Auction-Based Strategy for Distributed Task Allocation in Wireless Sensor Networks

Neda Edalat

NATIONAL UNIVERSITY OF SINGAPORE

August 2010

Auction-Based Strategy for Distributed Task Allocation in Wireless Sensor Networks

Neda Edalat

(*B.Sc., Shiraz University of Technology, Iran*)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

August 2010

Acknowledgements

It has been a wonderful experience being a Master student at the Electrical and Computer Engineering department at National University of Singapore. I am grateful to my professor and advisor Associate Professor Tham Chen Khong whose experience, guidance and inspiration were integral towards my master degree. I am also deeply indebted to my co-supervisor Dr. Xiao Wendong for his stimulating suggestions and encouragement which helped me in the research and writing of this thesis. I would like to express my gratitude to my husband without which I could not have achieved my academic goals. I would like to thank my colleagues and friends who believed in me and supported me through every moment of this academic journey in Singapore, specifically named Dr. Ong Lee Ling Sharon. Last, but definitely not the least, I would like to thank my parents for the constant emotional and moral support they have provided and without whom I would not have reached so far.

Dedicated to,

My beloved parents, My husband,

and

My advisor

Associate Professor Tham Chen Khong

Contents

Summary

Game theory provides a mathematical tool for the analysis of distributed decision making interactions between agents with conflicting interests. We apply game theory for task allocation in wireless sensor networks (WSNs) where the decision makers in the game are the sensor nodes willing to perform the task to maximize their profits. They have to cope with limited resources (i.e., available energy levels) that imposes a conflict of interest. Given the resource-constrained and distributed nature of WSNs, one of the fundamental challenges is to achieve a fair energy balance amongst nodes to maximize the overall network lifetime. Auction-based schemes, owing to their perceived fairness and allocation efficiency, are among the well-known game theoretic mechanisms for the distributed task allocation. In this work, the real-time distributed task allocation problem is formulated as an incomplete information, incentive compatible and economically-robust reverse auction game. This dynamic scheme accounts for the characteristics of the WSNs environment such as unexpected communication delay and node failure. In the proposed game theoretic model, the distributed best response for bid updates globally converges to the unique Nash Equilibrium in a completely asynchronous manner. This scheme also accommodates for the node failure during task assignment via a recovery phase.

Another problem addressed in this work is the winner determination problem.

Given a distributed pool of bids from bidders (i.e., sensor nodes), a centralized Winner Determination Protocol (WDP) would require costly message exchanges with high energy consumption and overhead. Hence, we propose the Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP) for the reverse auction-based scheme. Our simulation results show a fairer energy balance achieved through this bid formulation in comparison to other well-known static schemes. Moreover, by utilizing the ED-WDP among the numerous distributed resources, the message exchange overhead, energy consumption and delay for winner determination are significantly reduced compared to a centralized WDP.

List of Figures

List of Tables

List of Symbols

Notation	Definition
A_i	Set of strategies for player i
N	Set of players
U_i	Node's Payoff
C_{ij}	Real cost of the node i for executing task j
b_i	Bid value
b_{-i}	Others' bid value
$R(b_i; b_{-i})$	Second lowest price as a reward in the reverse auction game
ψ	Auctioneer's payoff
EP	The price for available energy of each node
S	Task size
BP	Based price of task considering task size and energy price
RT	Sensor node's processor release time
DL	Task deadline
λ	Function of current time and task deadline used in the cost formulation
γ	Function of current time and resource release time used in the cost formulation

RV	Auctioneer's reserve valuation (budget)
$f(bid; RV, C_{ij})$	Distribution of the bidder's belief about the auctioneer's preference
β	Distribution parameter of $f(bid; RV, C_{ij})$ distribution
B_i	Best response of bid's update
$NumPred$	Number of predecessor in task graph
κ	Number of prior rounds of winner's bid value for calculating market price
ρ	Number of iterations for consistent distributed parameter history CDPH
k	Iterations on which the bidder continuously wins with consistent β
T_0	Initial utility value
θ	Step
α_0	Initial α value
β_0	Initial β value
ϵ	Threshold for step
a	Scaling parameter in bid formulation
b	Preferred coefficient in bid formulation
T_W	Waiting time before the winning node sends its bid

T	Number of tasks (iterations) in task graph
$g(bid; \alpha, RV)$	Market trend distribution in auctioneer's belief
α	Distribution parameter of $g(bid; \alpha, RV)$
$\hat{g}$	Safeguard factor
CW	Contention Window
N_{round}	Number of rounds for negotiation
T_{total}	The total delay upon receiving a winning bid
U_{ex}	Auctioneer's expected utility function
μ	Adjustment parameter in algorithm 2

Abbreviations

WSNs	Wireless Sensor Networks
WDP	Winner Determination Protocol
D-WDP	Distributed Winner Determination Protocol
C-WDP	Centralized Winner Determination Protocol
MAC	Medium Access Control
CW	Contention Window
DAG	Directed Acyclic Graph
DVS	Dynamic Voltage Scaling
EST	Earliest Start Time
LST	Latest Start Time
CN	Critical Nodes
MM	Mission Manager
NE	Nash Equilibrium
RV	Reserved Valuation
TD	Task Deadline
RT	Processor Release Time
MP	Market Price
CDPH	Consistent Distribution Parameter History
CW	Contention Window

Publications

Edalat, Neda; Tham, Chen-Khong; Xiao, Wendong; An Auction-Based Strategy for Distributed Task Allocation in Wireless Sensor Networks, submitted to *IEEE Transactions on Mobile Computing (TMC)*, submitted at August 2010.

Edalat, Neda ; Xiao, Wendong; Tham, Chen-Khong; Keikha, Ehsan, Distributed winner determination protocol for reverse auction-based task allocation in pervasive computing, *Source: 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops, PERCOM Workshops 2010,* p 780 - 783,March 2010.

Edalat, Neda; Xiao, Wendong; Tham, Chen-Khong; Keikha, Ehsan; Ong, Lee-Ling, A price-based adaptive task allocation for wireless sensor network, *Source: 2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, MASS '09*, p 888 - 893, October 2009.

Chapter 1

Introduction

In the last few years, wireless sensor networks (WSNs) [1,2] have drawn the attention of the research community, driven by a wealth of theoretical and practical challenges. This progressive research in WSNs explored various new applications enabled by larger scale networks of sensor nodes capable of sensing information from the environment, process the sensed data and transmits it to the remote location. WSNs are mostly used in low bandwidth and delay tolerant applications ranging from civil and military to environmental and healthcare monitoring. WSNs are generally composed of a large number of sensors with relatively low computation capacity and limited energy supply [2].

One of the fundamental challenges in WSNs is attaining proper resource management via energy efficient design and operation. In-network processing emerges as an orthogonal approach to significantly decrease network's energy consumption by eliminating redundancy and reducing communicated information volume [2]. The in-network processing applications may require computationally intensive operations to be performed in the network subject to certain constraints. For instance, in target tracking applications [3], sensors collaboratively measure and estimate the location of moving targets or classify targets. To conserve energy and reduce the communication load, operations such as Bayesian Estimation and data fusion must be executed in the WSN. In the case of tracking or detecting multiple high-speed moving targets, these operations must be finished in a timely manner with an eye toward limited energy consumption. For video sensor networks, in-network processing such as image registration and distributed visual surveillance [4] may demand considerable computation power that is beyond the capacity of each individual sensor. Thus, it is desirable to develop a general solution to provide the minimum computation capacity required by in-network processing. In WSNs with densely deployed nodes, a promising solution is to have sensors collaboratively process information with distributed computation load among sensors. To achieve application independent parallel processing, distributed and real-time task scheduling and decision making are the problems that must be solved.

Decision-makings and resource allocations require gathering and coordinating information spread across sensors' information processes and software agents. Requiring these interacting entities to share, all their local information is infeasible since this could lead to information overload or the violation of privacy issues. Thus, for the benefits of recent sensor technology developments to reach end users, without overloading them, automated and distributed information and resource management algorithms need to be developed that can provide decision-making entities with access to significant time-critical information, while filtering out irrelevant data.

An ideal solution to the resource management problem through the task allocation in WSNs is the development of a system architecture and distributed algorithms that:

a) is generalizable and can be adapted to wide range of sensor network domains

b) provides for distributed, decentralized control

c) results in optimal (or sufficiently optimal) allocation of sensor resources.

This work developed the comprehensive resource management via the efficient task allocation in WSNs that possesses the above attributes, and thus can successfully account for the heterogeneity of the sensors, threat levels in the environment and provide for distributed and decentralized control.

1.1 Problem Definition and Motivation

Applications for wireless sensor networks may be decomposed into tasks which are deployed and scheduled on different sensor nodes in the network. In other words, the application level tasks are decomposed into the low-level tasks which can be executed at the sensor nodes. The low-level task sequences and dependencies are represented by a directed acyclic graph (DAG). In a DAG graph, the vertices represent low-level tasks and the edges represent the precedence relationship between tasks. Task allocation algorithms assign these tasks to specific sensor nodes in the network for execution. This model of real-time task allocation is illustrated in Figure 1.1.

In static task allocation, given the DAG and set of initial available resources, the queue of tasks is assigned to sensor nodes before the task execution started. However, given the uncertain, unpredictable and distributed nature of WSNs, existing static (offline) task scheduling [5–12] may not be practical. Therefore, there is a need for a real-time and adaptive task allocation scheme that accounts for the characteristics of

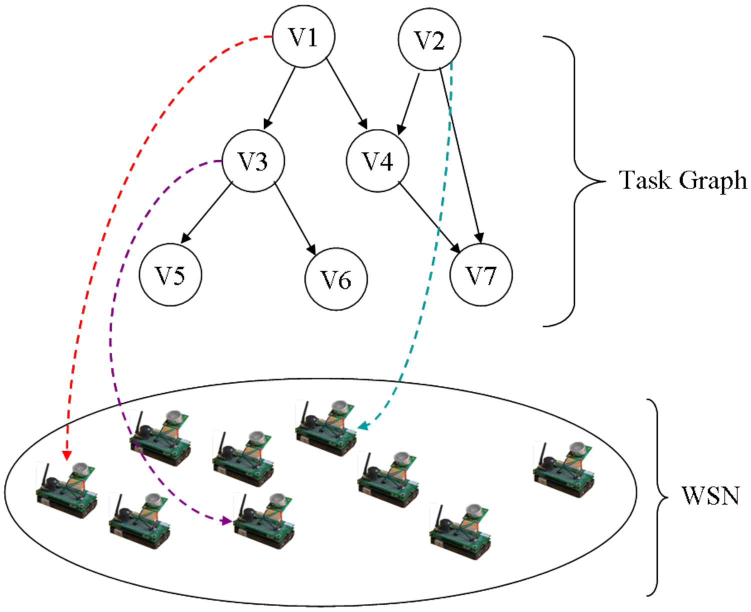


Figure 1.1: Real-time task allocation in a WSN

the WSNs environment such as *unexpected communication delay, packet loss and node failure during task assignment.* Considering the resource-constrained and distributed nature of WSNs, one of the fundamental challenges in WSNs is to achieve a fair energy balance amongst nodes to maximize the overall network lifetime through task allocation and in-network processing. However, the proposed static task allocation algorithms with energy balancing consideration [5–7] did not take into account the real energy availability at each epoch of task allocation. Thus, the design of an adaptive and real-time task assignment scheme which considers available resources at each epoch of task allocation is of essential necessity. On the other hand, due to distributed nature of WSNs, distributed task allocation and decision making schemes with small computation and communication complexities are demanded [48–51].

Game theory provides a mathematical tool for the analysis of distributed decision making interactions between agents with conflicting interests [13–15]. We apply game theory for task allocation in wireless sensor networks (WSNs) where the decision makers in the game are the sensor nodes willing to perform the task to maximize their profits. They have to cope with limited resources (i.e., available energy levels) that imposes a conflict of interest.

Auction-based schemes [17–27], owing to their perceived fairness and allocation efficiency, are among the well-known market-based schemes [33,61] and game theoretic-based mechanisms that can be used for distributed task allocation to achieve *fair energy balance amongst sensor nodes.*

In this work, the real-time distributed task allocation problem is formulated as an *incomplete information reverse auction game.* In the proposed game the second-lowest-price sealed-bid is utilized as a dominant strategy for the players which are the sensor nodes. The main goal is to find the suitable sensor node (player) to perform the arrival task with the goal of maximizing the energy balance among the resource-constrained sensor nodes and consequently the overall network lifetime while considering application's deadline.

In an auction design, a process is said to be incentive compatible if all of the players fare best when they truthfully reveal any private information during the auction. Truthfulness, individual rationality and budget balance are the three critical properties required to design economic-robust reverse auctions that create the incentive for the bidders and auctioneer to participate in the reverse auction game. In truthful auctions, the dominate strategy for bidders is to bid truthfully, thereby, eliminating the fear of market manipulation and the overhead of strategizing over others. With the true valuations, the auctioneer can allocate the task efficiently to sellers who value it the least.

Given a game where the group of players interactively make their decisions, it is natural to ask "What will the outcome of a game be like?" The answer is given by Nash equilibrium, which is an equilibrium where everyone plays the best strategy when taking decision-making of others into account. Then, the next questions are "Does a Nash equilibrium always exist in a game?" and "Is it unique?". In our proposed reverse auction the distributed best response for bid updates converge globally to the unique Nash Equilibrium in a completely asynchronous manner. We show that the socially optimal allocation can always be achieved at an equilibrium where no node can increase its profit by unilaterally changing its bids. The most significant challenge in designing this auction game is how to make the auction economically robust while enabling task allocation among sensor nodes. The proposed reverse auction creates the incentive compatibility for the players and meets the conditions to achieve the economic-robust auction. The winner determination in the auction-based scheme such as [21–28] essentially requires costly message exchanges with enormous overheads. To address such challenging issues, a novel Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP) for winner determination in the auction-based task allocation is proposed and compared with other winner determination schemes. The proposed protocol that has the interpretation of contention-based MAC (Medium Access Control) protocol operates in two phases. In the first phase, *elimination via budget value*, the number of players are eliminated via the budget value which is set by auctioneer and is the value that auctioneer wills to pay for arriving task. The consequence of this phase is to eliminate the players with low amount of available energy levels for the rest of competition. In the second phase, *waiting time reduction*, the duration of ideal listening mode as one of the important source of energy consumption for the players are significantly limited.

1.2 Outline of the Thesis

In this dissertation, the adaptive, distributed and real-time task allocation strategy in WSNs is proposed. The outline of the dissertation is as follows:

Chapter 2 presents the sufficient backgrounds on the basic of game theory and auction theory as the main tools of our proposed solution for real-time task allocation and distributed decision making problem. The related works are also presented in Chapter 2.

The market-based architecture and game model for task allocation problem are introduced in Chapter 3. In this chapter, the architecture components, the game model of reverse auction and the required economic properties for this model are discussed.

The task allocation phases and bid formulation are presented in Chapter 4. The bidder's payoff function, the Nash Equilibrium and distributed iterative best response update algorithm are presented. This chapter also provides the proofs for the most of theorems and lemmas in this dissertation. Finally, in this chapter, the fast recovery algorithm in case of sensor node's failure based during the task assignment phase is discussed.

In Chapter 5, the decision making protocols for winner determination in reverse auction-based task allocation are introduced. Our proposed Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP) which operates in two phases are presented and evaluated. Finally, this chapter explains how the auctioneer achieves the best budget value based on the adaptive algorithm that runs in an asynchronous manner.

Chapter 6 shows the simulation results and performance evaluations. The simulation results show a fair energy balance achieved through the bid formulation. The convergence of the proposed algorithms are also illustrated. Another set of simulations are carried out to evaluate the energy balancing, total energy consumption and total schedule length in our proposed distributed task allocation method and winner determination protocols.

The conclusion of this work and future work are presented in Chapter 7.

Chapter 2

Background

2.1 Basics of Game Theory

Game theory [13–16] is a branch of applied mathematics, and it is used to analyze problems with conflicting objectives among interacting decision-makers. It has been used primarily in economics and has also been applied to other areas, including politics, biology and networking. A broad overview of game theory and its application to different problems in networking and communications can be found in [29–32] and the references therein. More recently, researchers are using game theory to deal with job and resource allocation in wireless networks and services: the decision makers in this game are the wireless service providers and endusers. These decision makers have to deal with a limited network and radio resources that imposes a conflict of interest between them. A game consists of players, the possible actions of the players, and consequences of the actions. For notational purpose, a game is always expressed by the (N, S, U) tuple, where N denotes the set of players, S denotes the strategy space of the players and U denotes the set of utility functions. The players are decision-makers, who choose how they act. Formally, a game can be defined by a conflict among several (two or more) players, where the players strive to ensure the best possible consequence according to their preferences. The preferences of a player are expressed through a utility function, which maps every consequence to a real number, or with preference relations, which define the ranking of the consequences. An utility function can be defined as a mathematical characterization that represents the benefits and cost incurred by the players in the game. The most fundamental assumption in game theory is rationality. Rational players are assumed to always maximize their profit or payoff. If the game is not deterministic, the players maximize their expected payoff. The idea of maximizing the expected payoff was justified by the seminal work of von Neumann and Morgenstern in 1944 [29]. Maximizing ones payoff is often referred to as selfishness. This is true in the sense that all players try to gain the highest possible utility. However, a high utility does not necessarily mean that the players act selfishly. Any kind of behavior can be modeled with a suitable utility function. A game describes the actions the players can take as well as the consequences of the actions. The solution of a game is a description of outcomes that may emerge in the game if the players act rationally and intelligently. Generally, a solution is an outcome from which no player wants to deviate unilaterally.

2.1.1 Classification of Games

Before we proceed any further, let us discuss the classifications of games that are relevant to this research.

⧫ **Cooperative Vs. Non-cooperative Games**: Game theory can be divided into non- cooperative and cooperative game theory. In cooperative games, the joint actions of groups are analyzed, i.e., what is the outcome if a group of players cooperate. Cooperative game theory looks at reasonable or fair outcomes when players form coalition and share resources. It answers questions such as which players will form a coalition and how will resources be divided within these coalitions. In non-cooperative games, the actions of individual players are considered where cooperation from each of the players must be selfenforcing. Most game theoretic research has been conducted using non-cooperative games, but there are also approaches using cooperative games.

⧫ **Complete Vs. Incomplete-information Games**: Depending on whether or not each player knows the other players payoff functions, a game can be formulated either as a complete or incomplete information game. If every player is aware of the strategies and utilities of all the other players, the game is said to have complete information. If not, the game has incomplete information. Given a situation, i.e., some information, a game can be a complete or incompleteinformation game depending on the goal we are seeking.

