithm LGR2

begin

Initialize the Lagrangean multiplier vector (v^1,v^2,v^3) to be all zero vectors;

run **Calculate_*h*$_g$** to determine *h*.

UB := very large number; *LB* := 0;

improve_counter := 0;

step_size_coefficient := 2;

for *iteration* := 1 **to** *Max_Iteration_Number* **do**

begin

run subproblem (SUB4-4);

run subproblem (SUB4-5);

run subproblem (SUB4-6);

```
calculate Z_D2;
if Z_D2 > LB then
    LB := Z_D2;    improve_counter := 0;
else improve_counter := improve_counter + 1;
if improve_counter = Improve_Threshold then
    improve_counter := 0;    δ := δ / 2;
run Primal_Heuristic_Algorithm of EDCR;
if ub < UB then UB := ub;    /* ub is the newly computed upper bound. */
run update-step-size;
run update-Lagrangean-multiplier;
end;
end;
```

The computational complexity for LGR1 is $O(|N|^2|S| + \sum_{n \in N}|R_n|)$ for each iteration.

5.3 Getting Primal Feasible Solutions for Data Aggregation Tree with QoS Routing

To obtain the primal feasible solutions for a data aggregation tree with the QoS routing problem, we consider solutions to the Lagrangean relaxation (LR3) problem. Once the routing path, x_{sp}, for each source, *s,* is determined, all other decision variables, e.g., r_n and y_{nk}, can be calculated and the total energy consumption of the data aggregation tree can be obtained. The solution of (SUB 4-9) is probably the most promising feasible solution to the primal problem, yet it may violate the tree constraint. Thus, we propose a drop heuristic to eliminate those links that form the cycle on the tree.

The steps of the drop heuristic are as follows:

1. Based on the solutions of (SUB 4-9) we can get the set of decision variables, x_{sp}, from which we can decide which link, y_{nk}, is used on the routing path by source *s*. After determining y_{nk}, if y_{nk} is 1, we set the arc weight of it corresponding link to be $\frac{\sum_{s\in S} u^1_{nks}}{|S|} + u^2_{nk}$; otherwise, we set the arc weight to be infinity.
2. According to the arc weight calculated in Step 1, we sort the links from small to large.
3. We sequentially examine all links from the link with the largest arc weight to the smallest, but we ignore the links with infinity costs. We remove each link say link (n, k) from the routing path and check whether every source node still has a routing path to the sink node. If any source node is unable to reach the sink node after removing link (n, k), we restore link (n, k) onto the routing path. If every

source still has a routing path to reach the sink node, we remove (n,k) and investigate the next link until all the links used by the union of routing path x_{sp} have been examined.

After executing the drop heuristic we get a data aggregation tree without any cycles. The computational complexity of the drop heuristic is $O(|S||N|^3)$.

In order to decrease the maximum end-to-end delay of the data aggregation tree obtained by the drop heuristic, we have developed a rerouting heuristic to improve the solution quality of getting primal feasible solutions. The steps of the rerouting heuristic are as follows:

1. Identify the path (denoted as P) that incurs the maximum end-to-end delay.
2. Investigate nodes located on P one by one. For each checked node (denoted as n), examine each node (denoted as k) within the transmission radius of n. If the maximum end-to-end delay of node n plus the link delay, $l_{(n,k)}$, is smaller than the maximum end-to-end delay of node k, then reroute the outgoing link on the routing path of node n from the original routing link to the outgoing link (n,k). If no node k can be rerouted by n, then check the next node on P until the sink node is reached.
3. Update the decision variable $y_{(n,k)}$ and recalculate the maximum end-to-end delay of the new data aggregation tree.
4. If no node on path P can be rerouted, then stop the heuristic; otherwise, go to Step 1.

The pseudo code of the rerouting heuristic is as follows:

Rerouting Heuristic

```
begin
    Update_Success := true;
    while Update_Success = true do
    begin
        Update_Success := false;
        Identify the path P that incurs the maximum end-to-end delay.
        for each node n on path P do
        begin
            for each node k within transmission radius of node n do
            begin
                if M_n + L_(n,k) < M_k then
                    run update_ y_(n,k);
                    Update_Success := true;
                    break;
            end;
            if Update_Sucess = true then
                break;
        end;
        run recalculate_Maximum_End-to-End_Delay;
    end;
end;
```

The computational complexity of the rerouting heuristic is $O(|N|^4)$.

In the following, we show the complete algorithm (denoted as LGR3) for solving (IP3).

Algorithm LGR3

begin

Initialize the Lagrangean multiplier vector ($u^1, u^2, u^3, u^4, u^5, u^6, u^7, u^8$) to be all zero vectors;

run **Calculate_h_g** to determine h.

UB := very large number; *LB* := 0;

improve_counter := 0;

step_size_coefficient := 2;

for *iteration* := 1 **to** *Max_Iteration_Number* **do**

begin

run subproblem (SUB4-7);

run subproblem (SUB4-8);

run subproblem (SUB4-9);

run subproblem (SUB4-10);

run subproblem (SUB4-11);

run subproblem (SUB4-12);

calculate Z_{D3};

if $Z_{D3} > LB$ **then**

$LB := Z_{D3}$; *improve_counter* := 0;

else *improve_counter* := *improve_counter* + 1;

if *improve_counter* = *Improve_Threshold* **then**

improve_counter := 0; $\delta := \delta / 2$;

run Primal_Heuristic_Algorithm of EDCR with QoS;

if *ub* < *UB* **then** *UB* := *ub*; /* *ub* is the newly computed upper bound. */

run update-step-size;

run update-Lagrangean-multiplier;

end;

end;

The computational complexity for LGR3 is $O(|N|^4)$ for each iteration.

Chapter 6 Computational Experiments

6.1 Computational Experiments of DCR

6.1.1 Experiment Environments

The proposed algorithms for the DCR problem in wireless sensor networks are coded in C and run on a PC with INTEL^{TM} PIII-1.3G. *Max_Iteration_Number* and *Improve_Threshold* are set to 2000 and 50 respectively. The step size coefficient, δ, is initialized as 2 and is halved if the objective function value of the dual problem is not improved for iterations up to *Improve_Threshold*.

Two source placement models, namely *event-driven* and *random-source* models, are tested. The *event-drive* model, described in Chapter 1, requires that all sensor nodes within sensing range of a specific event become source nodes. In a *random-source* model, non-sink nodes are randomly selected to be data source nodes. The random-source model differs from *event-driven* model in that the source nodes are not necessarily clustered. Query-based applications and periodic applications could be classified as the *random-source* model.