⧫ **Pure strategy Vs. Mixed strategy**: If a player selects one of the strategies from his strategy set with probability 1, then the player is playing a pure strategy. In contrast, in mixed strategy profile, a player has several pure strategies in the strategy space and the player decides to play each of the pure strategies with some probability, i.e., the selection is randomized. Thus, in mixed strategy, the strategy space has some probability distribution which corresponds to how frequently each of the strategies is chosen.

⧫ **Static games Vs. Dynamic games**: In a static game, the players make decisions only once, i.e., the players have only one move. The strategies are chosen simultaneously by the players without knowledge of other players strategies. Even though the decisions can be taken at different time instants, the game is simultaneous because each player has no information about the decisions of others; thus, it is as if the decisions are made simultaneously. In contrast to the static games, if the players interact multiple times by playing the game iteratively, the game is called a dynamic, or repeated game. Unlike static games, players may have some information about the strategy profiles of other players and thus may contingent their play on past moves.

2.1.2 Analyzing Games and Nash Equilibrium

Once the game is formulated, it needs to be solved. Solving a game means predicting the strategy of the players, considering the information the game offers and assuming that the players are rational. There are several possible ways to solve a game: iterated dominance, best response, backward induction and many more. A detailed study on these techniques can be found in [14, 15]. In this research, we focus on the best response strategy. The best response of a player $i \in N$ is to choose a strategy $s_i \in S$ when the strategy vector s_{-i} is chosen by all the opponents. The objective of player i is to maximize the utility u_i. More formally, the best response strategy can be defined as follows.

Definition 1. *The best response br_i of player i to the opponents strategy profile s_{-i} is a strategy s_i such that:*

$$br_i = \arg\max_{s_i \in S} u_i(s_i, s_{-i}) \tag{2.1}$$

From the above definition, one can find that if the strategies taken by the players are mutual best responses to each other, then no player would like to deviate from the given strategy profile. To identify such strategy profiles, John Nash introduced the famous equilibrium concept known as Nash equilibrium [37]. The concept of Nash equilibrium can be formally defined as follows.

Definition 2. *The strategy profile s^* constitutes a Nash equilibrium if and only if, for each player i,*

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \quad \forall s_i \in S \tag{2.2}$$

The above definition means that in a Nash equilibrium state, none of the players would unilaterally change the strategy to increase the utility. Thus Nash equilibrium brings the game to a steady state, from which the players would not like to deviate as that would not increase their benefits any more.

2.2 Basics of Auction Theory

An auction is the process of buying and selling goods by offering them up for bids (i.e., an offered price), taking bids, and then selling the item to the highest bidder. In economic theory, an auction is a method for determining the value of a commodity that has an *undetermined* or *variable* price. In some cases, there is a minimum or reserve price; if the bidding does not reach the minimum, there is no sale. Traditional auctions involve single seller and many buyers. The buyers compete among themselves to procure the goods of their choice by placing a bid, which they feel most appropriate.

2.2.1 Types of Auctions

Kalagnanam and Parkes [44] have suggested a framework for classifying auctions based on six major factors as outlined below.

1) Resources/ Tasks: Resources/ Tasks are the entities over which the negotiation in an auction are conducted. The resources could be a single item or multiple items, with single or multiple units of each item.

2) Market structure: There are three types of market structures in auctions. In *forward auctions*, a single seller sells resources to multiple buyers. In *reverse auctions*, a single buyer attempts to source resources from multiple suppliers, as is common in procurement. Auctions with multiple buyers and sellers are called *double auctions* or exchanges.

3) Preference structure: The preferences define an agent's utility for different outcomes in the auction. For example, when negotiating over multiple units, agents might indicate a decreasing marginal utility for additional units. An agent's preference structure is important when negotiation occurs over attributes of an item, for designing scoring rules used to signal information, etc.

4) Bid structure: The structure of the bids within the auction defines the flexibility with which agents can express their resource requirements. For a simple single unit, single item commodity, the bids required are simple statements of a willingness to pay/accept. However, for a multi unit, identical items setting bids need to specify price and quantity. This introduces the possibility for allowing volume discounts. With multiple items, bids may specify all or nothing, with a price on a bundle of items.

5) Winner determination: Other phrases which are used synonymously with winner determination are market clearing, bid evaluation, and bid allocation. In the case of forward auctions, winner determination refers to choosing an optimal mix of buyers who would be awarded the items. In the case of reverse auctions, winner determination refers to choosing an optimal mix of sellers who would be awarded the contracts for supplying the required items. In the case of an exchange, winner determination refers to determining an optimal match between buyers and sellers. The computational complexity of the winner determination problem is an important issue to be considered in designing auctions.

6) Information feedback: An auction protocol may be a direct mechanism or an indirect one. In a direct mechanism, such as a sealed bid auction, agents submit bids without receiving feedback, such as price signals, from the auction. In an indirect mechanism, such as an ascending-price auction, agents can adjust bids in response to information feedback from the auction. Feedback about the state of the auction is usually characterized by a price signal and a provisional allocation, and provides sufficient information about the bids of winning agents to enable an agent to redefine its bids.

There are several kinds of auction models as shown in 2.1. Depending on whether the bidding strategies of each of the bidders are disclosed to the other bidders, open and closed bid auctions are designed. In open auctions [40, 41], bids are open to everybody so that a players strategy is known to other players and players usually take their turns one by one until the winner(s) evolve. Bids generated by players in open bid auction can be either in increasing or decreasing order. Couple of famous increasing bid open auction are English auction [42] and Yankee auction. Dutch auction on the other hand is a famous decreasing open bid auction. Dutch-style auction satisfies the property that privacy of losing bids is preserved after auction closes [43]. An important perspective of increasing auction is that it is more in the favor of bidders than the auctioneers. Moreover, increasing open bid auction helps bidders in early round to recognize each other and thus act collusively. Increasing auction also detract low potential bidders because they know a bidder with higher bid will always exceed their bids. Closed bid (or sealed bid as they are more popularly known as) auctions are opposite to open bid auctions and bids/strategies are not known to everybody. Only the organizer of the auction will know about the bids submitted by the bidders and will act accordingly. Bids are kept secret until the opening phase, and then all bids are opened and compared to determine the highest one. Thus, closed bid auctions do not promote collusion. Couple of the famous closed bid auctions are first price sealed bid auction and second price sealed bid auction. In a first price auction, the winners payment is equal to the winners bid while in a second price auction, the winners payment is equal to the second highest bid. Open bid auctions are best generalized as complete information games while closed auctions are incomplete information games.

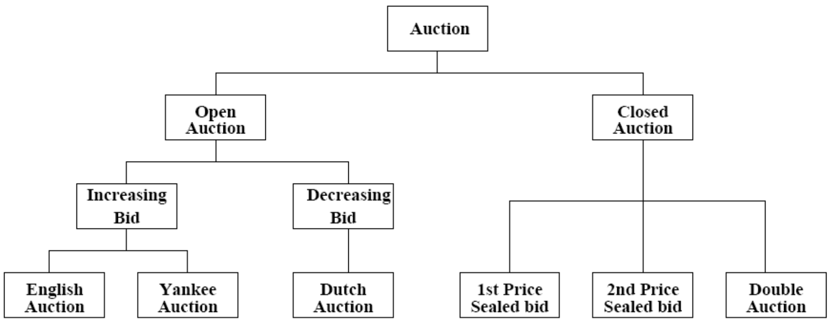


Figure 2.1: Different types of auction

2.2.2 Auction Design

Good auction design is important for any type of successful auction and often varies depending on the item on which the auction is held. The auctions held in Ebay are typically used to sell an art object or a valuable item. Bidding starts at a certain price defined by auctioneer and then the competing bidders increase their bids. If a bid provided by a bidder is not exceeded by any other bidder then the auction on that object stops and final bidder becomes the winner. There are three important issues behind any auction design. They are (i) attracting bidders (enticing bidders by increasing their probability of winning), (ii) preventing collusion thus preventing bidders to control the auction and (iii) maximizing auctioneers revenue. It is not at all intended that only bidders with higher purchasing power should get most of the items. The goal is to increase competition among the WSPs and bring fresh new ideas and services. As a result, it is necessary to make even the low potential bidders, who have a low demand of items, interested to take part in the auction.

2.3 Related Work

2.3.1 Task Allocation in Wireless Sensor Networks

The wireless sensor networks (WSNs) are envisioned to observe large environments at close range for extended periods of time. WSNs are generally composed of a large number of sensors with relatively low computation capacity and limited energy supply [1]. One of the fundamental challenges in WSNs is attaining energy efficiency at all levels of design and operation.

Applications for WSNs may be decomposed into the low-level tasks which are deployed and scheduled on different sensor nodes in the network. Task allocation algorithms assign these tasks to specific sensor nodes in the network for execution. In static task allocation, given the DAG and the initial available resources, the queue of tasks is assigned to sensor nodes before the task execution started. However, given the uncertain, unpredictable and distributed nature of WSNs, existing static (offline) task scheduling [5–12] may not be practical. Therefore, there is a need for a real-time and adaptive task allocation scheme that accounts for the characteristics of the WSNs environment such as *unexpected communication delay, packet loss and node failure during task assignment.*

Considering the resource-constrained and distributed nature of WSNs, one of the fundamental challenges is to achieve a fair energy balance amongst nodes to maximize the overall network lifetime through task allocation. However, the proposed static task allocation algorithms with energy balancing consideration [5–7] did not take into account the real energy availability at each epoch of task allocation. Thus, the design of an adaptive and real-time energy balanced task assignment scheme which considers available resources at each epoch of task allocation is of essential necessity.

Existing work on static task scheduling [5–12] achieves the energy balance objective by regulating the energy consumption via Dynamic Voltage Scaling (DVS) [62]. DVS, by decreasing the CPU speed reduces computational energy consumption; however this results in a longer schedule length. In a couple of works by Prassana [6, 7], given each nodes initial available energy, each cluster of tasks are assigned to the sensor nodes as *a whole rather than adaptively allocating the individual task at each epoch* by considering resource availability at that epoch.

Pricing scheme [45, 46] for the task scheduling problem is emerged as a promising solution to achieve a fair energy balance amongst nodes; since this technique adapt to changes in the environment. The load balancing and pricing has been recently discussed in the literature for grid computing [46]. However, the application of the pricing schemes for task allocation in WSNs with limited resources, is almost unexplored.

In this work, the reverse auction game is proposed as the well-known pricing solution for task allocation problem, one which places emphasis on a *fair energy balance* among nodes in order to *maximize network lifetime*. The task allocation is modeled as a market architecture. The consumer is modeled as an *auctioneer* and the sensor nodes represent the *sellers* in our scheme. When a task is to be allocated, the auctioneer broadcasts information about the tasks to the sellers. Each seller calculates its cost based on its available energy level and application delay constrain on the proposed cost formulation. Then each seller bids to achieve the Nash Equilibrium as a desired output of reverse auction game, through distributed adaptive update algorithm. Sellers with higher bids are likely to have less remaining energies in future, so the bid of the seller can be adjusted to influence the decision making for the task allocation. In the case of an unexpected situation such as node failure *during the task assignment*, this scheme would run the dynamic recovery phase. Whereas, in [47] failure is considered only for the case that node failure happens *before the task assignment phase* and generated an alternative schedule. This proposed scheme is a scalable and adaptive solution for distributed task allocation in WSNs. This scheme is scalable as it is independent of the number of available nodes and will adapt if the number of nodes changes. As the allocation is performed in real-time, each node would adaptively react to the changes in resource availability and utilize new available resources at each time epoch, hence, it is the adaptive scheme.

2.3.2 Market-based Architecture for Resource Management

Market-based architecture [33–35] provides a valuable and principled paradigm for designing systems that solve the dynamic and distributed resource allocation problem based on the pricing systems; since market-based schemes have the inherent ability to deal with non-commensurate entities. Markets can be used for finding optimal allocations in a cooperative environment and there is a one-to-one mapping between a sensor management scenario and traditional market. Market-based algorithms have been used for the distributed resource allocation in a wide ranging of scenarios including bandwidth allocation [55], network information services [56], distributed operating systems [57] and electric load distribution [58]. Market-based mechanisms have also been applied to distributed scheduling with promising results [59]. This approach uses the fundamentals of economic theory for designing and implementing resource allocation problems. The basic idea behind these algorithms is that price-based systems facilitate efficient resource allocation in computational systems, just as they do in human societies. Resource-seeking entities are modeled as independent agents, with autonomy to decide about how to use their respective resources. These agents interact via a market that uses a pricing system to arrive at a common scale of value across the various resources. The common-value scale is then used by the individual agents for making trade-off decisions about acquiring or selling goods. Market-oriented approaches usually involve auction mechanism for scheduling [60], where agents send bids to an auctioneer for various commodities and the auctioneer determines the resource allocations. Reverse auction [21] is type of auction where the role of buyer and seller are reverse and the primary objective is to drive purchase prices downward. Single buyer and multiple sellers have been used in reverse auction, such as, the procurement system. Its goal is to find the suitable resources (cheapest sellers) to accomplish the consumer's arrival task.

In this work, we model the resource management scenario as a competitive market, wherein the sensor manager holds a reverse auction to buy the various goods produced by the sensors and the communication channels.

2.3.3 Auction-based Resource and Task Allocation

Auction-based schemes [19], [20], owing to their perceived fairness and allocation efficiency, are among the well-known game theoretic-based mechanisms that can be used for task allocation. Recent auction-based methods that have been proposed for the purpose of resource allocation problem in different applications can be seen in [21–27]. Huang et al. in [25] proposed two auction schemes, SNR auction and power auction to determine relay selection and relay power allocation in cooperative communications. In another works from Huang et al. [26], the similar auction scheme has been used for spectrum sharing. Fu et al. [24] used auction as a stochastic game for repeated resource allocation. In [22], reverse auctions were used to formulate the scheduling problem in multi-rate wireless systems. In their framework, the users compete against each other to sell a set of slots to the base station. In [23], auction-based resource allocation in a grid computing system was proposed. In this work, a resource consumer invites a public bidding where the resource provider bids according to his load. One of the important issues in any auction scheme design is to be incentive compatible and can guarantee the trustworthiness of the bidders. However, these issues were not addressed in the above mentioned works. Moreover, the winner determination method provided by these auction methods are centralized and require a high message exchange overhead between the base station or resource consumer and all the users. This would also result in high latency for determining the winning bidder.

Chapter 3

Market-Based Architecture and Game Model for Task Allocation

3.1 Introduction

Market-based architecture provides a valuable and principled paradigm for designing systems to solve the dynamic and distributed task allocation problem. We have modeled the task allocation scenario as a competitive market where the main goal is to find the suitable sensor nodes to perform the arrival task with the goal of maximizing the whole network's lifetime among the resource-constrained sensor nodes. Game theory provides a mathematical tool for the analysis of distributed decision making interactions between the agents with conflicting interests. On the other hand, auction-based schemes, owing to their perceived fairness and allocation efficiency, are among the well-known game theoretic-based mechanisms that can be used for task allocation. In this chapter, the game model and required economic properties of proposed reverse auction-based task allocation is discussed.

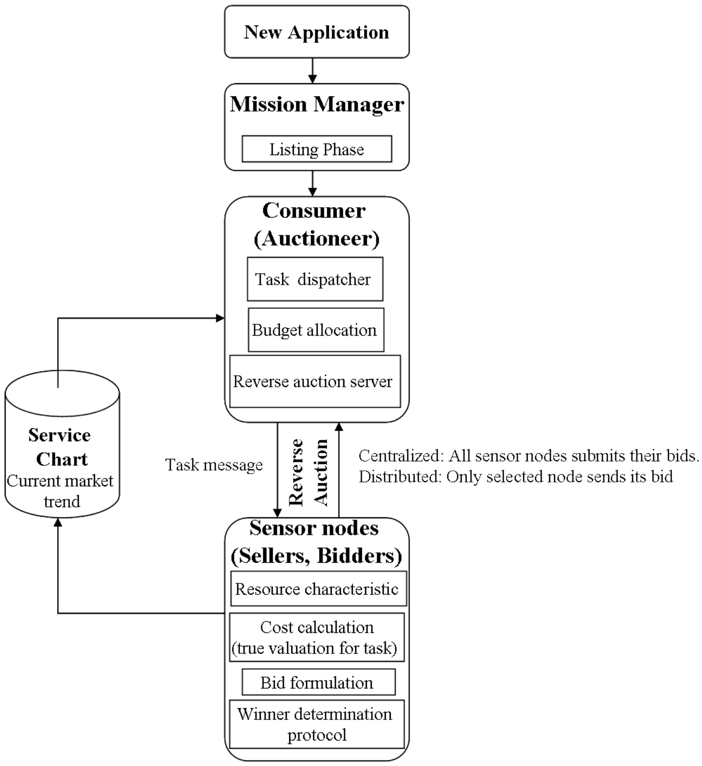


Figure 3.1: Market based architecture for task allocation

3.2 Market Architectures Components

In this work, the adaptive task allocation scheme is modeled as a competitive market. The market architecture shown in Figure 3.1, comprises of a mission manager, consumer, seller and service chart. When a new application is instantiated in the network, the input of that is fed into the Mission Manager. The components of this architecture are:

Mission manager (MM): This component assesses mission-level decisions, such as deciding the priority of the various goals for accomplishing mission objectives, and allocating these goal responsibilities to consumer agents. At the mission manager, the application level tasks are decomposed into the low-level tasks which can be executed at the sensor nodes. For example when the application is target tracking, it is required to be decomposed to low level task such as sensing, processing, sending and receiving that can be done by sensor nodes. The low-level tasks sequences and dependencies are represented by a directed acyclic graph (DAG). In a DAG graph, the vertices represent low-level tasks and the edges represent the precedence relationship between tasks. Figure 3.2 shows DAG for single target tracking application. The high-level tasks are decomposed to the low level tasks and come in the proper sequences represented by DAG. Another functionality of MM is to list the tasks in the queue based on their Earliest Start Time (EST) and Latest Start Time (LST) calculated based on list scheduling. Should concurrent tasks exist in the list, a higher priority is assigned to tasks with a larger number of successors in the task graph. The MM then allocates the various task responsibilities to the auctioneer.

Consumer (auctioneer): The consumer acts as an auctioneer. With each task arrival, the consumer communicates the task message as $< Task, TaskSize, TaskDeadline, Budget, \alpha >$ to the sellers, where *task deadline* is obtained during the listing phase in MM and *budget* is the value the auctioneer is willing to pay for the current task. The parameter α will be explained in next chapter. The auctioneer also assigns the task to the winning seller. Should there be more than one consumer, the mission manager breaks the task graph and allocates different sets of tasks to different consumers. Some of important auctioneer's functionalities are task dispatcher, budget allocator and reverse auction server. Briefly, the auctioneer's functionality are task dispatching, budget allocation and managing the reverse auction.

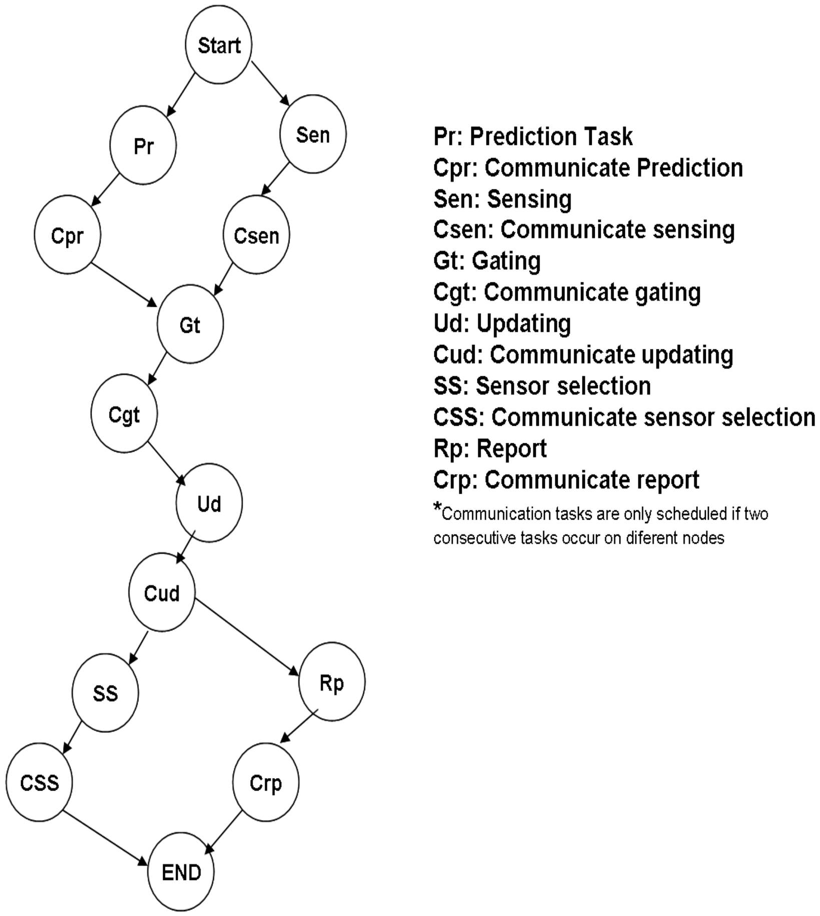


Figure 3.2: Task graph for a single target tracking application

Seller: The sellers are the sensor nodes which are modeled as the selfish and rational agents. When a task message is received from the consumer, the nodes calculate their cost for accomplishing the current task based on their current status of energy availability, communication cost, task deadline and resource release time. Eventually they set their bid based on their cost and auctioneer's budget according to the game defined on next chapter. In Distributed winner determination scheme, the seller by its functionality acts the important role in determining the winner.

Service Chart: The service chart acts as a buffer and maintains a history of the previous winning seller's bid information to assist the consumer to adjust the budget with appropriate value. The service chart specifies detailed domain information such as sensors' field locations and characteristics and the available communication bandwidth. So the bid formulator formulates bids from actual resources for the high-level tasks, using the service chart database which specifies the detailed domain information such as sensors' field locations and characteristics.

Task message format: Two formats of task message have been considered to communicate from consumer to sellers for the distributed and centralized winner determination method. The task message format for centralize method is {Task, Task Size, Task Deadline} where "Task" can be any low-level tasks such as sensing, processing, sending, receiving and etc. The "Task Size" is the expected CPU cycle required for accomplishing task and assumed to be known by consumer (auctioneer). The "Task Deadline" is the Latest Start Time calculated offline on *Listing phase* (will explain on 4.2) This format for the distributed method would change to the {Task, Task Size, Task Deadline, Budget,α}. Where "Budget" is the value that consumer willingness to pay for current task and "α" is the distribution parameter used in winner determination method (will explain on section 5.4).

Market-based architecture uses discrete time slots to schedule tasks. The flow chart of the market-based task allocation is explained as follows:

Market Initialization: When the new task arrives consumer sets the budget for accomplishing that task. It then asks for the bid from bidders (sensor nodes).

Bid Update: At the beginning of each round, bidders can

- Send new bids,
- Remove their current bids from the auction, or
- Modify the parameter of their existing bids.

Consumer asks for the bid for accomplishing task by this format {Task, Task Size, Task Deadline, Budget,α}. Then sensor nodes update their bid upon receiving this message.

Round Initialization: The auctioneer accepts new bids or updates to existing bids during each round of scheduling from the auction. If no message regarding a particular bid is received by the auctioneer, it means that non of the sensor nodes are willingness to do the task by auctioneer's budget; so that the auctioneer needs to adjust its budget.

Resource Bid Formulation: Each sensor node calculates its bid based on its available resources at each time epoch (details for bid formulation will be discussed on next chapter).

Task Allocation: In reverse auction competition, the winner would be the sensor node with the lowest bid. The task allocation runs in three phases: listing phase, task assignment phase and recovery phase in case of node failure.

Round Termination: The auctioneer updates its belief about the current market trends based on the winner's bid. The budget value needs to adjusted for the next round on "Market Initialization" part.

3.3 Reverse Auction Model

Reverse auction is a dynamic pricing method reflects the supply-demand relationship and the resources' value over time. It provides an effective and efficient solution for tasks and resources allocation. In reverse auction, the role of buyer and seller are reversed. A buyer places a request to purchase a particular item and multiple sellers bid to sell the requested item, which is similar to the procurement system [39]. The winner of a reverse auction is the seller who offers the lowest price. The reverse auction protocol applied in this work is similar to second-price sealed-bid (Vickrey auction) [18] but in a reverse manner. Vickrey auction is primarily forward auctions which involves a single seller and multiple buyers. The buyers compete among themselves in order to procure the goods of their choice. The bidder with the highest bid wins the item (willing to buy) by the price of second highest bid. Where as, in the proposed reverse auction, the seller with the lowest bid wins the competition for accomplishing the task with the price of *second lowest bid*. Hence, it is so-called *Second-Lowest-Price Sealed-Bid* or *reverse Vickrey protocol*.

In our reverse auction model figure 5.1, the consumer acts as an *auctioneer* and desires to buy some resources for accomplishing its task and without any knowledge of resources' price set by seller. Then, each bidder knows the value of its own resources and calculates the cost and bids according to its own preference to maximize its own payoff function. Winner would be the bidder with the lowest bid. The *Second-Lowest-Price Sealed-Bid reverse auction* algorithm is as follows:

- The auctioneer calculates its budget for the arriving task.

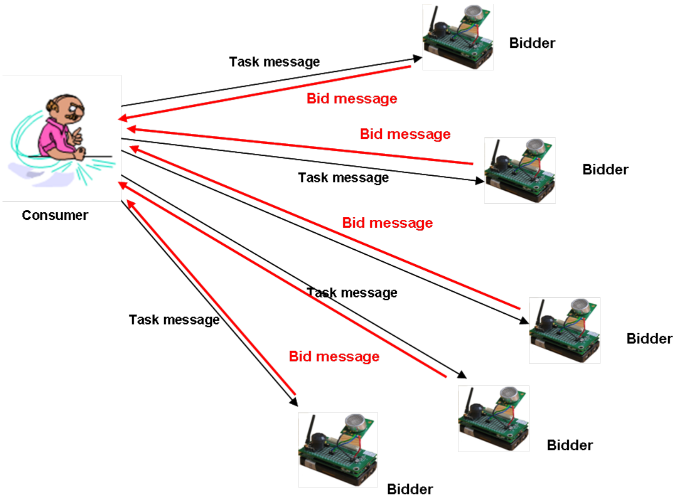


Figure 3.3: Reverse-auction based task allocation

- The auctioneer invites the public bidding.
- The seller calculates the cost based on our proposed cost formulation. (will explained in section 4.3.2)
- The seller sets the bid value. (will explained in section 4.4)
- If the bid is lower than the auctioneer's reserve valuation (budget), the seller continues the competition.
- The winner would be the seller with the lowest bid value with the price of the second lowest bid.
- If all the bids are higher than reserve price by auctioneer then it adjusts its current budget so that at least one node is selected.
- Then the auctioneer allocates the task to the winner.

The proposed reverse auction is *incentive compatible* and all of the bidders fare best when they truthfully reveal any private valuations. These game theoretic issues are discussed in detail on the following section.

3.4 Game Model of Reverse Auction

The reverse auction is an incomplete information game in the sense that the players do not have any information of other players' payoffs and their best strategies. The proposed reverse auction scheme is similar to a Vickrey auction [17] (i.e. second-price sealed-bid) but in a reverse manner. In a Vickrey auction the bidder with the highest bid wins the item it is willing to buy and pays the price of second highest bid. However, in the proposed reverse auction, the seller with the lowest bid wins the competition to execute the task with the price of the second lowest bid. Thus, we shall call this the *Second-Lowest-Price Sealed-Bid protocol.* The normal form of the game is $G =< N, \{A_i\}, \{U_i\} >$ where N is the set of players, A_i is the set of strategies for player i, and U_i is the utility function for player i. For our defined game: **Players (N)**: These are sensor nodes (sellers) that compete to get the task at their desired price.

Protocol: Second-Lowest-Price Sealed-Bid. This protocol maintains the incentive for the player to bid truthfully (refer to Lemma 1 below for more details).

Strategy: In this game, the strategy is selecting the proper bid. This bid can be a function of node's private value and its prior estimates of others' valuations.

Best strategy (A_i): The dominant strategy is to bid near one's true valuation which is the real cost for accomplishing the task.

Node's Payoff (U_i): Each sensor node i chooses its bid b_i to maximize its payoff or utility. In other words, the payoff function represents the expected profit for making bid b_i:

$$U_i(b_i; b_{-i}, C_{ij}) = \beta_i \times (R(b_i; b_{-i}) - C_{ij}). \tag{3.1}$$

where

$$\beta_i = \begin{cases} 0, & \text{if} \quad b_i \quad \text{loses} \\ 1, & \text{if} \quad b_i \quad \text{wins} \end{cases} \tag{3.2}$$

where b_{-i} refers to others' bids and $R(b_i; b_{-i})$ is the second lowest price that the player can earn as a reward if it wins. C_{ij} is the real cost for the node i for executing task j based on its available resources (refer to section 4.3.2 for details).

Nash Equilibrium (NE): The desired outcome of this reverse auction is the Nash equilibrium, which is the bidding profile b_i^* such that no player (in the domain N) wants to deviate unilaterally, i.e.,

$$U_i(b_i^*; b_{-i}^*) \geq U_i(b_i; b_{-i}^*), \forall i \in N, \forall b_i \geq 0. \tag{3.3}$$

Lemma 1. *The Nash Equilibrium (best strategy) for each player in the reverse auction game is to tell the near true valuation (real cost).*

Proof: This is a simple logical proof. All the players are modeled as selfish and rational agents and compete to maximize their payoff value. The players are aware of the fact that, in this game, the bidders with lower bids have a higher probability of winning. However, in the case of the winning with a bid lower than the task's real cost C_{ij}, it may lead to negative payoff, because, based on the second-lowest-price sealed-bid, there is a possibility that $R(b_i; b_{-i}) \leq C_{ij}$. If the bidder i bids near to its true valuation C_{ij} for executing the task j and wins the game, its payoff $U_i(b_i; b_{-i}, C_{ij}) \geq 0$, due to second-lowest-price sealed-bid protocol $R(b_i; b_{-i}) \geq C_{ij}$. This creates an incentive for the bidder to tell the true valuation. If the bidder loses, the payoff $U_i(b_i; b_{-i}, C_{ij}) = 0$. Hence, the best strategy for the bidder is to bid based on its real cost value and preference. ∎

Each bidder is unaware of other bidders' preference. Hence, without knowledge of the other players' best strategies, it would not be possible to calculate the best response that converges to the NE through the payoff function defined in equation 3.1. Another alternative for the payoff function, whose best response can be calculated in an asynchronous distributed fashion will be defined in section 4.3.2.

3.4.1 Required Economic Properties

There are three properties to ensure that auctions are economically-robust and bidder participation is encouraged [19], [20]: (1) truthfulness (incentive compatibility), (2) individual rationality and (3) ex-post budget balanced. To define them formally, we first introduce the following notations: for seller m bid B_m , the true valuation is the actual cost calculated by seller for executing task C_m, R_m is the reward that can be earned if it wins, RV is auctioneer's budget for each round of reverse auction and bidder utility is as defined in equation 3.3.

Definition 3. *A reverse auction is truthful and incentive compatible if no matter how other players bid, no seller m can improve its own utility by bidding untruthfully ($B_m \neq C_m$).*

Truthfulness is essential to resist market manipulation and ensure auction fairness and efficiency. In untruthful auctions, selfish bidders can manipulate their bids to game the system and obtain outcomes that favor themselves but hurt others. In truthful auctions, the dominant strategy for bidders is to bid truthfully, thereby eliminating the fear of market manipulation. With the true valuations, the auctioneer can allocate the task efficiently to sellers whose cost is least.

Definition 4. *A reverse auction is individual rational if no winning seller is paid less than its bid:*

$$R_m \geq B_m, \forall \;\; seller \;\; m \tag{3.4}$$

This property guarantees non-negative utilities for bidders who bid truthfully, providing them incentives to participate.

Definition 5. *A reverse auction is ex-post budget balanced if the auctioneer's profit* $\Psi \geq 0$. *The profit is defined as the difference between the auctioneer's reserve valuation (RV) and the expense paid to sellers by auctioneer:*

$$\Psi = (RV_n - R_m) \geq 0, \forall \;\; seller \;\; m \in M, \;\; task \;\; n \tag{3.5}$$

This property ensures that the auctioneer has incentives to set up the auction. In the next chapter, we show that our proposed distributed reverse auction is *economically-robust* according to the above definitions.

Chapter 4

Reverse Auction-based Task Allocation

4.1 Introduction

This chapter, mainly presents our reverse auction-based scheme for real-time task scheduling with the objective of energy balance and delay minimization. Once the new application is initiated, the application level tasks are decomposed into the low-level tasks which can be comprehensible for the sensor nodes in the listing phase (is described in detail in section 4.2. Upon each task arrival, auctioneer starts a new round of reverse auction game. Upon receiving a task message from the auctioneer, each seller (sensor node) calculates its true valuation for accomplishing that task as in 4.3.2. Then each seller bids to achieve the Nash Equilibrium as a desired output of reverse auction game, through distributed adaptive update algorithm. In the case of an unexpected situation such as node failure *during the task assignment*, the last phase (recovery phase) will dynamically run the fast recovery algorithm. The following assumptions on the wireless network are made:

- All the sensor node are one-hop away from resource requestor.
- Communications and computation can occur concurrently.
- Time synchronization is available within the cluster of sensor nodes.

To aid the reader in navigating the subsequent sections of this work, a road map showing the functionalities at the auctioneer and at the bidders (sensor nodes) is provided in Table 1.

Table 4.1: Road map of auctioneer and bidders functionalities

Auctioneer functionality	Budget adjustment: section 5.4.1, equation (5.3) Probability distribution: equation (5.2) Auctioneer utility function: equation (5.6) α adaptation: section 5.4.4 and Algorithm 2
Sensor Nodes (Bidders) functionality	Cost formulation (section 4.3.2, equation (4.6)) Probability distribution: equation (4.11) Bidder payoff function and NE calculation: equations (4.12), (4.13) and (4.15) Adaptive algorithm to adjust β: section 4.4.3 and Algorithm 1 Waiting time reduction: section 5.4.2 and equation (5.4)

4.2 Listing Phase

The listing phase computes the task sequence provided by the DAG to obtain the earliest start time (EST) and the latest start time (LST) of each task prior to starting the task assignment phase and given these values, these tasks are queued into a list. The idea of this listing phase is similar to is similar to work in [47], [52]. The Earliest Start Time $EST(v_i)$ of task v_i is first calculated for each vertex by traversing the DAG downward from the entry-tasks to the exit-task. The Latest Start Time $LST(v_i)$ of task vi is then calculated in the reverse direction. During the calculation, the entry-tasks have $EST = 0$ and the exit-task has $LST = EST$. The formulas to calculate EST and LST are as follows:

$$EST(v_i) \quad = \max\nolimits_{v_m \in pred(v_i)} EST(v_m) + t_m \tag{4.1}$$

$$LST(v_i) \quad = \min\nolimits_{v_m \in succ(v_i)} LST(v_m) - t_m \tag{4.2}$$

where $pred(v_i)$ and $succ(v_i)$ are the set of immediate predecessors and successors of v_i, respectively, and t_m is the execution time of the task on sensor nodes. Then, the Critical Nodes (CN) are pushed into the stack S in the decreasing order of their LST. Here, a CN vertex is a vertex with the same value of EST and LST. Consequently, if top(S) has un-stacked immediate predecessors, the immediate predecessor with the minimum LST is pushed into the stack; otherwise, top(S) is popped and enqueued into a queue L. The Listing Phase ends when the stack is empty. After the listing phase, the task graph is sequentialized into a queue and ready for the price-based task assignment phase.

The tasks are queued for assignment to sensor nodes based on the EST. We

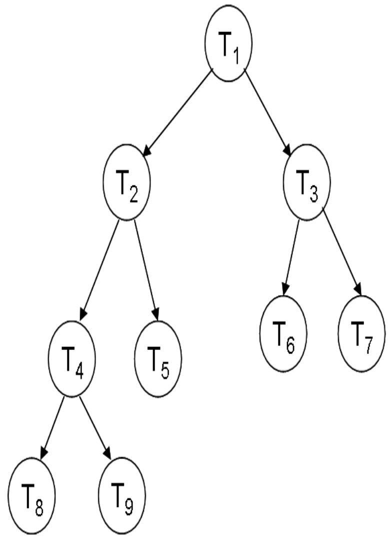


Figure 4.1: Example of DAG for assigning priority

utilize this EST value of each task for the time at which the reverse auction for assigning that task initiates by auctioneer. The LST is used as the task deadline for the assignment phase.