We construct a network topology consisting of N = 300 sensor nodes randomly placed in a 1×1 square area. The power aware transmission cost, (a_l), is defined as $100\times$Euclidean distance if the link length does not exceed the transmission radius. In Figure 6-1 and Figure 6-3, the communication radius is configured as 0.125. In other words, the link with a length greater than 0.125 will have an extremely high transmission cost. Hence, in Figure 6-1 and Figure 6-3, a_l = $100\times$Euclidean distance if length of link $l \leq 0.125$, otherwise $a_l = \infty$. In Figures 6-1 ~ 6-4, SPT, CNS, and GIT are the solution approaches proposed in [12]. Heuristic 1 and heuristic 2 are the solution approaches of DCR proposed in Chapter 5. Each plotted point in Figures 6-1 ~ 6-4 is the mean value over 10 simulation results. The experimental parameters used in the experiments are listed in Table 6-1.

Table 6-1. Experimental parameter settings for DCR problem

Parameter	Value
Number of Nodes	300
Number of Iterations	2000
Improvement Counter	50
Initial Upper Bound	Solution of 1st Getting Primal Feasible
Initial Multiplier	0
Initial Scalar of Step Size	2
Test Platform	CPU: INTEL™ Pentium-III 1.3 GH RAM: 512 MB OS: Window XP

6.1.2 Experiment Results

Figure 6-1 shows the transmission cost of different numbers of source nodes in a random-source model, where the communication radius = 0.125. We can see that the second heuristic proposed in Chapter 5 outperforms the other four solution approaches under all different numbers of source nodes. As the number of source nodes increases, the improvement ratio is more significant. Figure 6-2 shows the transmission cost under different communication radii for fixed 10 source nodes in random-source model. Heuristic 2 still outperforms the other approaches. Note that as the communication radius decreases, the improvement ratio of the second heuristic is larger. This occurs because when the transmission radius is small, only links with shorter distance can exist and routing path needs more hops to reach its destination. Therefore, the advantage resulting from data aggregation will be more significant.

Figure 6-3 shows the transmission cost of different numbers of source nodes in an event-driven model, where the communication radius = 0.125. Figure 6-4 shows the transmission cost for different communication radii for 10 fixed source nodes in the event-driven model. Similar computational results can be observed in Figure 6-3 and Figure 6-4 for the event-driven model. Heuristic 2 still outperforms the other solution approaches. It is interesting to observe that the improvement ratio in the random-source model is often larger than in the event-driven model. This is because sources are randomly selected, not clustered, in the random-source model; thus, the advantages of the tree will be more significant.

In order to measure effectiveness of the second heuristic, we define an improvement ratio as (other approach− heuristic 2) / (heuristic 2) × 100. Table 6-2 shows the improvement ratio for Figures 6-1 ~ 6-4. From Table 6-2, the improvement ratio of the second heuristic over SPT, CNS and GIT is up to 169%, 94% and 18% respectively.

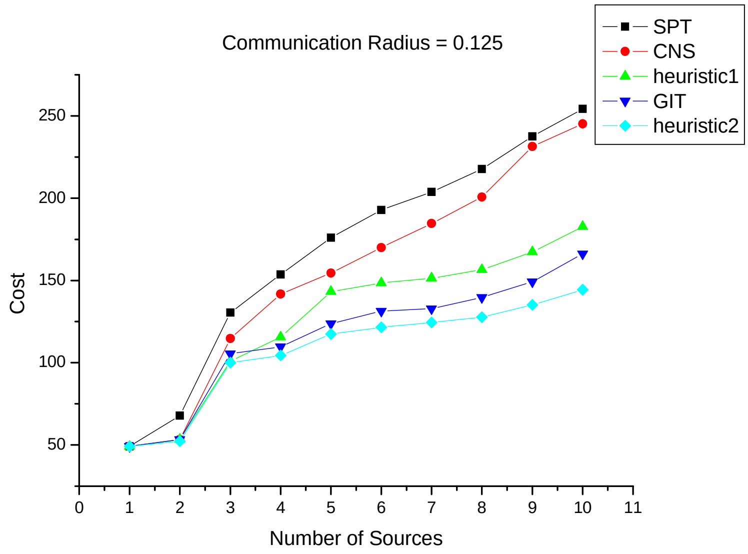


Figure 6-1. Transmission cost v.s. the number of sources in the random-source model

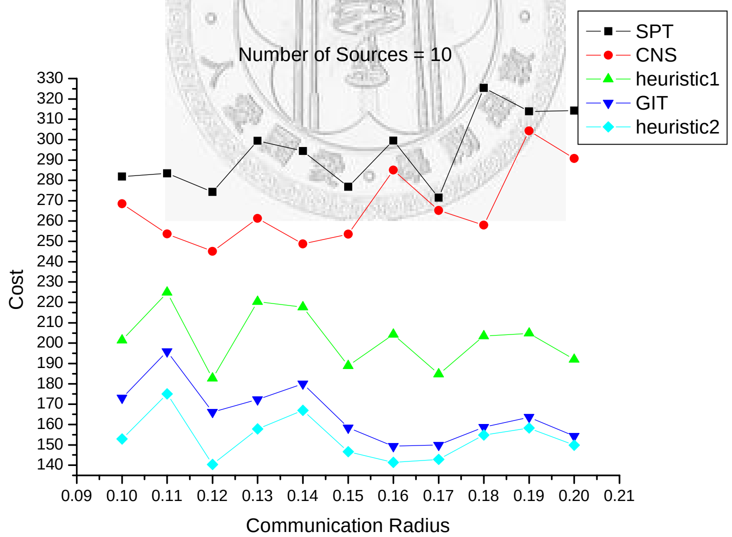


Figure 6-2. Transmission cost v.s. the communication radius in the random-source model

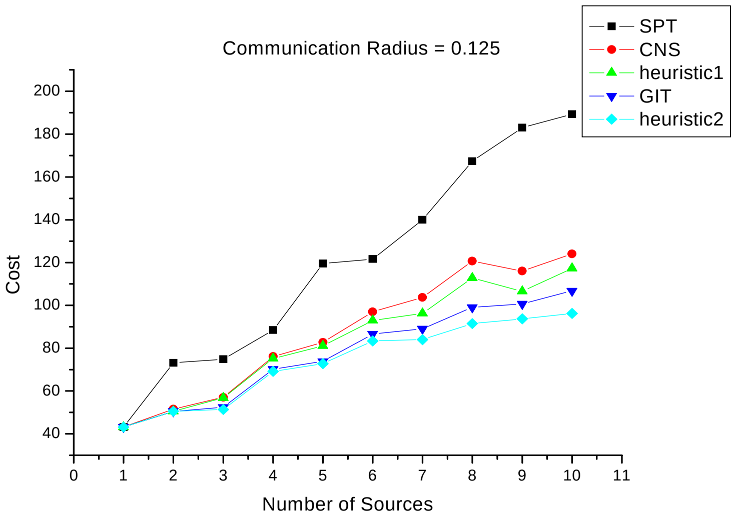


Figure 6-3. Transmission cost v.s. the number of sources in the event-driven model