It should be noted that the EST and LST are for the purpose of evaluating the critical path of a DAG, and EST and LST do not represent the actual execution start time of tasks. Unlike work in [47], the EST and LST computed in this stage may be altered in the task assignment phase and dynamically due to packet loss or communication delay. Should there be tasks with concurrent EST, a higher priority is assigned to the tasks with more successors (children) in the task graph. For example in Figure 4.1 task 2 and task 3 can be done concurrently; we assign the higher priority to task 2 with totally four children rather than task 2 with two children.

4.3 Task Assignment Phase

After the listing phase, the assignment phase is performed in real-time. All the tasks that were enqueued at previous phase are consequently dequeued and the reverse auction based competition is initiated by consumer (auctioneer) to allocate the coming task to the appropriate nodes. The design objective of this task scheduling is to allocate the task in real-time and achieves a fair energy balancing among the sensor nodes based on *current availability of resources*. The pricing scheme would continuously adapt to changes of availability of resources. We also consider the fact that consumer does not have any knowledge about the energy level of each node (seller). To achieve the energy balancing objective, cost formulation has been proposed that need to be individually calculated by each distributed nodes based on its energy level and other parameters that discussed on this section.

4.3.1 Parameters for Cost Formulation

Our proposed cost formulation is parameterized by six variables; task size, energy price, base price, communication cost, task deadline and processor release time, some of which were used in [54].

- **Task Size (**S**)** refers to the expected CPU cycle required to compute or communicate the task.
- **Energy Price (**EP**)** is generated by each node per unit task based on its level

of remaining energy and is defined as:

$$EP_i = \frac{a}{1 - e^{-E_i/b}} \tag{4.3}$$

where E_i is the remaining energy of node i, a is a scaling parameters and b is a preferred coefficient which can appropriately represent the markup of energy price as the energy is consumed. This price is inversely proportional to the energy level, thus allowing nodes with higher energy (lower price) to be selected.

- **Base Price** (BP) is defined as computational cost for doing task j by node i which can be calculated as:

$$BP_{ij} = S_j \times EP_i \tag{4.4}$$

- **Communication Cost** ($CommCost$) is the cost of migrating the output of one task on one node to another task on an alternate node. It is a function of the distance between the nodes and the size of the data packet. The communication cost $c_{i,j}$ [53] for transferring data from job v_i (scheduled on p_m) to job v_j (scheduled on p_n), is defined as:

$$c_{i,j} = S + R\mu_{i,j} \tag{4.5}$$

where

S is the communication startup cost for a node (in secs),

$\mu_{i,j}$ is the amount of data transmitted from job v_i to job v_j (in bytes),

R is the communication cost per transferred byte (in sec/byte).

Our pricing scheme accounts for the communication cost when assigning tasks. It means that if $(CommCost_{jk} + BP_k) < BP_j$ then the output data of the last task at node j is migrated to node k which is assigned the current task.

- **Task Deadline (DL)** which is the latest start time (LST) defined in the listing phase. When two tasks are required to be scheduled concurrently, priority is given to the task with a closer deadline.
- **Sensor node Processor Release Time (RT)** is the time at which the already scheduled tasks at the node would finish.

4.3.2 Energy Balance Cost Formulation

In this part, the true valuation of the task execution by each of the players is formulated. One contribution of this work is a task allocation scheme that places emphasis on a fair energy balance amongst nodes constrained by the schedule length of the application. Hence, the cost formulation considers the energy availability as well as the timing factors such as task deadline and sensor node's processor release time. Should a situation arise such that a computationally expensive task arrives while only sensor nodes with low energy are available, our scheme may wait for another node with a relatively higher amount of energy to complete its task then assign this new task to that node in order to achieve a longer network lifetime.

In order to explain the timing factors used in the proposed cost formulation, an example is made:

Consider two sensor node (bidder) R_1 and R_2 where $EP_{R1} \ll EP_{R2}$ and they are

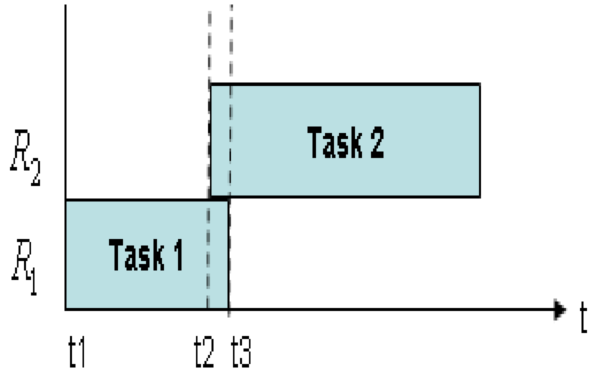


Figure 4.2: Illustration of time consideration for task scheduling

supposed to execute task 1 and task 2 (the example is shown in Figure 4.2). At t_1, task 1 is allocated to R_1 due to its lower cost value (higher energy available). At t_2, task 2 is available to be assigned. In this situation, our cost formulation may favor a fair energy balance over a longer makespan by assigning Task 2 to node R_1 at the completion of Task 1 at t_3. The cost for assigning task j to node i at time t is:

$$C_{ij}(t) = (CommCost + BP_{ij})\left[1 + \exp\left[-\frac{\lambda(t, DL_j)}{\gamma(t, RT_i)}\right]\right]^* \tag{4.6}$$

where t is the arrival time of the new task and $\lambda(t, DL_j)$ and $\gamma(t, RT_i)$ [1] are defined as follows:

$$\lambda(t, DL_j) = \begin{cases} k(DL_j - t), & \text{for } t < DL_j \quad (4.7) \\ \epsilon, & \text{for } t \geq DL_j \quad (4.8) \end{cases}$$

and

$$\gamma(t, RT_i) = \begin{cases} (RT_i - t), & \text{for } t < RT_i \quad (4.9) \\ 0, & \text{for } t \geq RT_i \quad (4.10) \end{cases}$$

where t is current time, DL_j and RT_i are task deadline and resource release time respectively.

[1] If the sensor node's processor is free when the new task arrives, the $exp(.)$ term evaluates to $-\infty$ which results in 1 for the entire term in the square bracket(*).

A fair energy balance amongst nodes can be achieved with this bid formulation. Regarding to $\gamma(t, RT_i)$ value, When the current time t is close to the release time of a processor RT_i with high energy availability, a low value would be set, increasing the selection possibility by the auctioneer. It can also be shown in equation 4.10 that if at current time the resource (sensor node) is free so the time consideration part from equation 4.6 would be omitted and the bid is only the function of energy considerations.

Regarding to $\lambda(t, DL_j)$ the deadline of this task DL_j to be assigned is also considered (the task deadline is the latest start time (LST) of the task). The tasks with urgent deadlines would be allocated to a node that is available at a closer released time (or a node that readily available) at the expense of an unfair energy balance.

Lemma 2. *Let the time consideration part of cost formulation on Equation 4.6 to be* $\zeta = \left[1 + exp\left[-\frac{\lambda(t,DL_j)}{\gamma(t,RT_i)}\right]\right]$. *Then* $\forall\ DL,\ RT,\ t > 0\ ,\ \exists\ 1 \leq \zeta < 2$.

Proof: For **upper bound** if $t \geq RT_i$ from equation 4.10 then $\gamma(t, RT_i) = 0$ and $\lambda(t, DL_j) > 0$, so $\zeta = \left[1 + exp\left[-\frac{\lambda(t,DL_j)}{0}\right]\right] = 1$. For **lower bound** if $t \geq DL_j$ from equation 4.8, $\lambda(t, DL_j) = \epsilon$ and $\gamma(t, RT_i) \geq 0$ so $\zeta = \left[1 + exp\left[-\frac{\epsilon}{\gamma(t,RT_i)}\right]\right] \simeq 2$. ∎

Figure 4.3 shows an example of cost value from one specific node for accomplishing one task over time where $DL > RT$ ($DL = 80$ and $RT = 50$). As can be seen, the cost value before the resource release time is decreased over time, so that the probability of being selected on the competition increases. Figure 4.4 shows the cost variation over the time where $DL < RT$ ($DL = 50$ and $RT = 80$). As can be seen, the cost value before the task's deadline increases because the resource at that time is busy

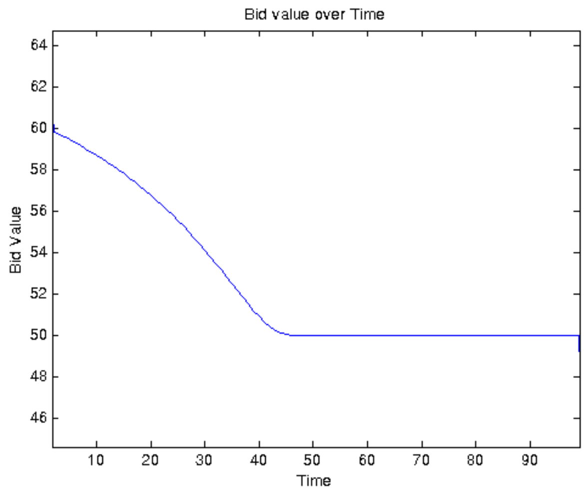


Figure 4.3: Cost value over Time when $DL > RT$ ($DL = 80$ and $RT = 50$)

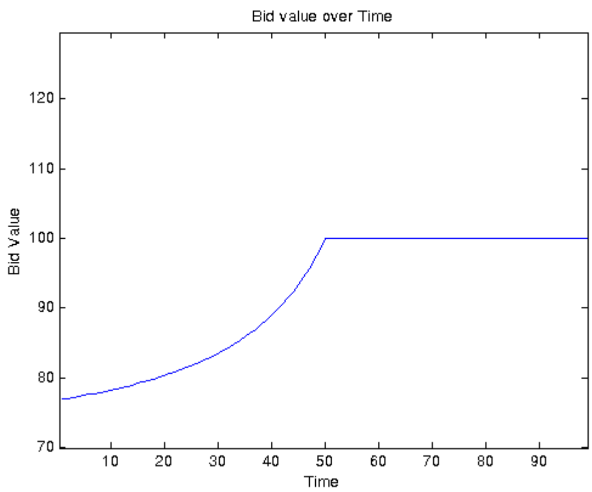


Figure 4.4: Cost value over Time when $RT > DL$ ($DL = 50$ and $RT = 80$)

($RT = 80$) and we have an urgent deadline so the cost value would be high to reduce the probability of being selected on the competition.

4.4 Bidding to Achieve NE in a Distributed Fashion

In this game, the players are modeled as selfish and rational agents. Hence, after they calculate their cost for executing the task, they may want to bid the price that can maximize their payoff. For the incomplete information reverse auction game, the players do not reveal their computational strategies. Here, each proposal and bid response provide some information about future predictions and decisions. They act as signals that help the bidders and auctioneer to update their beliefs about what the other has computed.

4.4.1 Attributes of Auctioneer and Bidders

The real-life parameters affecting the game are modeled by the following attributes:

Bidder's Cost (C_{ij}) : This is the true value for executing the task j by node i. This value has energy and time considerations and has been defined in section 4.3.2.

Reserve Valuation (RV): This is the maximum possible budget the auctioneer is willing to pay to get the task done. The rationality behind this definition is to eliminate the number of competitors whose cost is higher than the budget. The auctioneer's global objective is to achieve the energy balance and delay reduction in task scheduling and assignment. Hence for each reverse auction interval, it adjusts the budget to the value that maximizes the auctioneer's expected surplus.

Market Price (MP): This value is calculated based on the statistics maintained by the bidder and auctioneer maintain a history keeping track of recent reverse auction games. This history will help to determine the "market value" of the resources the sensor network application is seeking. Hence, it depends on the kind of tasks that was distributed by the auctioneer before and executed by sensor nodes.

Bid Value: This is basically the price offered by the bidder which can maximize the bidder's utility function. The standard price corresponding to the highest utility gives the offered bid price.

Probability Distribution $f(bid; RV, C_{ij})$: This is the probability distribution of the bidder's belief about the auctioneer's preference of the winning bid. It has the form of a *truncated, decreasing geometric distribution.* The effect is that the lower offered price has a much higher probability of winning. In other words the perceived probability decreases monotonically with the bid value.

$$f(bid; RV, C_{ij}) = \begin{cases} \frac{(1-\beta)\times\beta^{(bid-C_{ij})}}{1-\beta^{(RV+1)}}, & C_{ij} \leq bid < RV \\ 0, & otherwise \end{cases} \tag{4.11}$$

where β is a distribution parameter. Considering a bidder i with the cost for accomplishing task j, $C_{ij} = 40$ and $RV = 100$, the probability distribution with different β values $0 < \beta < 1$ is shown in Figure 4.5.

4.4.2 Bidder's Payoff Function

In order to achieve the best bid value in the incomplete information game, each player has to calculate its payoff without knowledge of the other players' preference and bid values. Hence, the payoff function is designed so that it can be calculated in

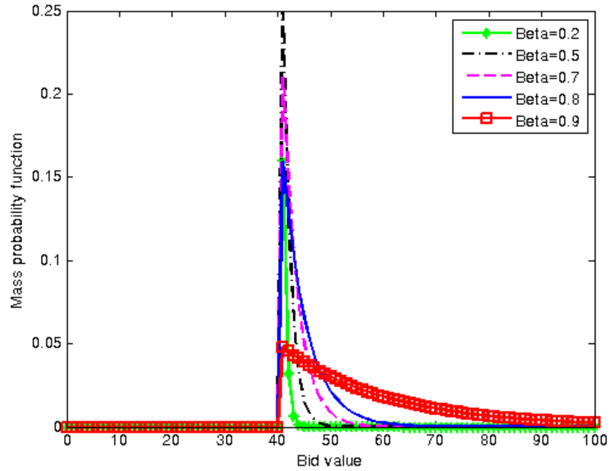


Figure 4.5: Probability distribution for different β.

a distributed fashion based on the local information at each player and their belief about the auctioneer's preference. The concave and non-smooth payoff function is defined as follows:

$$U_i = (bid - C_{ij}) \times f(bid; RV, C_{ij}). \tag{4.12}$$

Hence, the bidder's best response (Nash Equilibrium) is as follows:

$$B_i(bid, RV) = \{bid^* | bid^* = \arg\max U_i(\widetilde{bid}; RV)\}. \tag{4.13}$$

Theorem 1. *There exists upper and lower bounds such that a unique NE exists if the bid is selected from the following range: $C_{ij} < bid < RV$; otherwise, no NE exists.*

Proof: From equation 4.11 and 4.12, the bidder's payoff can be written as follows:

$$U_i = \begin{cases} \frac{(1-\beta)\times\beta^{(bid-C_{ij})}}{1-\beta^{(RV+1)}}, & C_{ij} \leq bid < RV \\ 0, & otherwise \end{cases} \tag{4.14}$$

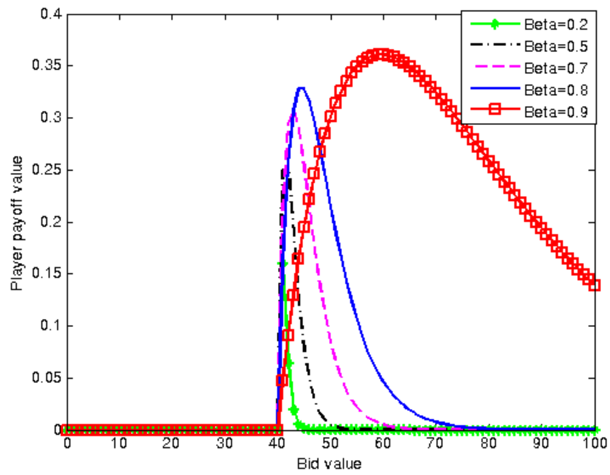


Figure 4.6: Best response for the defined bidder's payoff function.

From this equation, the unique NE is not be a continuous function of the bid due to the discontinuity of the player's payoff function. In particular, the unique NE could be all zero for any bid outside of the upper and lower bounds. Consider a bidder i with the cost for accomplishing task j, $C_{ij} = 40$ and $RV = 100$. The payoff and the best response for different β are demonstrated in Figure 4.6. ∎

Theorem 2. *Considering the payoff function defined in equation (4.12) where* $0 < \beta < 1$ *, the NE is a value near to the bid's lower bound (cost value).*

Proof: We need to show that $bid^* = C_{ij} + \zeta$ where ζ is a small positive value. The payoff function defined in equation (4.12) for $C_{ij} < bid < RV$ is

$$U_i = (bid - C_{ij}) \times \frac{(1-\beta) \times \beta^{(bid - C_{ij})}}{1 - \beta^{(RV+1)}} \tag{4.15}$$

which for the interval $C_{ij} < bid < RV$ is continuous and differentiable, so the non-negative root of the derivative of the payoff function is $bid^* = \frac{-1}{\ln \beta} + C_{ij}$. Since we assume $0 < \beta < 1$, then $\ln \beta$ is negative and $\zeta = \frac{-1}{\ln \beta}$ is a small positive value. ∎

Lemma 3. *The proposed distributed reverse auction achieves truthfulness (incentive compatibility) and individual rationality.*

Proof: The proof for truthfulness is provided in the proof for theorem 2. Since the proposed reverse auction is a second-lowest-price sealed-bid, for the winner with the truthful bid the reward is higher than the value that it bids ($R_m \geq B_m$), so it is individual rational. This will guarantee non-negative utilities for bidders who bid truthfully. ∎

4.4.3 Asynchronous Best Response Updates of Bids

In this section, we want to address how each bidder can adaptively modify the distribution parameter β to achieve the best response value that can maximize the bidder's payoff in a *distributed fashion*. To achieve this, we allow the players to iteratively update their bids based on the best response functions results from the β in an asynchronous fashion. A *distributed asynchronous update algorithm* (Algorithm 1) for adjusting β is proposed. The rationale behind this algorithm is simply as follows:

The distribution parameter β is initialized to some random value β_0. There are two conditions for modifying this value. First, if the bidder loses the competition, it means that the proposed bid was too high so that it can not be selected as a best (lowest) bid value. In this situation, the node's belief about auctioneer preference has to be modified by adjusting the β value for next reverse auction round. Secondly, on the opposite side, if the bidder wins the competition for several times continuously, it means that there is still a probability of winning even if a higher bid is offered.

Algorithm 1 Asynchronous algorithm to adjust β

Initialize: $\beta = \beta_0$, $Step = \theta$, $k = 0$, $t = 0$, $k' = \eta$
  for each iteration of reverse auction game **do**
    $t = t + 1$
    if $Step > \epsilon$ **then**
      if $U_t \geq U_{t-1}$ **then**
        $k = k + 1$
        if $k \geq k'$ **then**
          $\beta_{t+1} = \beta_t + step$
          $k = 0$
        else
          $\beta_{t+1} = \beta_t$
        end if
      else if $U_t < U_{t-1}$ **then**
        $\beta_{t+1} = \beta_t - step$
      end if
      Decrease $Step$: if the number of β values which are near the mean value of β is high and also if β moves toward the mean value
    end if
  end for

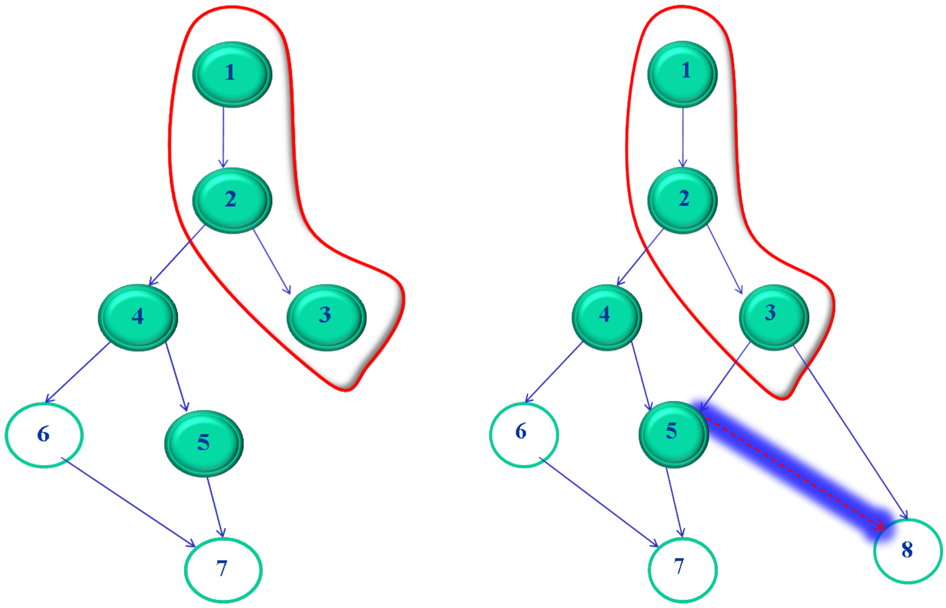
Again, in this situation, the node's belief about the auctioneer's preference needs to be modified by adapting the β value. The probability distribution of the node's belief about the auctioneer's preference and the player's payoff value with different β have been shown in Figure 4.5 and 4.6. The asynchronous algorithm for updating distribution parameter is shown in algorithm 1. Given a distributed pool of bids from bidders (sensor nodes), a centralized Winner Determination Protocol (WDP) would require costly message exchanges with high overheads. We now propose a distributed WDP in the next chapter.

4.5 Recovery Phase

In WSNs, sensors are prone to failures. In case of sensor failures, the current application executing instance may stop due to task dependencies defined in the task graph, i.e. the data carried by failure sensor node may be demanded as the input for another task execution. In such cases, the lost data of failed node needs to be promptly recovered for the subsequent application executing instances. Instead of rescheduling from scratch, which can be time consuming, low-complexity recovery algorithms are preferred.

The proposed method enables recovery from node failure *during the online task assignment phase*. The previous works, such as [8, 47] only considered node failure *prior to task assignment* by not selecting the node that had failed. In the recovery phase, firstly, the tasks that had been assigned to the failed node are recovered from its successors deployed on other nodes. In other words, it is determined if there are any tasks that run on the failed node that they need to be assigned again to another nodes. If these tasks have no successors or no undeployed successors, then redeployment is unnecessary. Redeployment is also unrequired if the output of this task exists on another node. This situation occurs when the data as the task output had been previously communicated to another node as the task's successor. Figure 4.5 shows an example of node failure where redeployment of tasks done by failed node is unnecessary. Task 1, 2 and 3 have been done by failed node and all the tasks with filled color have been already deployed. In case "1" redeployment is unnecessary since the tasks done by failed node (task 1, 2, 3) don't have any undeployed successors.

In case “2” redeployment is also unnecessary since output of task exists on another nodes.



(a) Case 1 (b) Case 2

Figure 4.7: Unnecessary cases for redeployment of failure node’s tasks

However, if there exists no back-up copy of the output of these tasks then re-assignment of these tasks onto another sensor nodes is required. If the deadline where this output is valid has been exceeded, redeployment of this task would not be performed. Task assignment resumes as normal with the rest of the tasks after the recovery phase. The result of implementing a recovery phase which checks for previously communicated data from tasks on a failed node to avoid unnecessary task deployment is the savings in energy and time.

Although the recovery method proposed results in a slight increase in schedule length, simulation results shows the significant improvement in schedule length in comparison to rescheduling considered for static scheduling in addition to energy consumption and balance improvements.

Chapter 5

Winner Determination Protocols for Reverse Auction-Based Task Allocation

5.1 Introduction

So far, we have described how to bid to satisfy the energy balance objective for the auction-based distributed task allocation. However, given a distributed pool of bids from bidders in our auction game, a centralized Winner Determination Protocol (WDP) would require costly message exchanges with high overheads. This fact also needs to be considered that for sensor nodes the energy consumption for transmitting and receiving are significantly higher than other tasks such as sensing and processing [38]. Hence, a novel Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP) for the reverse auction-based task allocation scheme is proposed. This protocol has a contention-based MAC (Medium Access Control) protocol interpretation, where all the bidders compete during the contention window and only the winner of the game (the bidder with the lowest bid) will transmit its packet to the auctioneer. The main objectives of this protocol are to reduce the overhead, delay and energy consumption in message exchanges while determining the winning bidder.

In this chapter, the centralized and distributed WDP are introduced and compared with the proposed Delay Efficient Distributed Winner Determination Protocol (ED-WDP).

5.2 Centralized Winner Determination Protocol (C-WDP)

When a new task arrives, the auctioneer initiates the reverse auction round. It broadcasts a message containing $\langle$TASK, TaskSize, TaskDeadline$\rangle$ to all the sensor nodes (bidders). Each bidder calculates its bid, and sends that bid to the auctioneer. The auctioneer then selects the bidder with minimum bid among all the bidders. Then the task would be allocated to that winner bidder. The main disadvantages of this protocol are the overhead and the cost (energy consumption) for sending all the bids to the auctioneer specially in the network with a large number of distributed nodes. Moreover, in this method, the probability that collisions occur during message exchange is also high. Lastly, the auctioneer needs to wait for all the bidders to send their bid and then makes the decision. Given these disadvantages, for any auction-based scheme, it is more efficient to utilized the distributed decision making and message exchange.

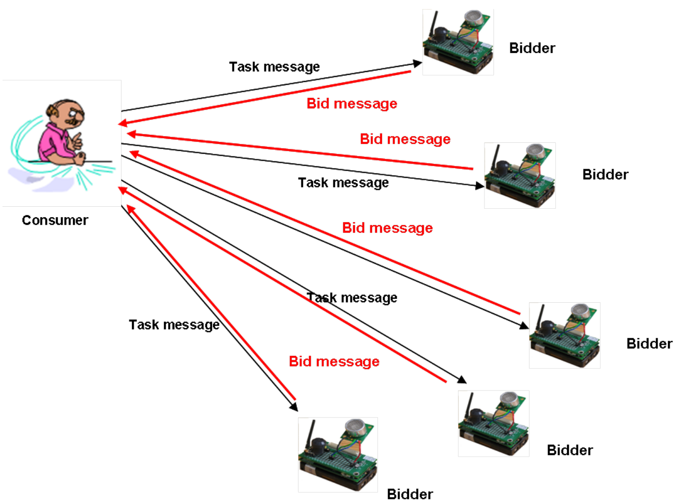


Figure 5.1: Centralized winner determination protocol

5.3 Distributed Winner Determination Protocol (D-WDP)

To reduce the communication overhead and energy consumption for the message exchanging on C-WDP, the distributed WDP is introduced. Each seller upon receiving auctioneer's message, calculates its cost and bid then instead of communicating its bid for accomplishing the current task immediately, each node sets a waiting time T_w proportional to its bid B (linearly maps the bid to the waiting time) and goes to a LISTEN mode. Mathematically, for current task j, the node i calculate its waiting time $T_w(i, j)$ by

$$T_w(i, j) = \ell \times B_{ij} \tag{5.1}$$

Where ℓ is a linear coefficient. When the waiting time is completed, the bidder would then send the message to auctioneer. It means that the bidder with the lowest bid

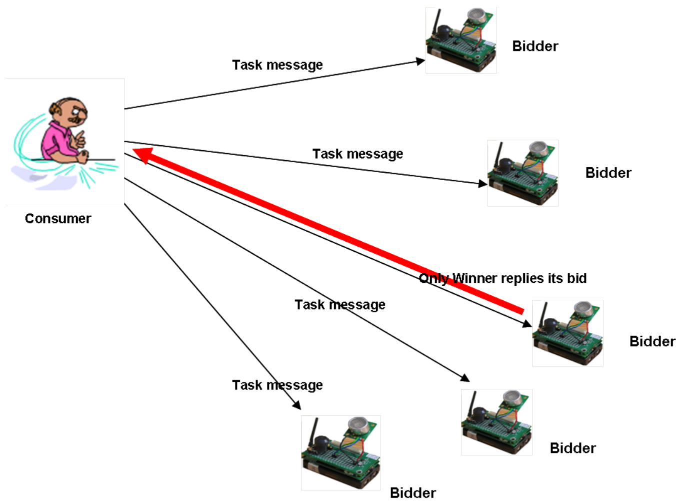


Figure 5.2: General distributed winner determination protocol.

(the winner) will send its bid first and be selected. Upon reception of a message from a winning bidder node, the remaining nodes (which are in a LISTEN mode) would leave the competition and avoid communicating their bids. Hence, the winning bidder is the only one who send the message to auctioneer. This scheme will considerably reduce the amount of *overhead* and *energy consumption* for sending non-winning messages to the auctioneer; however, it may result in the *latency* in responding to the auctioneer, caused by the waiting time.

In General Distributed scheme, two more issues are not considered. The first one is that the number of active competitors in the LISTEN mode with low energy ought to be reduced, as some of these nodes with low energy level may not win the competition. The second issue is how to reduce the waiting time for the response of the winner, which will also decrease the time that the other nodes spend in the LISTEN mode.

5.4 Energy and Delay Efficient Distributed Winner Determination Protocol

In order to determine the winner among distributed bidders in the reverse auction-based task allocation and also achieve the *auctioneer's objectives* which are *energy conservation* and *delay reduction* in message exchange, the novel Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP) is proposed. Upon each task arrival, a new reverse auction round is initiated. The sensor nodes (bidders) compete over the predefined Contention Window (CW) time. This protocol has a contention-based MAC (Medium Access Control) protocol interpretation, where all the bidders compete during the contention window and only the winner of the game (the bidder with lowest bid) will transmit its packet to auctioneer. The main objective of this protocol is to reduce the overhead, delay and energy consumption in message exchanges for determining the winning bidder. This protocol is partitioned into two phases.

5.4.1 Phase 1 - Elimination via Budget Value

One of the largest sources of energy consumption in wireless sensor nodes is the use of *idle listening*. Hence, reducing the number of sensor nodes in this mode is essential. In our proposed ED-WDP, the budget value is the Reserve Valuation (RV) set by the auctioneer and communicated to the bidders. This budget is the auctioneer's willingness to pay for the current task in the reverse auction. Upon receiving the task message which contains the RV value, each seller calculates its bid and compares it with RV set by the auctioneer. Bidders whose bids are higher than the budget will leave the reverse auction competition at the beginning and go in to the SLEEP mode. As a result, sellers that have lower amounts of energy and thus, higher bids will not be selected for the rest of the competition, as the auctioneer would prefer to pay a lower price for the current task. The budget (RV) is adapted to ensure that at least one node is selected via this phase. The issue which then arises is how to set the budget. If the auctioneer sets a high value as the budget for the arriving task, a large number of bidders may be selected to continue the game. This issue causes the higher total energy consumption for the node staying in the idle listening mode. On the other hand, by setting the low budget value which is below the market trend, no bidders may be chosen. If a multi-round reverse auction is used, the budget value will be adjusted in the following rounds.

The *truncated increasing geometric distribution* is used as the market trend distribution in auctioneer's belief. Such a probability distribution has been used in [36] for picking a transmission slot in the contention window-based MAC protocol (so-called SIFT) to reduce the response latency. This probability distribution allows the price near to the auctioneer's RV an exponentially higher probability of winning. The probability distribution of market price is as follows:

$$g(bid; \alpha, RV) = \frac{(1-\alpha)\alpha^{RV}}{1-\alpha^{RV}} \times \alpha^{-bid} \tag{5.2}$$

where the random variable $bid \in [1, RV]$ and α is the distribution parameter where $(0 < \alpha < 1)$. Figure 5.3 shows the probability distribution for market price given the

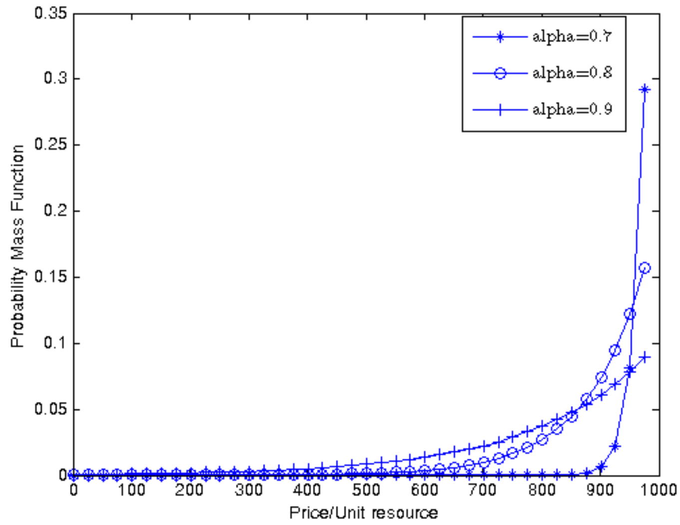


Figure 5.3: Probability distribution of market price for different α.

different values of α and the budget set to be 1000 and all the bids are [1,1000].

The auctioneer adjusts the budget based on a feedback received from the cost of the winner from previous iterations and the probability of previous winner's cost as follows:

$$RV_{t+1} = MP_t \times (1 + \hat{g}_t) \tag{5.3}$$

where $\hat{g}_t$ called the *safeguard factor* is the probability distribution value of the market price in the t^{th} iteration and the MP_t is the average of bids of the previous winners in κ prior rounds i.e. $Average(bid_t, bid_{t-1}, ..., bid_{t-\kappa+1})$. The initial budget is set to a large random value so that a large number of nodes will be selected; the budget value then adaptively changes based on the feedback on the cost of winners.

One of the economic properties that ensures that the auction is economic-robust and encourage auctioneer's participation is ex-post budget balancing. According to definition 5 a reverse auction is *ex-post budget balanced* if the auctioneer's payoff $\Psi \geq 0$. The payoff is defined as the difference between the reserve valuation (RV) and the expense paid to sellers R_m (second lowest bid value).

Lemma 4. *The proposed distributed reverse auction achieves ex-post budget balance i.e.* $\Psi = \sum_1^n \sum_1^m (RV_n - R_m) \geq 0, \forall \ \ seller \ \ m, \ \ task \ \ n$.

Proof: RV is the budget value that is set by the auctioneer and as can be seen from equation (5.3), due to the safeguard factor, it is always higher than the current market trend. In other worlds, the current market trend is the average of the rewards achieved by the bidder from the auctioneer. Hence, we can guarantee that $RV_n \geq R_m$.

■

This property ensures that the auctioneer has incentives to set up the reverse auction. The rationality behind the budget utilization in our reverse auction is to achieve the auctioneer's objective on overall energy conservation and energy balancing during winner determination. As the result the sensor node with the lower amount of energy quits the game initially before the winner determination protocol runs.

The initial budget is set at a large random value as that a large number of nodes will be selected; the budget value then adaptively changes given feedback on the winning node's bid. So far, the constant distribution parameter α has been assumed. However, the auctioneer needs to change its belief about the current market trend by adjusting different α values in probability distribution (Figure 5.3) after some iterations. In section 5.4.3, it will be shown that the different α values returns different budget adjustments and the distribution parameter converges to the near optimal value via an adaptive algorithm.

5.4.2 Phase 2 - Waiting Time Reduction

This phase is run in the sensor nodes (bidders) and its main objective is the delay reduction during the winner determination. After the elimination phase, each of the surviving sensor nodes for the competition individually sets the waiting time based on its own bid value.

When the waiting time is completed, the winning node is the first and only node to send its message and the remaining nodes (who are in LISTEN mode) receive the winner's message and understand that they have lost in that round of the game. This scheme will reduce the amount of overhead and energy consumption for sending the non-winning messages to the auctioneer. However, it may result in latency in responding to the auctioneer, due to the waiting time. The main purpose of this phase is to reduce the delay or waiting time before the first bidder (which will be the winner of the reverse auction) replies with its bid. In order to reduce the response time of the bidders, a non-uniform function is used to map the bid value to the waiting time. By reducing the waiting time, not only is the delay decreased but the energy consumption of all the active bidders is decreased by reducing the time these bidders remain in the LISTEN mode. As mentioned before, the similar idea has been used in [36] for picking a transmission slot in the contention window-based MAC protocol (so-called SIFT) to reduce the response latency.