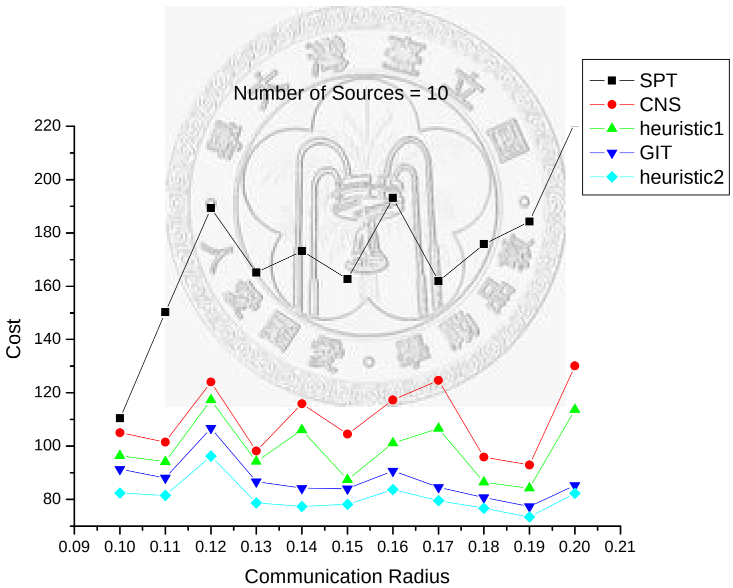


Figure 6-4. Transmission cost v.s. the communication radius in the event-driven model

Table 6-2. Improvement Ratio of Heuristic 2

Improvement Ratio (%)	Figure 6-1	Figure 6-2	Figure 6-3	Figure 6-4
SPT	75	110	97	169
CNS	71	94	33	58
GIT	15	18	11	12

6.2 Computational Experiments of EDCR

6.2.1 Experiment Environments

The proposed algorithms for the EDCR problem are coded in C and run on a PC with INTEL^{TM} PIV-2G. *Max_Iteration_Number* and *Improve_Threshold* are set to 1000 and 25 respectively. The step size coefficient, δ, is initialized as 2 and will be halved when the objective function value of the dual problem is not improved by iterations up to *Improve_Threshold*.

In the EDCR problem, the *random-source* model is tested. In *random-source* model, non-sink nodes are randomly selected to be data source nodes. We construct a network topology consisting of N = 150 sensor nodes randomly placed in a 1×1 square unit area. Since the transmission power falls as $1/d^n$ [14], n $\geqq$2 and d represents the Euclidean distance, if we want to transmit data to a receiver with a certain acceptable level of signal power, the transmission cost will be proportional to the square of the Euclidean distance. Thus, the cost of the power aware function, $e_n(r_n)$, is defined as the square of $100\times$Euclidean distance if the link length does not exceed the maximum transmission radius. The set of all possible transmission radii of sensor node n (R_n) is configured to begin from 0 and extend to the maximum transmission radius. Elements in the radius set are increased by 0.01 successively. In Figure 6-5, the maximum transmission radius is set to be 0.15. Each plotted point in Figure 6-5 and Figure 6-6 is the mean value over 5 experimental results. In all the experiments we assume that there is only one sensing group in sensor networks. In order to show the solution quality of our proposed algorithm, for comparison, we implement three algorithms developed in [12]. Table 6-3 shows the experimental parameter settings in the EDCR problem.

Table 6-3. Experimental parameter settings for EDCR problem

Parameter	Value
Number of Nodes	150
Number of Iterations	1000
Improvement Counter	25
Energy Consumption Function	Square of Euclidean Distance
Initial Upper Bound	Solution of 1st Getting Primal Feasible
Initial Multiplier	0
Initial Scalar of Step Size	2
Test Platform	CPU: INTEL™ Pentium-IV 2.0 GH RAM: 512 MB OS: Linux Red Hat 8.0

6.2.2 Experiment Results

Figure 6-5 shows the transmission cost of different numbers of source nodes in the random-source model. We can see that the second heuristic proposed Chapter 5 is superior to the other four solution approaches under all different numbers of source nodes. In addition, as the number of data source nodes grows the improvement ratio is more significant, which is similar to the DCR model. Figure 6-6 shows the transmission cost under different maximum transmission radii for 8 fixed source nodes in the random-source model. The maximum transmission radius is the maximum allowable transmission range that sensor nodes can chose. The second heuristic still outperforms the other approaches. Note that as the transmission radius decreases, the improvement ratio of the second heuristic increases. This is similar to the results in the DCR model. Another interesting point is that the cost decreases with increasing maximum transmission radius because of the expanded feasible region. However, we can observe that when the maximum transmission radius is increased to a certain point (e.g., 0.17 in Figure 6-6), the cost can not be reduced any further. This is because the energy consumption cost is defined as the square of the transmission radius, and is will be increased rapidly when a large transmission radius is adopted. Therefore, even though the maximum allowable transmission radius is increased, we will not be willing to utilize it.

From Table 6-4, the improvement ratio of the second heuristic in EDCR over SPT, CNS and GIT is up to 59%, 49% and 10% respectively.