An example shown in Figure 5.4 demonstrates the differences of the waiting time

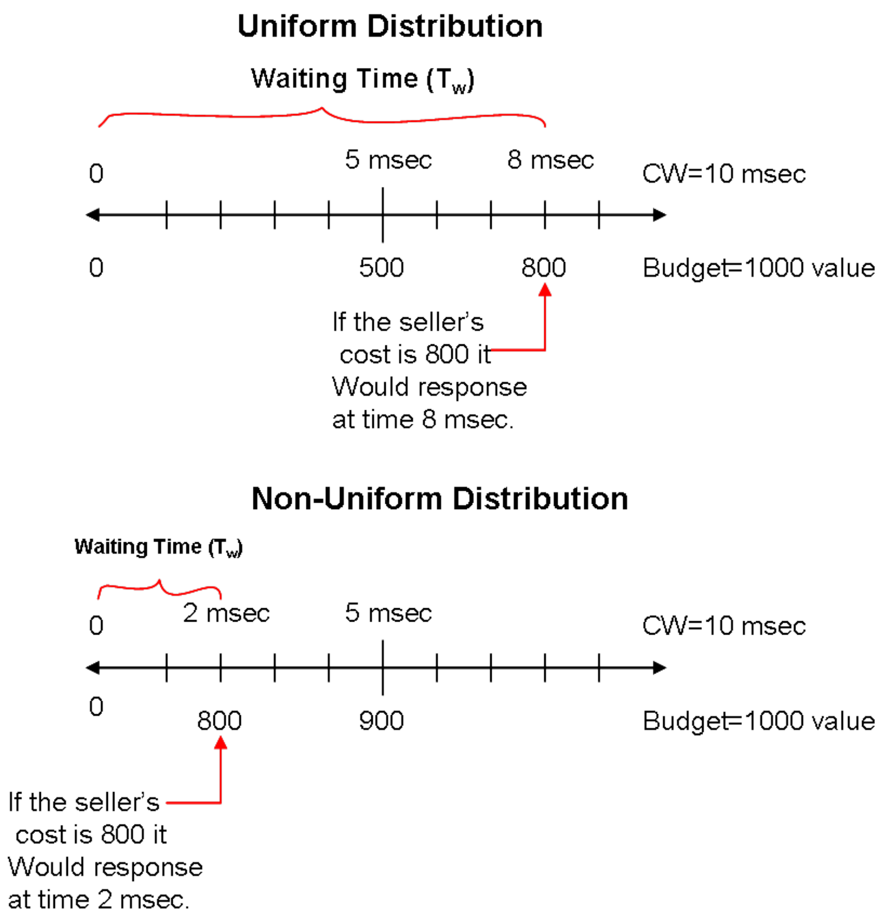


Figure 5.4: Example of the uniform and non-uniform bids and waiting times mapping.

cause by mapping the high bid value (comparable to budget value) to the uniform and non-uniform distribution. In the first case, the contention time window is mapped uniformly such that given a budget of 1000 per unit task and a contention time window of 10 msec (or, if the seller's cost is 800 per unit task, the waiting time is 8 msec). However, utilizing non-uniform distribution for mapping the same cost of 800 per unit task, the sellers response is reduced to 2 msec. It significantly reduces the latency before the bidder reveals its bid.

The bidder receives RV and α value from the auctioneer task message and calculates its bid. If the bidder's bid exceeds the budget, that bidder will leave the competition and go to SLEEP mode. Given the distribution probability $g(bid; \alpha, RV)$ defined in equation (5.2), the bidder i maps its bid value to the waiting time non-uniformly as follows:

$$Tw_i(\alpha) = \frac{bid_i \times g(bid_i; \alpha, RV)}{RV} \times CW \tag{5.4}$$

where CW is the fixed contention window that the bidders compete over. The calculated waiting time would increase the resolution for the time intervals where the most of current bids (market price) are. At the end of the contention window, all the nodes revert to ACTIVE mode in order to be ready to bid for the new arriving task. Figure 5.5 shows the different waiting time T_w over the costs for different α value from equation 5.2 given a CW of 100 msec and a budget of 1000 per unit resource. The waiting time is significantly reduced in comparison to a uniform mapping of the bid to the waiting time.

In conclusion, the reason for utilizing the increasing geometric probability distribution as the auctioneer's belief of the market price is to achieve the followings' goals:

- To adjust the budget value: Auctioneer needs to have an idea of the current market trend to adjust its budget value from equation 5.3. If the current market trend is near to the budget value, it will be increased for the next iteration in order to increase the probability that the acceptable budget value is maintained where at least one bidder will be selected.
- To map the bidder's bid to the waiting time: In order to reduce the time the

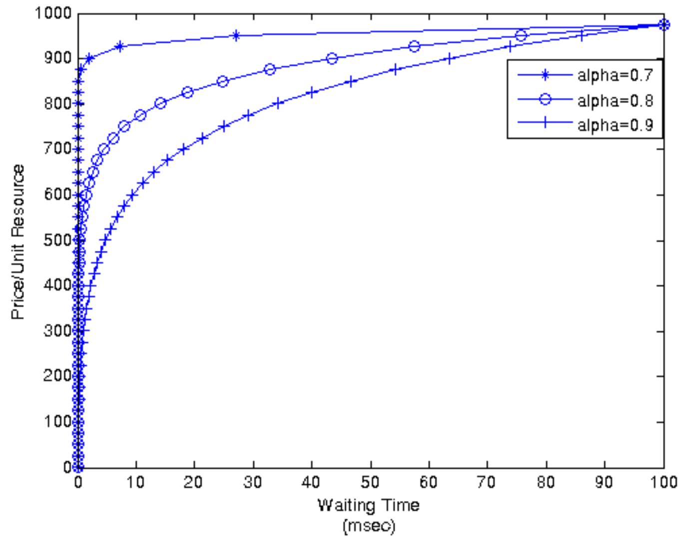


Figure 5.5: Bids vs waiting times for different α

sellers (sensor nodes) are in LISTEN mode, where no nodes transmit their bids, and also to increase the resolution for the time intervals that most of current bids are.

5.4.3 Comparison of Different Distribution Parameter

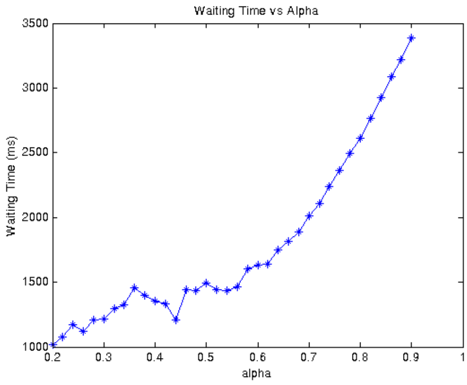
The auctioneer needs to correct its belief about the current market trend by adapting the distribution parameter α. The auctioneer faces the dilemma of setting the best α value, since it influences the budget (RV) and waiting time. A higher α value results in a longer waiting time (the time before the winner responds) but a lower negotiation time for adjusting the budget value. The negotiation time is the contention window multiplies by the negotiation round. The negotiation round is required when all of the bidders are eliminated on phase 1 with the calculated budget, so the budget needs to be adjusted again. A lower α value has the opposite effect. Hence, the total delay upon receiving a winning bid in our decision making mechanism is:

$$T_{total}(\alpha) = T_w(\alpha) + CW * (N_{round}(\alpha) - 1) \tag{5.5}$$

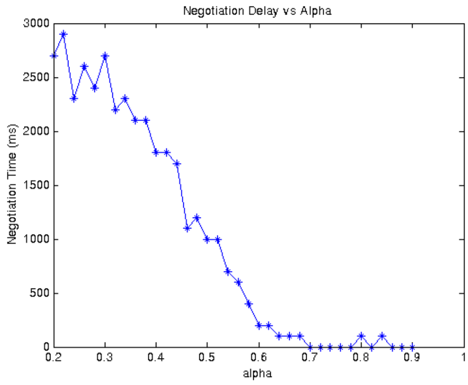
where N_{round} is the number of rounds of negotiation which is a function of α, CW is the contention window and T_w is the waiting time (winner's response latency) mentioned above (equation 5.4). At this stage, our objective is to adapt the α value iteratively to achieve the α which results in lower total delay T_{total}. Before that, we need to compute the optimal value for α that the adaptive algorithm is supposed to converge to it. Simulations are performed for different α values (from 0.2 to 0.9) with $CW = 100$ msec for 40 iterations at each α. Figures 5.6(a) and 5.6(b) show the waiting time T_w and negotiation time at different α values. Figure 5.6(c) shows the total delay T_{total} at different α values. The optimal α value which minimizes the total delay is thus obtained between α=0.6 and α=0.7.

5.4.4 Distributed Iterative Best Response Update

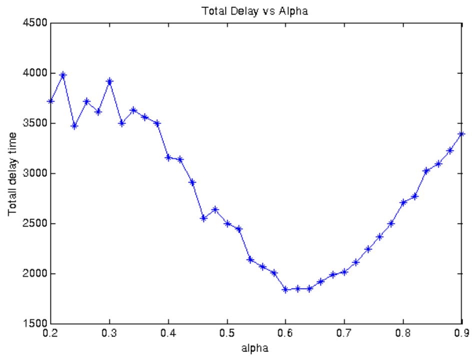
The α value is adapted by the auctioneer to converge to the optimal value, i.e. the minimum value shown in Figure 5.6(c). This α value is communicated to the sellers via the task message on each task arrival. By utilizing an adaptive algorithm, α can be adjusted in an asynchronous manner according to the history of its previous values. It would approach the optimal α which will returns the best budget value and waiting time in the current market environment. This alpha value is communicated to the sellers with the task message {Task, Task Size, Task Deadline, Budget, α} with each task arrival.



(a) Waiting Time



(b) Negotiation Time



(c) Total Delay

Figure 5.6: Waiting Time, Negotiation Time and Total Delay for α=[0.2 0.9].

Definition 6. *A* consistent distribution parameter history *of an auctioneer, denoted as* ‘CDPH’, *is the history in which all the ρ auction rounds shares the same α as the distribution parameter.*

Definition 7. *In order to achieve the auctioneer's objectives which are the delay reduction and energy conservation during message exchange the auctioneer's expected utility function for each task (i) arrival from the task domain $\mathbf{T}$ is set as follows:*

$$U_{ex}(\alpha_i) = -\sum_{j=1}^{\rho}(T_w^j(\alpha_i) + CW \times (N_{round}^j(\alpha_i) - 1)) \quad \forall i \in \mathbf{T} \tag{5.6}$$

where T_w and N_{round} are the function of α value and ρ is number of iteration required for observing α_i performance in $CDPH$.

Definition 8. *The best response for iteration i (task arrival i) is to achieve α that can maximize this utility function as:*

$$\chi(\alpha_i) = \{\alpha_i | \alpha_i = \arg \max_{0<\widetilde{\alpha_i}<1} (U_{ex}(\widetilde{\alpha_i}))\} \tag{5.7}$$

The general idea is that once a new utility history with consistent α after ρ rounds is formed, the probability distribution used by the auctioneer is increased or decreased according to the auctioneer's 1st and 2nd most recent utility histories with consistent α. By utilizing an adaptive algorithm, α can be adjusted in an asynchronous manner according to the history of its previous values. It would approach the optimal α which will returns the best budget value and waiting time in the current market environment.

The next alpha value may be increased or decreased based on the two most recent consistent α histories. To obtain the ith α history, the α value is evaluated based on the expected utility function of auctioneer on Definition 7.

If there is an improvement in the expected utility U compared to the utility in the previous CDPH period, U', the positive adjustment parameter $\mu = 1$ is set for the next iteration. In other words, α is further adapted is the same direction. If the utility function returns a worse result, the previous adjustment was improper and the adjustment parameter is set to $\mu = -1$ and α is further adapted in the opposite direction.

Algorithm 2 shows the adaptive strategy used. At this stage, α_0, *step*, μ and U'

are initialized. The α is adjusted when a new $CDPH$ is formed. For adjusting the α, the auctioneer computes the expected utility of recently formed $CDPH$ and records this new value in U. Then it modifies the α according to the value of μ and the relationship between U and U' (line 10-14). If the previous adjustment of α, which is illustrated by μ leads to a decrease of utility, the adjustment parameter will be change to the opposite value which means that the previous and next adjustments of the α value are different(line 15-19). In this reverse auction the value of α is changed after each $CDPH$ formed and will slowly converge due to *decreasing step*. The step is decreased (line 7) under some conditions: first if the number of α values which are near to the mean value of α are high; second if the α moves toward the mean value. The adaptation process is finished when step is smaller than ϵ and the auctioneer used this α for the rest of subsequence rounds.

Algorithm 2 α adaptation algorithm

Initialize: $\alpha = \alpha_0$, $step = \theta$, $\mu = 1$, $U' = T_0$

  while reverse auction is not finished **do**

    if a new consistent distribution probability history (CDPH) is formed and $step > \epsilon$ **then**

      Compute $U_{ex}(\alpha)$;

      $U = U_{ex}(\alpha)$ and $\alpha' = \alpha$;

      Decrease $step$: if the number of α values which are near to the mean value of α are high and also if α moves towards the mean value.

      if $U < U'$ **then**

        $\alpha = \alpha - \mu \times step$

      else if $U \geq U'$ **then**

        $\alpha = \alpha + \mu \times step$

      end if

      if $\alpha \geq \alpha'$ **then**

        $\mu = 1$

      else if $\alpha < \alpha'$ **then**

        $\mu = -1$

      end if

      $U' = U$

    end if

  end while

Chapter 6

Simulations

6.1 Simulation Setup and Parameters

In this chapter, we describe the simulation study performed to evaluate the performance of our distributed reverse auction-based task allocation schemes and algorithms as follows:

- The convergence of the proposed distributed iterative best response algorithm for the bid updates.
- The performance of our task allocation scheme in terms of energy balancing and energy utilization.
- The effect of our proposed task allocation scheme in terms of the schedule length and energy consumption over a different number of available nodes.
- The performance of our proposed fast recovery scheme for the node failure during the real-time task assignment.
- The performance evaluation of the Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP).
- The convergence of algorithm 2 (section 5.4.4).

The algorithms were implemented using Matlab simulator running Windows XP. The task graphs used in the experiment were randomly generated using Task Graphs For Free (TGFF) [63]. We applied our real-time task allocation scheme and proposed algorithms to a WSN with heterogeneous nodes and compared their performance to the baseline case when the offline task scheduling scheme is used.

The simulation parameters are shown in Table 6.1.

6.2 Analysis of Simulation Results

6.2.1 Bid Convergence under Asynchronous Update Algorithm

The best bid values, which would result in the Nash Equilibrium (NE) defined in equations (4.12), (4.13) and (4.15) are evaluated. Furthermore, the convergence of bid values to the near optimal value given by theorem 2 is shown. The convergence occurs via the algorithm 1 which iteratively modifies β in an asynchronous manner. We first observe the bidding behavior of one bidder. The cost for accomplishing tasks 1 and 2 are 30 and 20 respectively, and the auctioneer's budget for task 1 and task 2 are 100 and 60 respectively. Figure 6.1 shows the bidding behavior for tasks 1 and 2 and their convergence to the near optimal value. The convergence of the distribution parameter β through the proposed algorithm is illustrated in Figure 6.2.

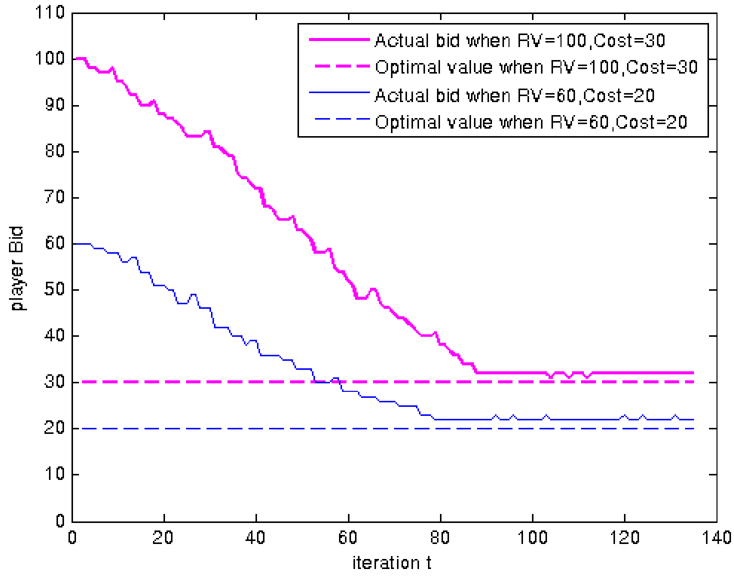


Figure 6.1: Convergence of bid under asynchronous updates.

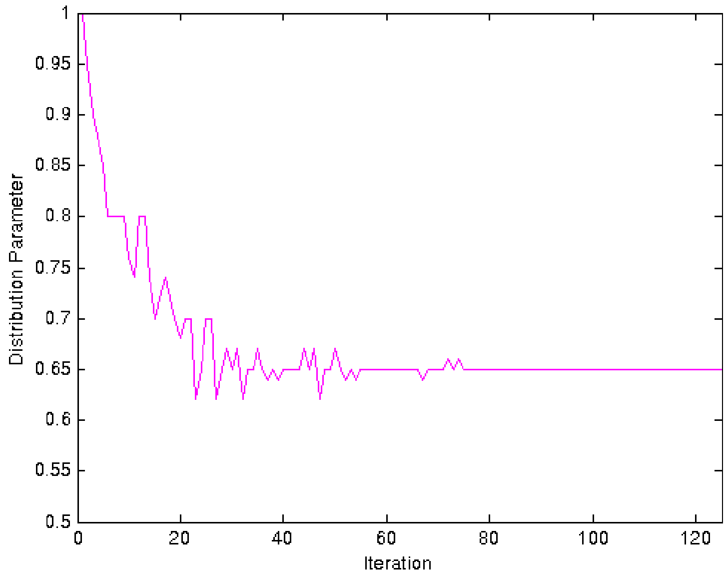


Figure 6.2: Convergence of distribution parameter β under asynchronous updates algorithm.

6.2.2 Performance Evaluation of Energy Balance and Energy Utilization

Another set of simulations was carried out to evaluate the energy balancing and energy utilization performance of the proposed distributed task allocation method. In this simulation, a task graph of 35 tasks, where each task has a maximum number of 3 predecessors is assign to 15 nodes. One sample of task graph is shown in Fig. 6.3. The real WSNs' application is not as complex as our assumptions for task graph. Hence, obviously for real application such as target tracking as shown the task graph in Fig. 3.2 would be much simpler. The nodes have the following initial energy levels $[3.4, 2.4, 2.0, 3.3, 2.0, 2.3, 2.7, 3.2, 3.1, 2.0, 1.9, 2.9, 2.7, 3.0, 3.0]$ Joule. The energy consumption for transmitting is based on the MICAz mote datasheet [64]: $E_{Sending} = 0.017$ mJ, $E_{Receiving(Listenmode)} = 0.031$ mJ. Comparisons of energy balancing performance are made between the proposed reverse auction-based scheme with ED-WDP and a static task allocation scheme. The static task allocation scheme used in our simulation is the Critical Node Path Tree (CNPT) algorithm [8] modified to schedule the tasks offline with an energy balance objective, referred to as EB-CNPT. In the EB-CNPT algorithm, given the initial available energy levels at each node and the number of tasks for an application, the clusters of tasks are formed. Before the task execution phase, each cluster of tasks are assigned to the sensor node as a whole, instead of allocating each individual task at each epoch by considering the resource and energy level availability of each sensor node at that epoch. In original CNPT algorithm, the only objective is to minimize the total schedule length considering the application deadline; however, in EB-CNPT, in addition to the schedule length, the initial energy level of nodes before the task assignment phase is also considered.

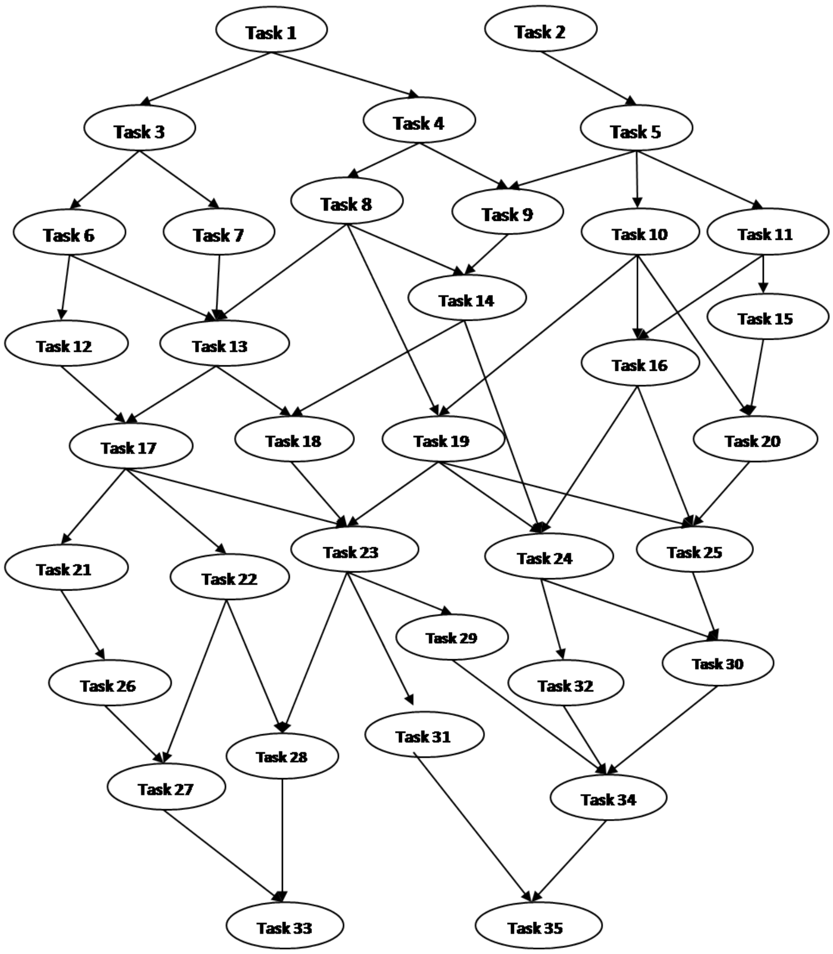


Figure 6.3: The sample of random generated task graph.

The remaining energy (Figure 6.4) for all the nodes appears more balanced when the proposed reverse auction scheme was used. The explanation is that the bid formulation at each iteration of task allocation takes into account and continuously adapts to the available resources. Figure 6.4 illustrates that given unbalanced initial energy level of each of 15 nodes, how our proposed reverse auction-based task allocation performs in comparison to static case with energy balance consideration (EB-CNPT).

In order to show the deviation from the mean of the remaining energy of all the 15 sensor nodes, the variance of data is used. The variance formulation for the available remaining energy is calculated as

$$Var = \frac{1}{n}\sum_{i=1}^{n}(E_i - \overline{E})^2 \tag{6.1}$$

where n is number of sensor nodes, E is the remaining energy level of sensor nodes and $\overline{E}$ is the mean which is calculated as $\overline{E} = \frac{1}{n}\sum_{i=1}^{n} E_i$. The higher variance value means the unbalanced available energy level among the sensor nodes. Figure 6.5 shows the performance of our scheme in terms of energy balancing and compared with the static energy balanced scheme (EB-CNPT).

6.2.2.1 Discussion on protocol cost:

Although the extra message exchange is required for online and real-time task allocation, the online task allocation is able to account the real availability of resources for task allocation. The performance in terms of energy balancing is already shown. However, we attempt to design the protocol (ED-WDP) that minimizes the number of message exchange required. As the result of two-fold phases of ED-WDP, only the winning node replies to auctioneer to avoid the extra message exchange typically requires for auction bidding mechanism.

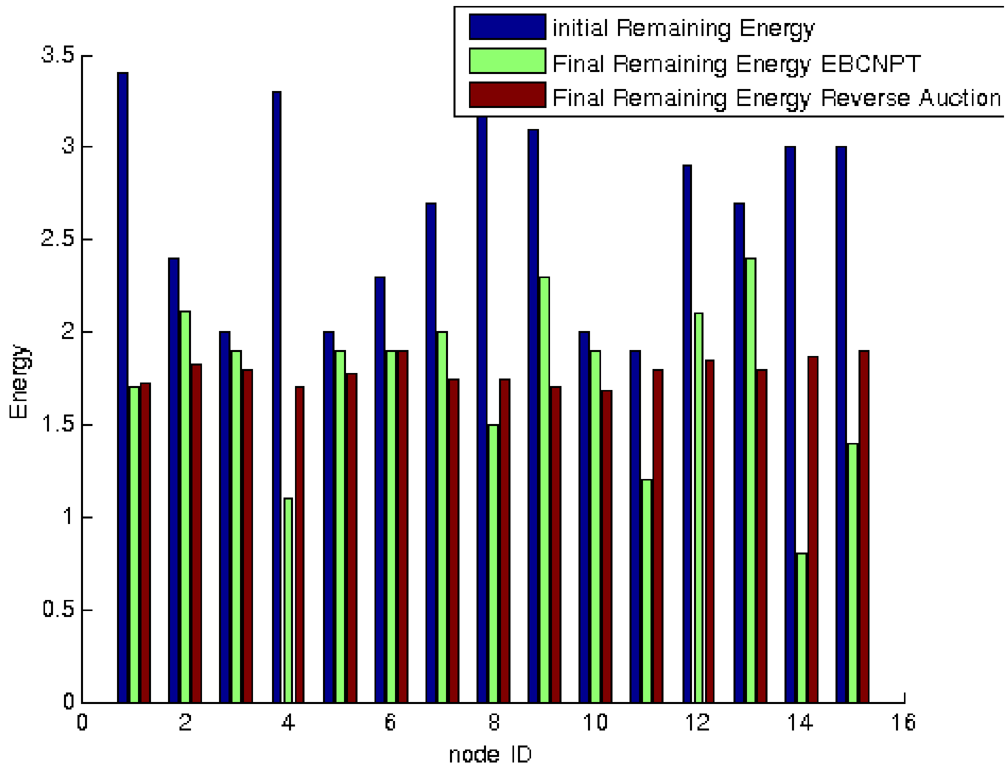


Figure 6.4: Comparison of the Level of energy balancing after allocating 35 tasks to 15 nodes.

6.2.3 Performance Evaluation of Energy Consumption and Schedule Length

The comparisons are made on the schedule length and energy consumption of our dynamic task allocation schemes with the three WDPs and the static task allocation EB-CNPT over an increasing number of available nodes. In the static task allocation, the schedule length for each node is the time taken for all the assigned tasks to be done. However, in the real-time task assignment, the schedule length is further increased by the time required for message exchange. The total schedule length considered in

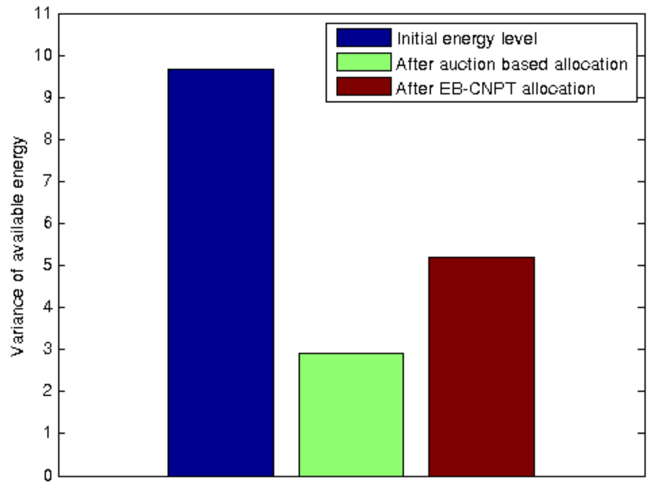


Figure 6.5: The performance of scheme in terms of energy balancing.

the simulation results is the maximum scheduling length among all the nodes. The schedule length and energy consumption achieved in static task scheduling is meant to be considered as the best scheduling scheme compared to the dynamic and real-time task allocation. However, in the static task allocation, the unexpected situations such as the packet loss, communication delay and node failure have not been considered.

Figure 6.6 shows the total energy consumption for adaptive task allocation with three Winner Determination Protocols $WDPs$ and static scheduling (EB-CNPT) when the number of available nodes are increased. The three protocols are: (1) centralized WDP (C-WDP) where all the bidders communicate their bids to the auctioneer; (2) distributed protocol called D-WDP where only one winning bidder sends its bid and waiting time is linearly proportional to bid the value; and (3) one is the proposed ED-WDP scheme.

As can be seen from Figure 6.6, the energy consumption of nodes increases by increasing the available nodes due to increasing the communication cost for migrating data among the nodes. The energy consumption of our scheme is lower than the static task allocation method due to the communication cost is considered on the bid value. Among the message exchange methods applied, the C-WDP results in the highest communication overhead and energy consumption. The ED-WDP scheme has the lowest energy consumption as only one node would communicate its bid and the time the rest of the nodes spend in the LISTEN mode is also reduced. Energy consumption is also reduced by allowing nodes with relatively low energies (nodes with bids that do not meet the budget) to be sent to the SLEEP mode directly.

Figure 6.7 shows the schedule length would decrease by increasing the number of available node as the tasks may be allocated over a larger number of nodes. The schedule length for real-time auction based task allocation is meant to be higher than static task scheduling due to the time taken to exchange messages; however since the efficient message exchange is used in our real-time allocation, it results in the comparable schedule length. Figure 6.7 also shows that among the different protocols, the ED-WDP (our proposed scheme) results in the lowest schedule length and the distributed scheme results in the highest schedule length due to higher T_w.

6.2.4 Performance Evaluation of Fast Recovery Scheme for Node Failures

In another set of experiments, the effect of node failure during the online task assignment phase is simulated. For the static case, rescheduling of the whole task graph has

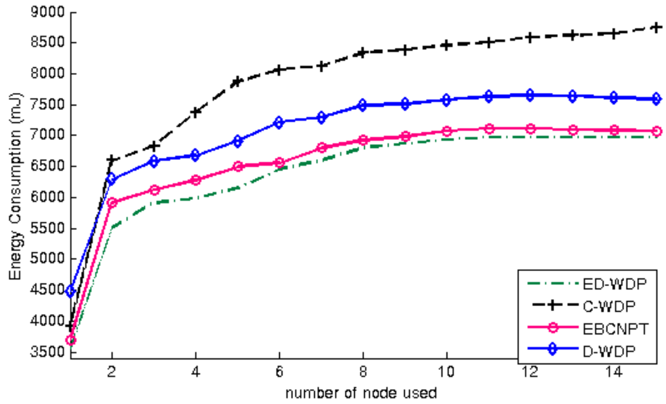


Figure 6.6: The total energy consumption when the number of nodes increases.

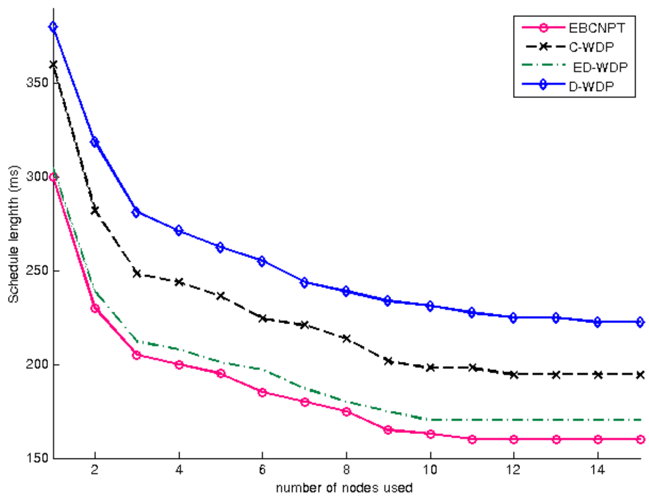


Figure 6.7: Schedule length when the number of nodes increases.

been performed. This is compared with the adaptive recovery phase in our proposed scheme. Node failure is simulated to occur at different times from the start time of task assignment. Figure 6.8 shows the performance of our dynamic scheme compared to the static case in terms of schedule length. Results exhibit a significantly lower schedule length and therefore, lower energy consumption in comparison to static case

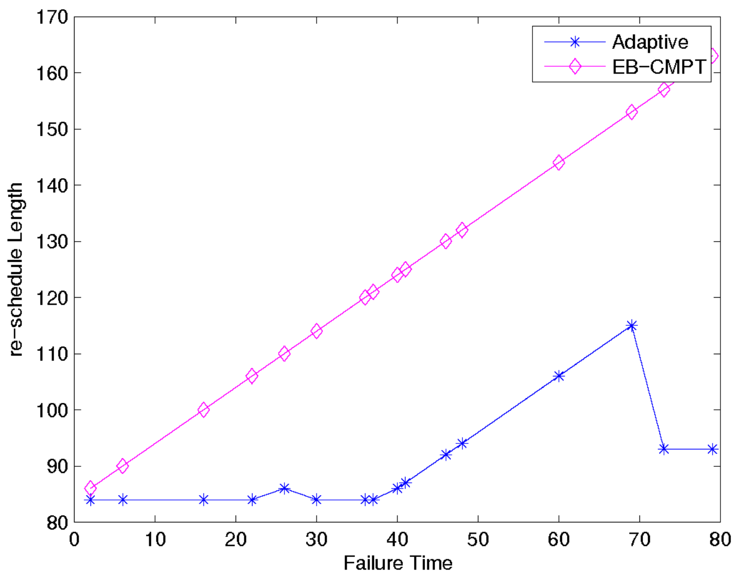


Figure 6.8: Scheduling length vs failure time

as shown in Figure 6.9. When node failure occurs during the early stages of task assignment, the schedule length is almost constant as not many tasks have been completed. When node failure occurs during the middle phases of the task assignment, the schedule length increases due to the number of uncommunicated dependencies resulting in the rescheduling and redeployment of many tasks. However, node failure occurs during the later times, many of the tasks have been completed and their dependencies have been communicated and therefore do not need to be redeployed.

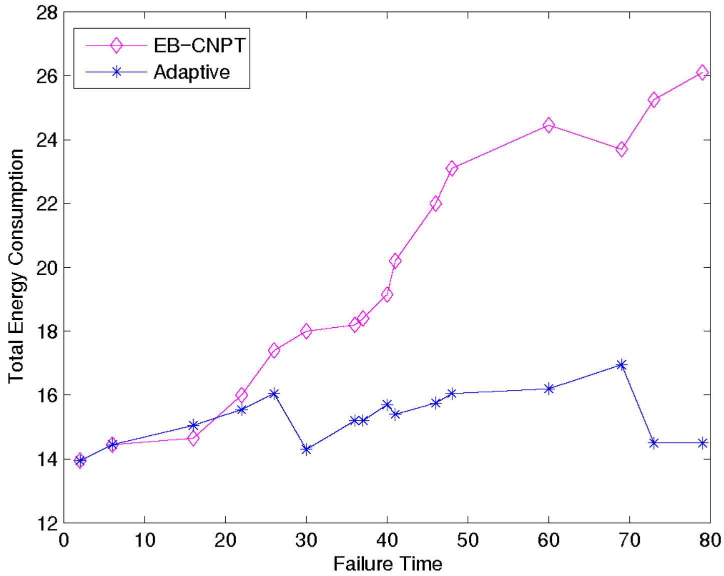


Figure 6.9: Energy consumption vs failure time

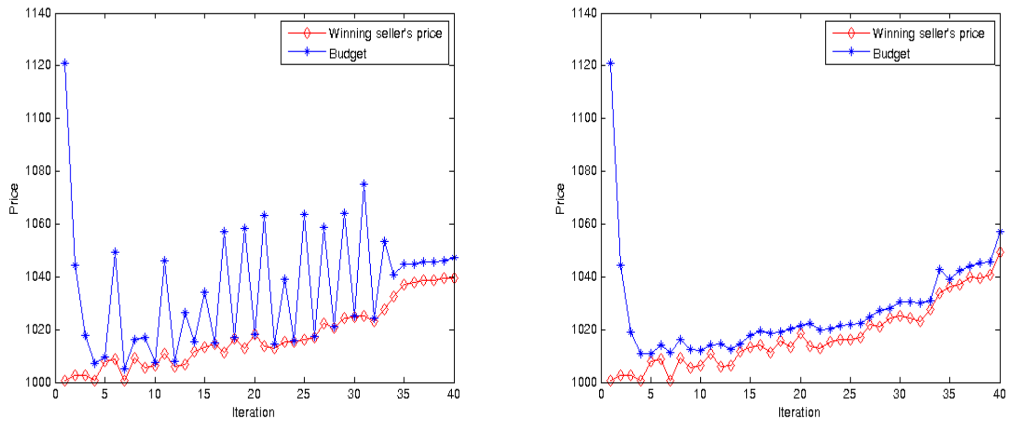
6.2.5 Performance Evaluation of Energy and Delay Efficient Distributed Winner Determination Protocol (ED-WDP)

To evaluate the ED-WDP in terms of total waiting time T_{total} and energy consumption, the simulation is carried out to show the result of adaptive algorithm 2 for adjusting the α value and its convergence after some iterations of task allocation. The best α value would lead to the best budget adjustment set by auctioneer and proper waiting time set by bidders. As explained before, the auctioneer faces the dilemma for setting the distribution parameter α. A higher α value results in a longer waiting time (the time before the winner responds) but a lower negotiation time for adjusting the budget value. The negotiation time is the contention window multiplies by the negotiation round. The negotiation round is required when all of the bidders are eliminated on phase 1 with the calculated budget, so the budget needs to be adjusted again. A lower α value has the opposite effect. The tradeoffs and comparisons of budget value, waiting time, number of negotiations and number of active nodes selected, given the low α value (0.2) and high α value (0.9) are made for a number of 40 tasks (iterations) and 10 nodes (sellers) with a contention window (CW) of 100 msec. Figure 6.2.5 shows the budget value set by auctioneer with the low α value and the high α value during different iterations. The high α results in a smooth estimated budget value during different iterations. Figure 6.2.5 illustrates the waiting time before the winner sends out its bid value achieved by low and high α values during iterations of tasks allocation. The number of negotiation rounds required for budget setting achieved by low and high α values during some iterations are shown in Figure 6.2.5. Figure 6.2.5 illustrates the number of selected nodes from 'elimination phase' and budget settings that is achieved by low and high α value.