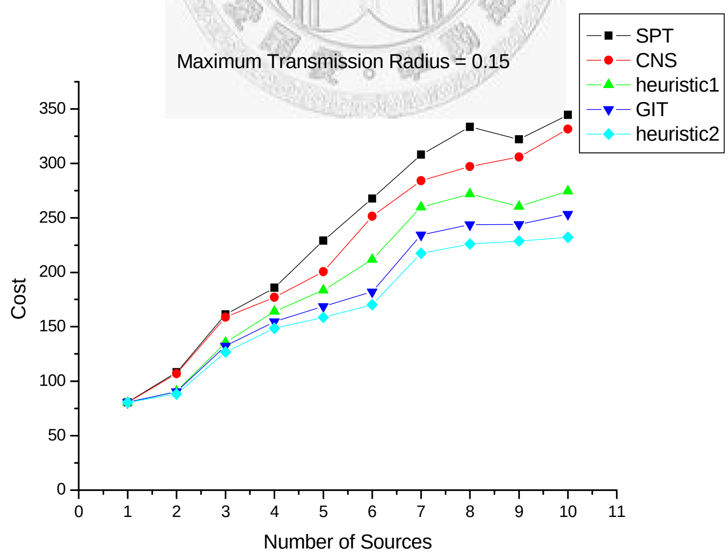


Figure 6-5. Transmission power consumption cost v.s. the number of sources in the random-source model

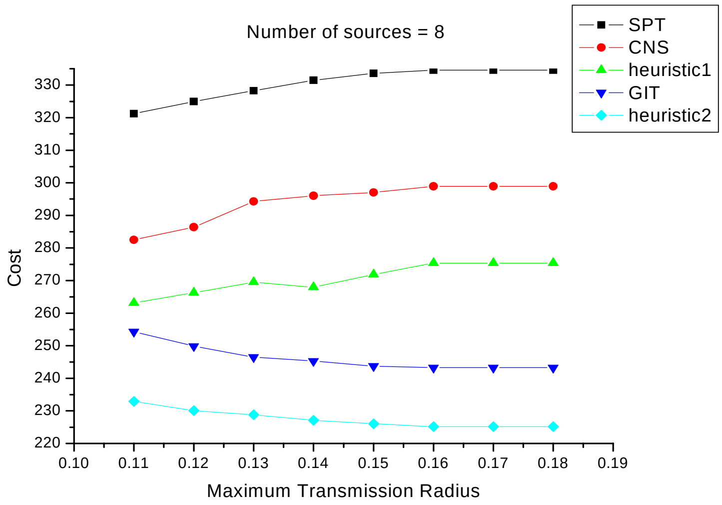


Figure 6-6. Transmission power consumption cost v.s. the maximum transmission radius in the random-source model

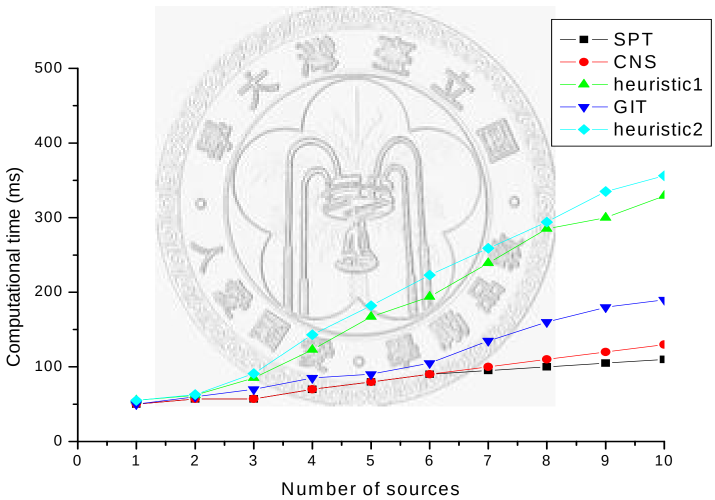


Figure 6-7. Computational time per iteration

Table 6-4. Improvement Ratio of Heuristic 2

Improvement Ratio (%)	Figure 6-5	Figure 6-6
SPT	59	49
CNS	49	33
GIT	10	10

Figure 6-7 shows the computational time comparison of all algorithms per iteration under different numbers of sources. Although our proposed algorithms suffer from a slightly longer computational time, we can get a better data aggregation tree in terms of transmission cost and energy saving. Furthermore, the improvement ratio of our proposed algorithm is more significant when the number of source nodes increases. To summarize, compared to existing heuristics, although our algorithms require a slightly longer computational time, they have better data-centric aggregation capability and better solution quality, particularly when the number of source nodes increases and the sensor node deployment area is large.

6.3 Computational Experiments of Data Aggregation Tree with QoS Routing

6.3.1 Experiment Environments

The proposed algorithms for constructing a data aggregation tree with QoS routing are coded in C and run on a PC with INTEL^{TM} PIV-2G. *Max_Iteration_Number* and *Improve_Threshold* are set to 2000 and 30 respectively. The step size coefficient, δ, is initialized as 2 and is halved when the objective function value of the dual problem is not improved by iterations up to *Improve_Threshold*.

We assume that a sensor network operates in periodic mode where, the sensor nodes periodically report information to the sink node. The network topology comprises $N = 150$ sensor nodes randomly placed in a 1×1 square unit area. The cost of the energy consumption function, $e_n(r_n)$, is defined as the square of $100 \times$Euclidean distance multiplied by energy consumption per millisecond when the sensor node is transmitting data. The set of all possible transmission radii of a sensor node n (R_n) is configured to begin from 0 and extend to the maximum transmission radius. Elements in the radius set are increased by 0.01 successively. To evaluate the solution quality of our proposed algorithm, we implement three existing algorithms for comparison. The GIT and CNS algorithms are proposed in [12] and the third algorithm, CCA, is described in [20]. Table 6-5 shows the experimental parameter settings used in this model.

Table 6-5. Experimental parameter settings

Parameter	Value
Number of Nodes	150
Number of Iterations	2000
Improvement Counter	30
CSMS Parameters	Standard Value [15]
Energy Consumption Function	Square of Euclidean Distance * Energy Consumption per Millisecond During Transmission
Initial Upper Bound	Solution of 1^{st} Getting Primal Feasible
Initial Multiplier	0
Initial Scalar of Step Size	2
Test Platform	CPU: INTEL^{TM} Pentium-III 2.0 GH RAM: 512 MB OS: Linux Red Hat 8.0

6.3.2 Experiment Results

Tables 6-6 and 6-7 show the total energy consumption and maximum end-to-end delay calculated by different algorithms under a different numbers of sources, respectively. It is obvious that Lagrangean relaxation-based algorithm can get better solution quality in terms of total energy consumption and maximum end-to-end delay. Although the maximum end-to-end delay of Lagrangean relaxation-based algorithm is sometimes slightly higher than CNS's, the data aggregation tree constructed by the Lagrangean relaxation-based algorithm can significantly reduce the total energy consumption of a data aggregation tree compared with other heuristics. According to the experimental results, the data aggregation tree constructed by the Lagrangean relaxation-based algorithm can not only maintain a good maximum end-to-end delay, but also reduce total energy consumption needed by aggregation tree to transmit data.

Table 6-6. Energy consumption experimental results with different numbers of sources

Number of Sources	LB	UB	Gap (%)	CCA	CNS	GIT	Imp. Ratio of CCA (%)	Imp. Ratio of CNS (%)	Imp. Ratio of GIT (%)
10	143.57	171.06	19.14	381.09	194.5	254.49	122.78	13.7	48.77
20	189.96	219.79	15.7	292.65	261.11	274.56	33.15	18.8	24.92
30	207.1	264.02	27.48	364.11	342.07	357.52	37.91	29.56	35.41
40	314.27	448.24	42.62	699.27	474.77	555.92	56	5.92	24.02
50	407.97	579.97	42.16	829.51	664.7	687.6	43.03	14.61	18.56
60	392.72	671.33	70.94	805.11	683.84	788.01	19.93	1.86	17.38
70	416.8	722.45	73.33	994	746.37	986.18	37.59	3.31	36.5
80	470.14	762.95	62.28	898.76	907.95	944.06	17.8	19	23.74
90	541.73	899.36	66.01	1198.4	1012.4	1202.8	33.25	12.57	33.74
100	587.93	986.94	67.86	1213.9	1126.6	1474.6	22.99	14.15	49.41

Table 6-7. Maximum end-to-end delay experimental results with different numbers of sources

Number of Sources	Delay LR (ms)	Delay CCA (ms)	Delay CNS (ms)	Delay GIT (ms)
10	44.2	51.13	39.24	94.97
20	43.74	40.73	43.25	57.63
30	43.78	45.41	37.7	81.28
40	60.46	63.89	43.51	132.28
50	66.98	71.54	55.9	139.24
60	62.18	63.52	51.94	168.22
70	61.59	73.43	54.71	191.91
80	65.67	70.59	62.4	157.38
90	71.55	79.96	68.82	204.74
100	73.45	93.36	62.47	226.5

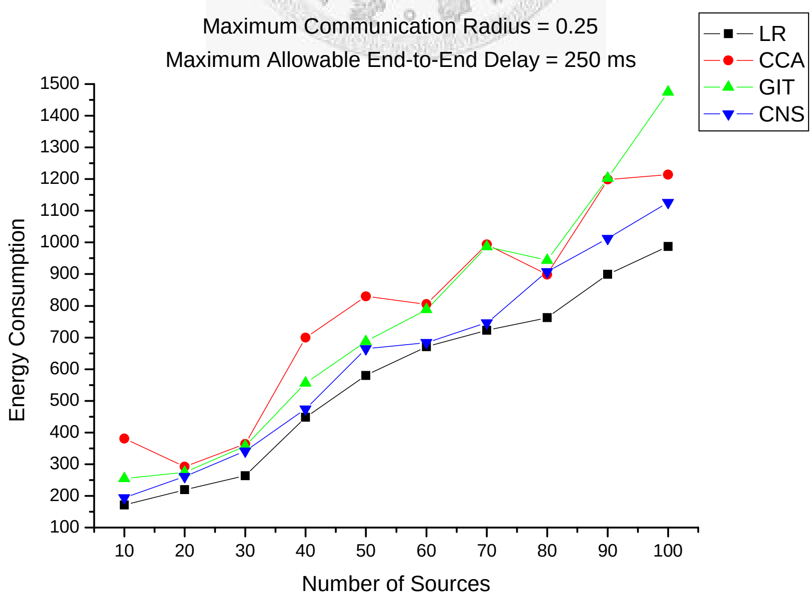


Figure 6-8. Energy consumption with different numbers of source nodes

From Figure 6-8 we can see that as the number of sources increases our proposed Lagrangean based algorithm has a smooth monotonic increment property. In the contrast, the other three heuristics do not have this good property, so we can observe that their curves will wobble as the number of sources increases.

Table 6-8 shows the experimental results under different maximum communication radii with 90 fixed source nodes. In all cases, the Lagrangean relaxation-based algorithm still gets a better solution than the other heuristics. Another interesting point is that as the maximum communication radius increases, the solution of our proposed algorithm changes slightly, whereas the solutions calculated by other algorithms increases dramatically.

Table 6-8. Energy consumption experimental results under different maximum communication radii

Maximum Communication Radius	LR	CCA	CNS	GIT	Imp. Ratio of CCA (%)	Imp. Ratio of CNS (%)	Imp. Ratio of GIT (%)
0.16	738.1	943.07	755.62	1245.12	27.77	2.37	68.69
0.17	773.09	988.95	799.83	1245.12	27.92	3.46	61.06
0.18	750.03	990	777.58	1245.07	31.99	3.67	66
0.19	816.79	1042.38	828.02	1245.07	27.62	1.38	52.43
0.2	843.65	960.15	863.38	1202.84	13.81	2.34	42.58
0.21	856.61	1057.91	900.18	1202.84	23.5	5.09	40.42
0.22	847.85	1120.19	922.33	1202.84	32.12	8.78	41.87
0.23	871.47	1161.63	907.76	1202.84	33.3	4.16	38.02
0.24	856.52	1147.16	934.77	1202.84	33.93	9.14	40.43
0.25	899.36	1198.4	1012.4	1202.84	33.25	12.57	33.74

Table 6-9. Maximum end-to-end delay experimental results under different maximum communication radii

Maximum Communication Radius	Delay LR (ms)	Delay CCA (ms)	Delay CNS (ms)	Delay GIT (ms)
0.16	74.7	89.95	77.51	210.02
0.17	78.12	84.68	82.99	210.02
0.18	67.99	83.36	69.81	209.82
0.19	71.18	83.11	71.51	209.82
0.2	66.58	79.35	69.83	204.74
0.21	78.44	79.3	69.65	204.74
0.22	82.84	78.91	65.35	204.74
0.23	62.15	78.15	63.67	204.74
0.24	74.57	75.37	65.84	204.74
0.25	71.55	79.96	68.82	204.74

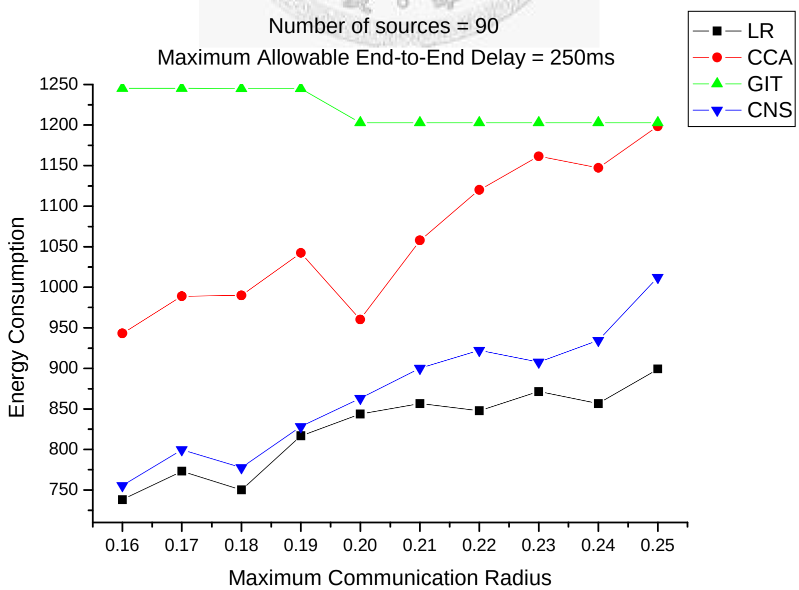


Figure 6-9. Energy consumption under different maximum communication radii

To evaluate the solution quality of our proposed algorithm under different maximum allowable end-to-end delay, we conduct an experiment that changes the maximum end-to-end delay. Table 6-10 summarizes the experimental results of different algorithms under different maximum allowable end-to-end delays. The experimental results are also depicted in Figure 6-10.

Through the experimental results we can observe that when the maximum allowable end-to-end delay constraint is looser, the solution obtained from Lagrangean relaxation-based algorithm is much better. Even if the maximum allowable end-to-end delay constraint is stringent, the Lagrangean relaxation-based algorithm can still calculate a good feasible solution, whereas GIT and CNS can not find a feasible solution. From the perspective of maximum end-to-end delay, although the delay calculated by our proposed algorithm is slightly higher than CNS, the benefit gained from the total energy consumption is significantly larger. This experiment shows that our proposed solution approach can obtain a good feasible solution under different levels of delay, and if the delay constraint is looser the improvement of Lagrangean relaxation-based algorithm over other heuristics is more significant in terms of total energy consumption.

Table 6-10. Experimental results under different maximum allowable end-to-end delays

Maximum Allowable End-to-End Delay (ms)	LR	CCA	CNS	GIT	Delay LR	Delay CCA	Delay CNS	Delay GIT
65	1035.11	N/A	N/A	N/A	64.09	79.96	68.82	204.74
70	1012.24	N/A	1012.4	N/A	67.03	79.96	68.82	204.74
75	987.52	N/A	1012.4	N/A	72.78	79.96	68.82	204.74
80	953.31	1198.4	1012.4	N/A	79.99	79.96	68.82	204.74
85	953.31	1198.4	1012.4	N/A	79.99	79.96	68.82	204.74
90	902.92	1198.4	1012.4	N/A	82.03	79.96	68.82	204.74
95	902.92	1198.4	1012.4	N/A	82.03	79.96	68.82	204.74
100	902.92	1198.4	1012.4	N/A	82.03	79.96	68.82	204.74
105	899.96	1198.4	1012.4	N/A	71.55	79.96	68.82	204.74
110	899.96	1198.4	1012.4	N/A	71.55	79.96	68.82	204.74
115	899.96	1198.4	1012.4	N/A	71.55	79.96	68.82	204.74

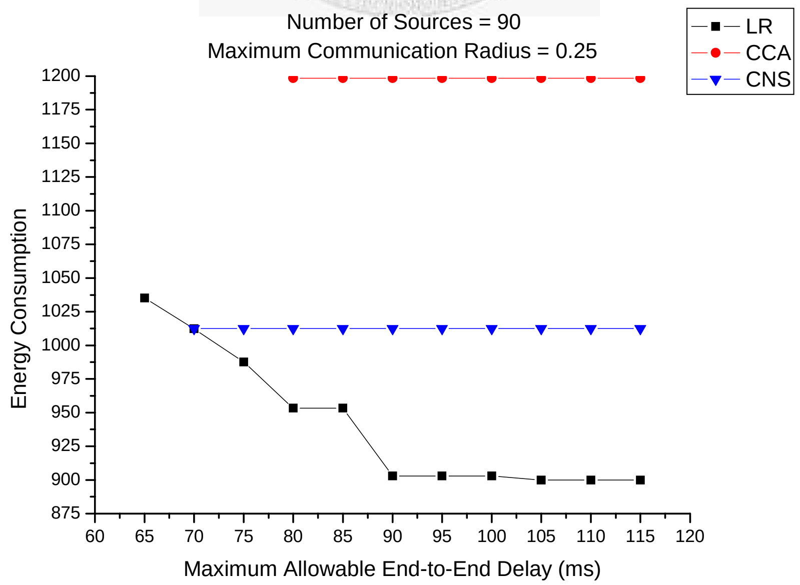


Figure 6-10. Energy consumption under different maximum allowable end-to-end delays

The following experiments evaluate the solution quality of different algorithms under different network sizes with 90 fixed sources and a loose delay constraint. Table 6-11 summarizes the experimental results and Figure 6-11 depicts the results. We can observe that as the network size increases, the Lagrangean relaxation-based algorithm can obtain a better solution than the other algorithms. The improvement ratio is much larger when network size is large. This experiment shows our proposed algorithm can be applied to large networks and still outperform other algorithms.

Table 6-11. Experimental results under different network sizes

Network Size	LR	CCA	CNS	GIT	Imp. Ratio of CCA (%)	Imp. Ratio of CNS (%)	Imp. Ratio of GIT (%)
100	832.08	863.92	857.12	1062.04	3.83	3.01	27.64
125	899.35	1033.4	984.06	1296.13	14.9	9.42	44.12
150	899.36	1198.4	1012.4	1202.8	33.25	12.57	33.74
175	1069	1353.3	1134.11	1253.32	26.59	6.09	17.24
200	1060	1422.86	1218.58	1281.61	34.23	14.96	20.91
225	1136.08	1619.39	1296.81	1555.47	42.54	14.15	36.92
250	1088.42	1631.34	1394.56	1267.68	49.88	28.13	16.47

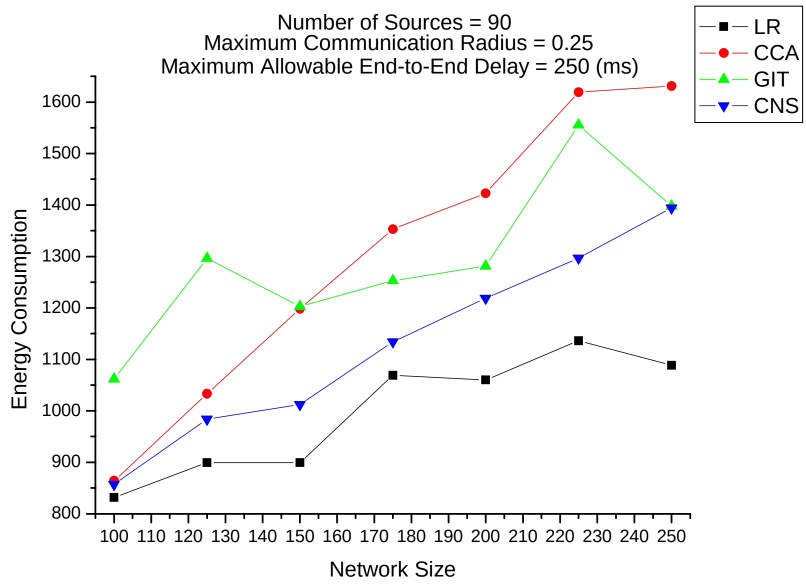


Figure 6-11. Energy consumption under different network sizes

Figure 6-12 shows the computational time of the Lagrangean relaxation-based algorithm per iteration. We can observe that the computational time increases when the number of sources increases. As the number of sources grows, the computational time is increased almost linearly.

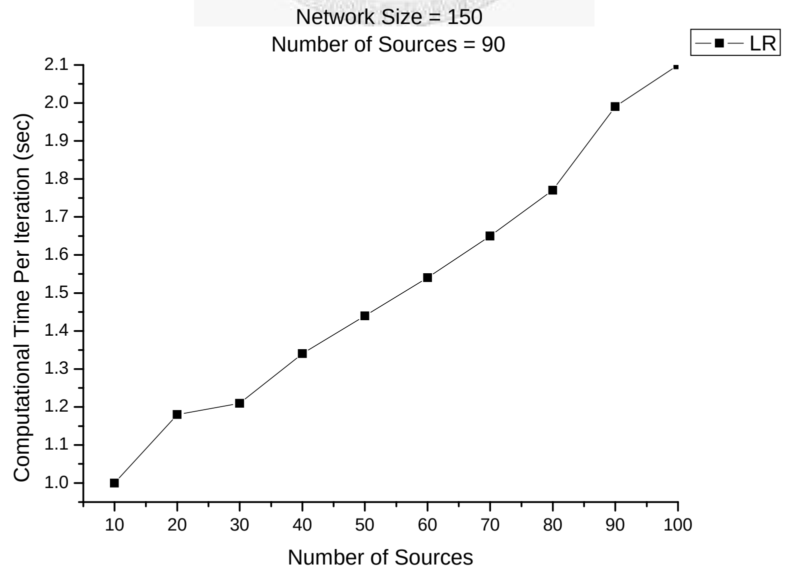


Figure 6-12. Computation time under different numbers of source nodes

Chapter 7 Conclusion and Future Works

7.1 Summary

The deployment of distributed networks characterized by small, inexpensive sensor nodes capable of sensing and wireless communication will soon become the reality due to the rapid advancement in microprocessor, memory, and radio techniques. However, the highly energy-constrained issue should be considered, because recharging the batteries of sensors is infeasible under many severe environments. Data-centric routing could effectively reduce the transmission energy of sensor nodes with data aggregation capabilities in wireless sensor networks. In this thesis, our work emphasizes the construction of an energy-efficient data aggregation tree that possesses good QoS and minimizes the energy consumption of sensor nodes simultaneously. We take the properties of QoS in wireless sensor networks as a new energy consumption metric that can not only maintain the traditional transmission delay, but also simultaneously reduce the energy consumption of sensor nodes operating in idle mode. How to construct a data aggregation tree that is energy-efficient and has QoS properties is a complicated problem needed to be investigated. To address this problem, we have proposed a solution approach based on Lagrangean relaxation to construct an energy-efficient data aggregation tree that considers routing assignment, transmission radius assignment, data retransmissions, and maximum end-to-end delay constraints.

In this thesis, we first propose a mixed integer and linear mathematical formulation for the data-centric routing problem with fixed communication radius. Solution approach based on Lagrangean relaxation and optimization-based heuristics are proposed to solve this problem. Besides the routing assignment, transmission radius assignment is also considered to address the self-organized property of sensor nodes. In the EDCR problem, we jointly consider transmission radius assignment and routing assignment in data-centric sensor networks. Lagrangean relaxation techniques in conjunction with optimization-based heuristics are proposed to solve the EDCR problem. Finally, the effects of data retransmission and maximum end-to-end delay are considered in order to construct a data aggregation tree with QoS routing, while simultaneously minimizing the total energy consumption.

The contribution of this paper can be expressed in terms of the mathematical formulation and experiment performance. For the formulation, we propose three precise mathematical expressions to model the problem of constructing a data aggregation tree efficiently. With regard to performance, the proposed Lagrangean relaxation and subgradient based algorithms outperform other heuristics, such as GIT, CCA, and CNS. According to the experiment results, the Lagrangean-based heuristic for the DCR and EDCR problem is superior to the existing approaches (SPT, CNS and GIT [6]) with improvement ratios of 169%, 94%, and 18% respectively. When considering the effects of maximum end-to-end delay and retransmission the proposed Lagrangean relaxation-based algorithm outperform better than the CCA, CNS, and GIT heuristics by 56%, 29.56%, and 49.41% respectively. From the perspective of the solution quality, we believe that our proposed optimization-based approaches can effectively and efficiently solve the energy-efficient data-centric routing problems in wireless sensor networks.

7.2 Future Works

For the energy-efficient data-centric routing problem in wireless sensor networks, there are still several ongoing research topics to be addressed.

In this thesis we consider the construction of an energy-efficient data aggregation tree with several routing assignment and QoS constraints. However, over time sensors on an aggregation tree will have less residual energy compared to other sensors that are not on the tree. Therefore, it would be beneficial to reconstruct the aggregation tree periodically based on the current aggregation tree to minimize the reconstruction cost.

Cluster-based sensor network is a new approach to establishing sensor networks. As some powerful sensor nodes act as super node namely, cluster head, these super nodes control sensors nearby to form a cluster. How to construct these clusters and the data aggregation tree based on cluster-based routing simultaneously is an extension of my thesis.

Another interesting research question is: How to maximize the system lifetime? This is now a hot topic in wireless sensor networks research. As we want to prolong the system lifetime of sensor networks as much as possible, carefully choosing the aggregation trees to be assigned to a sensor network could significantly sustain sensor network to function appropriately. Thus, given several candidate aggregation trees, how to allocate these candidate trees in different periods in order to maximize the total number of periods is another optimization problem worthy of investigation.

Reference

[1] R. K. Ahuja, T. L. Magnanti and J. B. Orlin, "Networks Flows—Theory, Algorithms, and Applications", *Prentice Hall*, 1993.

[2] K. Akkaya, M. Younis and M. Youssef, "Efficient Aggregation of Delay-Constrained Data in Wireless Sensor Networks", *Internet Compatible QoS in Ad Hoc Wireless Networks*, 2005.

[3] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, no. 8, pp. 102–114, August 2002.

[4] V. Annamalai, S. Gupta, and L. Schwiebert, "On tree-based converge-casting in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference*, Vol. 3, 2003, pp. 1942.

[5] J. Carle and D. Simplot, "Energy Efficient Area Monitoring by Sensor Networks", *IEEE Computer*, Vol. 37, No 2 (2004) 40-46.

[6] J.-H. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks", *Proc. IEEE INFOCOM* 2000, pp. 22–31, Tel Aviv, Israel, Mar. 2000.

[7] M. R. Garey and D. S. Johnson, "Computers and Intractability: A Guide to the Theory of NP-completeness", *Freeman*, San Francisco, 1979.

[8] J. Heidemann, F. Silve, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "Building Efficient Wireless Sensor Networks with Low-Level Naming", *18th ACM Symposium on Operating Systems Principles*, October 21-24, 2001.

[9] C. Intanagonwiwat, R. Govindan and D. Estrin, "Directed Fiffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *ACM/IEEE MOBICOM* August 2000.

[10] K. Kalpakis, K. Dasgupta and P. Namjoshi. "Efficient Algorithms for Maximum

Lifetime Data Gathering and Aggregation in Wireless Sensor Networks", *Computer Networks Journal,* 42(6):697–716, August 2003.

[11] Koushik Kar, Murali Kodialam, T. V. Lakshman, Leandros Tassiulas, "Routing for network capacity maximization in energy-constrained ad-hoc networks", *IEEE INFOCOM 2003 - The Conference on Computer Communications,* vol. 22, no. 1, Mar 2003 pp. 673-681.

[12] B. Krishnamachari, D. Estrin, and S.Wicker, "Modelling Data-Centric Routing in Wireless Sensor Networks", *IEEE INFOCOM* 2002.

[13] V. Raghunathan, C. Schurgers, S. Park, and M. B. Srivastava, "Energy-aware Wireless Microsensor Networks", *IEEE Signal Processing Magazine*, March 2002.

[14] V. Rodoplu, and Teresa H. Meng, "Minimum Energy Mobile Wireless Networks", *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 17, No. 8, August 1999.

[15] S.T. Sheu, T.-H Tsai and JH Chen, "MR 2 RP : The Multi-Rate and Multi-Range Routing Protocol for IEEE 802.11 Wireless Ad Hoc Networks", *ACM/Kluwer Wireless Networks*, Volume 9, Number 3, May 2003 pp. 165-177.

[16] S. Singh, M. Woo, and C. S. Raghavendra, "Power-Aware Routing in Mobile Ad Hoc Networks", *ACM/IEEE MOBICOM* 1998.

[17] I. Solis and K. Obraczka, "The Impact of Timing in Data Aggregation for Sensor Networks", *IEEE International Conference on Communications (ICC)*, Paris, France, June 2004.

[18] H. O. Tan and I. Korpeoglu, "Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks", *ACM SIGMOD Record*, vol. 32, no. 4, pp. 66-71, 2003.

[19] C.-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks", *IEEE Communications Magazine*, June 2001.

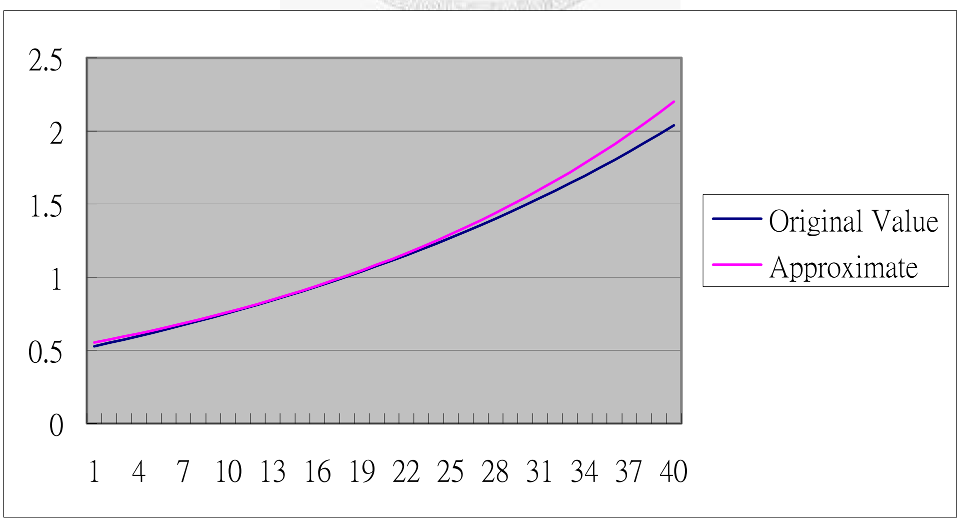
[20] S. Upadhyayula, V. Annamalai, and S. K. S. Gupta," A Low-Latency and Energy-Efficient Algorithm for Convergecast in Wireless Sensor Networks", *IEEE GLOBECOM*, 2003.

[21] M. Younis, K. Akkaya, M. Eltowiessy and A. Wadaa, "On Handling QoS Traffic in Wireless Sensor Networks", *HAWAII International Conference on System Sciences (HICSS-37)*, Big Island, Hawaii, January 2004.

[22] Moustafa A. Youssef, Mohamed F. Younis, Khaled A. Arisha, "A Constrained Shortest-path Energy-aware Routing Algorithm for Wireless Sensor Networks", *WCNC 2002 - IEEE Wireless Communications and Networking Conference,* vol. 3, no.1, March 2002 pp. 682-687.

[23] Y. Yu, B. Krishnamachari, and V. K. Prasanna, "Energy-Latency Tradeoffs for Data Gathering in Wireless Sensor Networks", *IEEE INFOCOM*, March 2004.

Appendix



Delay approximation function versus original function.

簡　　歷

姓　　名： 林 書 平

出 生 地： 台 灣 省 宜 蘭 縣

出 生 日： 中華民國七十年六月十四日

學　　歷：

學 士　　八十八年九月至九十二年六月

國立暨南大學資訊管理學系

碩 士　　九十二年九月至九十四年七月

國立台灣大學資訊管理學研究所

Publication:

1. Hong-Hsu Yen, Frank Yeong-Sung Lin, and Shu-Ping Lin, “Efficient Data-Centric Routing in Wireless Sensor Networks”, *IEEE International Conference on Communications (ICC)*, Saul, Korea, April 2005.