As shown in Figure 5.4.3, different α values result in different total waiting times.

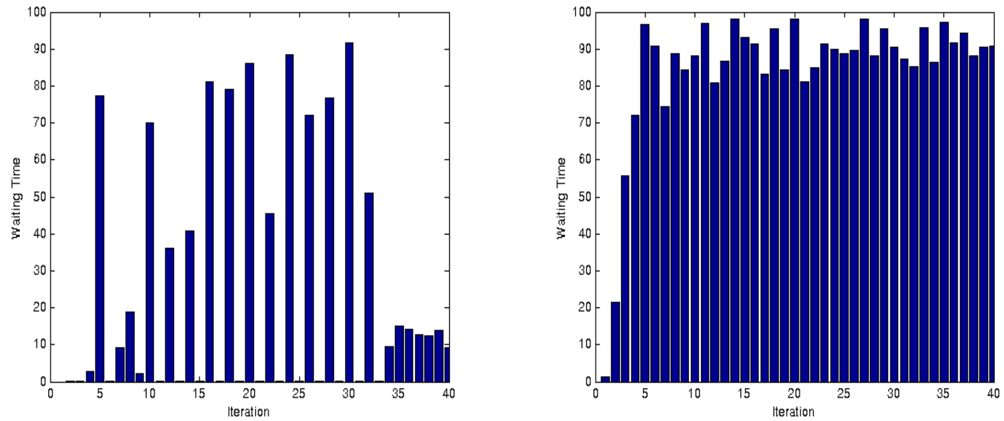
From Figure 5.6(c) the optimal value is determined to be between 0.6 and 0.7. The result of the adaptive algorithm for adjusting the α value, illustrates the convergence of α value to the near optimal value after some iterations. The initial α value is set to be 0.95 and after 22 iterations of task allocation this value converges to the near optimal value. The convergence result is shown in Figure 6.14.

Moreover, the total delay achieved by utilizing the adaptive α algorithm is compared with the total delay when the optimal α value, the fixed α values (low and high



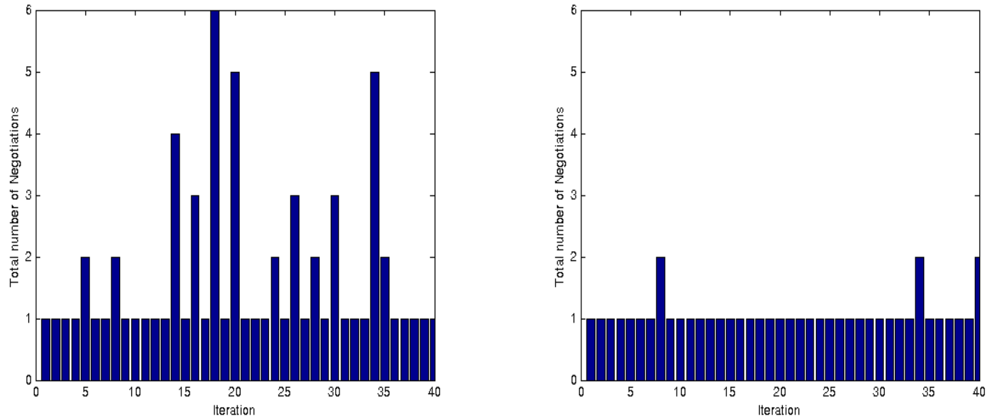
(a) (b)

Figure 6.10: (a) The budget value set by auctioneer with low α value during different iterations. (b) The budget value set by auctioneer with high α value during different iterations.



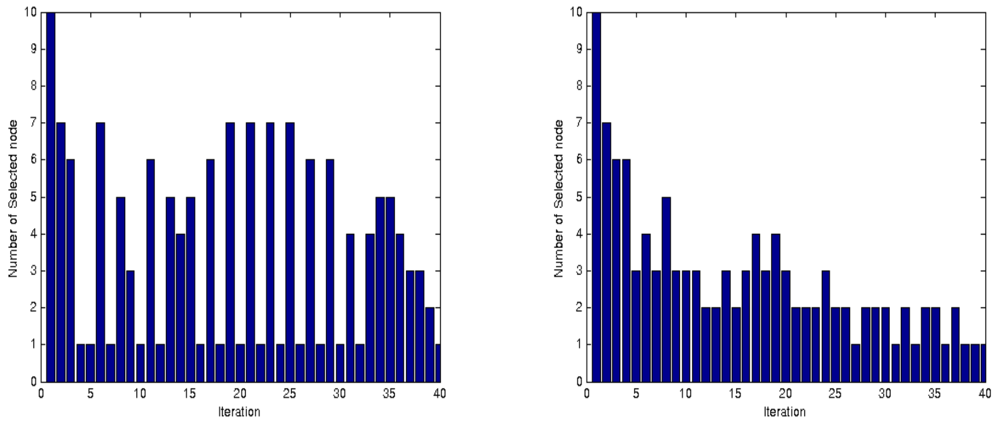
(a) (b)

Figure 6.11: (a) The waiting time before the winner sends out its bid value achieved by low α value. (b) The waiting time before the winner sends out its bid value achieved by high α value.



(a) (b)

Figure 6.12: (a) The number of negotiation rounds for budget setting achieved by low α value. (b) The number of negotiation rounds for budget setting achieved by high α value.



(a) (b)

Figure 6.13: (a) The number of selected nodes from 'elimination phase' achieved by low α value. (b) The number of selected nodes from 'elimination phase' achieved by high α value.

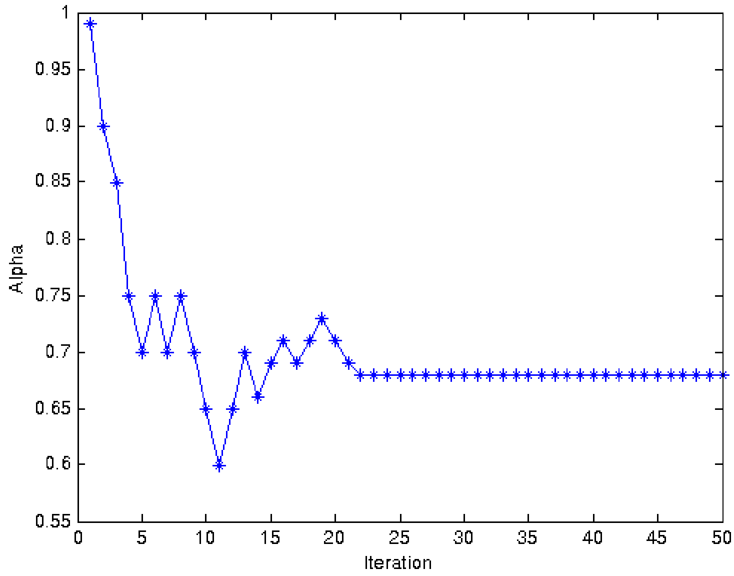


Figure 6.14: The convergence of α after several iterations

values) and a random α value (at each iteration α is randomly changed) are shown in Figure 6.15. The total delay for the case of adaptive α is near to that achieved with the optimal α value.

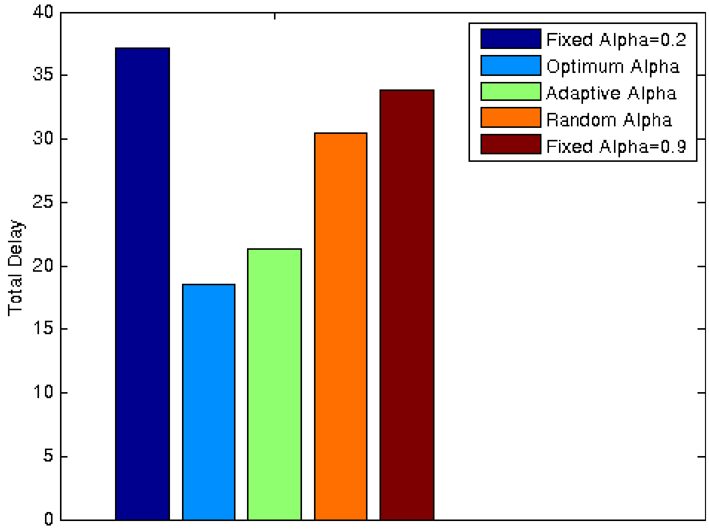


Figure 6.15: Comparison of total delay T_{total}.

Table 6.1: Simulation Parameters

Simulation Parameter	Value	Description
T	35	Number of tasks(iteration) in task graph
N	15	Number of sensor nodes
NumPred	3	Number of predecessor in task graph
κ	5	Number of prior rounds of winner's bid value for calculating market price
ρ	3	Number of iterations for consistent distribution parameter history CDPH
k	3	Number of iterations on which the bidder continuously win with consistent β
T_0	1100	Initial utility value
θ	0.15	Step
η	3	Number of observations of continuous winning with consistent β
α_0	0.95	Initial α value
β_0	0.9	Initial β value
ϵ	0.01	Threshold for step
a	0.5	Scaling parameter in bid formulation
b	1000	Preferred coefficient in bid formulation

Chapter 7

Conclusion and Future Work

In this dissertation, we have designed and evaluated the performance of the auction and game theory to deal with real-time task allocation in resource constrain wireless sensor networks. The proposed reverse auction scheme has been modeled as an incentive compatible and incomplete information game between auctioneer and bidders (sensor nodes). We have shown that each bidder by its non-smooth and concave payoff function can locally calculate its best bid response using an adaptive algorithm. The existence and uniqueness of the Nash Equilibrium has also been proved. We have shown that the proposed auction model meets the economically robust conditions such as truthfulness (incentive compatibility), individual rationality and ex-post budget balance. The main objective of this scheme which is maximizing network's lifetime (energy balancing), considering the application deadline has been achieved through the adaptive bidding and cost formulation. The cost formulation used continuously adapts to changes of the availabilities of resources. This scheme also accommodates for the node failure during task assignment via a recovery phase.

An energy and delay efficient decision making method has been obtained through the two-phased Energy and Delay Efficient Winner Determination Protocol ED-WDP. The proposed protocol that has the interpretation of contention-based MAC (Medium

Access Control) protocol operates in two phases. In the first phase, *elimination via budget value*, the number of players are eliminated via the budget value which is set by auctioneer and is the value that auctioneer wills to pay for arriving task. The consequence of this phase is to eliminate the players with low amount of available energy levels for the rest of competition. In the second phase, *waiting time reduction*, the duration of ideal listening mode as one of the important source of energy consumption for the players are significantly limited. Simulation results have shown the promising performance of the proposed schemes as well as the convergence of adaptive algorithms. These results have shown a fair energy balance achieved through the adaptive bidding and cost formulation, in comparison to the well-known static schemes. Moreover, compared to centralized WDP, by utilizing the ED-WDP among the numerous distributed resources, the message exchange overhead, energy consumption and delays for winner determination have been significantly reduced. In another set of experiments the effect of node failure during the real-time task assignment phase has been simulated and compared with the static recovery methods.

As part of our future work, we intend to investigate the problem of multiple concurrent tasks allocation through the combinatorial auction scheme. This problem potentially requires the combination or bundling of the proper resources from the set of distributed sensor nodes in the networks. Selecting the optimal set of resources is possibly very complex. Another problem that may arise is the winner determination problem for parallel scheduling of simultaneous tasks to multiple sensor nodes.

To apply the proposed dynamic reverse auction scheme for relay selection in cooperative communication is also challenging and interesting to explore.

Bibliography

[1] I.F. Akyidiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. "Wireless sensor networks: a survey," *Computer Networks (Elsevier) Journal*, pages 393-422, March 2002.

[2] I.F. Akyildiz, T. Melodia and K.R. Chowdhury, "A Survey on wireless multimedia sensor networks," *Computer Networks (Elsevier)*, vol. 51, pages 921 - 960, 2007.

[3] J. Liu, J. Reich and F. Zhao. "Collaborative in-network processing for target tracking," *EURASIP Journal on Applied Signal Processing*, pages 378 391,March 2003.

[4] B. Zitova and J. Flusser, "Image registrationmethods: A survey," *Elsevier Image and Vision Computing*, pages 977 1000, October 2003.

[5] L.K. Goh, V. Bharadwaj, "An energy-balanced task scheduling heuristic for heterogeneous wireless sensor networks," *Proc. of 15th International Conference of High Performance Computing (HiPC 2008)*, 2008.

[6] A. Pathak, V.K. Prasanna, "Energy-efficient task mapping for data-driven sensor network macroprogramming," *IEEE Transactions on Computers*, vol. 59, no. 7, pages 955 - 968, 2010.

[7] Y. Yu and V.K. Prasanna, "Energy-balanced task allocation for collaborative

processing in wireless sensor networks," *Mobile Networks and Applications*, volm 10, no. 1-2, pages 115 - 131, April 2005.

[8] Y. Tian, J. Boangoat and E. Ekici and F. Ozguner, "Real-time task mapping and scheduling for collaborative in-network processing in DVS-enabled wireless sensor networks," *Proc. 20th International Parallel and Distributed Processing Symposium* , Page 10, 2006.

[9] Y. Tian, Y. Gu, E. Ekici and F. Ozguner, "Dynamic critical-path task mapping and scheduling for collaborative in-network processing in multi-hop wireless sensor networks," *2006 International Conference on Parallel Processing Workshops*, 2006.

[10] R. Poornachandran, H. Ahmad and H. Cam, "Energy-Efficient Task scheduling for Wireless Sensor Nodes with Multiple Sensing Unit," *Conference Proc. of the 2005 IEEE International Performance, Computing and Communications Conference* pages 409 - 414, 2005.

[11] W. Xiao, S.M. Low, C.K Tham, S.K. Das, "Prediction based energy-efficient task allocation for delay-constrained wireless sensor networks," *Mesh and Ad Hoc Communications and Networks Workshops, SECON Workshops*, 2009.

[12] Y. Tian, F. Ozguner, E. Ekici, "Hyper-DAG based task scheduling wireless sensor networks in-network processing energy consumption parallel sensor processing," *20th International Symposium,* 2005.

[13] A.B. MacKenzie and L.A. Dasilva, "Game theory for wireless engineers," *Synthesis Lectures on Communications*, 2005.

[14] M.J. Osborne and A. Rubinstein, "A Course in Game Theory," *Cambridge, MA: The MIT Press*, 1994.

[15] D. Fudenberg and J. Tirole, "Game Theory," *MIT Press*, 1991.

[16] W. Saad, Z. Han, M. Debbah, A. Hjorungnes, T. Basar, "Coalitional game theory for communication networks - a tutorial," *IEEE Signal Processing Magazine,* vol. 26, no. 5, pages 77 - 97, Sept. 2009.

[17] V. Krishna, "Auction Theory," *Academic Press*, 2002.

[18] B.P. Harrenstein, M.D. Weerdt and V. Conitzer, "A qualitative Vickrey auction," *Proc. of the 2009 ACM Conference on Electronic Commerce*, pages 197 - 206, 2009.

[19] M. Babaioff and W.E. Walsh, "Incentive-compatible, budgetbalanced, yet highly efficient auctions for supply chain formation," *In Proc. of Economic Commerce*, 2003.

[20] X. Zhou and H. Zheng, "TRUST: a general framework for truthful double spectrum auctions," *IEEE INFOCOM. 28th International Conference on Computer Communications*, 2009.

[21] M. Schwind, "Dynamic Pricing and Automated Resource Allocation for Complex Information Services: Reinforcement Learning and Combinatorial Auctions," *Lecture Notes in Economics and Mathematical Systems, Springer*, 2007.

[22] S. Pal, S.R. Kundu, M. Chatterjee, S.K. Das, "Combinatorial reverse auction-based scheduling in multirate wireless systems," *IEEE Transactions on Computers*, vol. 56, no. 10, pages 1329 - 1341, Oct. 2007.

[23] Z. Liang, Y. Sun, L. Zhang and S. Dong, "Reverse auction-based grid resource allocation," *Agent Computing and Multi-Agent Systems*, 2006.

[24] F. Fu and U.C. Kozat, "Wireless Network Virtualization as A Sequential Auction Game," *Proc. of IEEE INFOCOM 2010, 29th International IEEE Conference on Computer Communications*, 2010.

[25] J. Huang, Z. Han, M. Chiang and H. V. Poor, "Auction-based resource allocation for cooperative communications," *IEEE Journal on Selected Areas in Communications*,vol. 26, no. 7, pages 1226-37, Sept. 2008.

[26] J. Huang, R.A. Berry and M.L. Honig, "Auction-based spectrum sharing," *Mobile Networks and Applications*, vol. 11, no. 3, pages 405-18, June 2006.

[27] L. Chen, B.K. Szymanski and J.W. Branch, "Auction-based congestion management for target tracking in wireless sensor networks," *IEEE International Conference on Pervasive Computing and Communications*, 2009.

[28] G.V. Radhanikanth and Y. Narahari, "Reverse combinatorial auction-based protocols for resource selection in grids," *International Journal of Grid and Utility Computing*, vol. 1, no. 2, pages 109 - 120, 2009.

[29] J. Neumann and O. Morgenstern, "Theory of Games and Economic Behavior," *Princeton University Press, Princeton*, 1944.

[30] P. Ghosh, N. Roy, S.K. Das, K. Basu, "A pricing strategy for job allocation in mobile grids using a non-cooperative bargaining theory framework" *Journal of Parallel and Distributed Computing*, vol. 65, no. 11, pages 1366 - 1383, Nov. 2005.

[31] P. Ghosh, N. Roy, S.K. Das, K. Basu, "A game theory-based pricing strategy to support single/multiclass job allocation schemes for bandwidth-constrained distributed computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 18, no. 3, pages 289 - 306, March 2007.

[32] P. Ghosh, N. Roy, S.K. Das, K. Basu, "A game theory based pricing strategy for job allocation in mobile grids," *18th International Parallel and Distributed Processing Symposium*, 2004.

[33] A. Viswanath, T. Mullen, D. Hall and A. Garga, "A. MASM: a market architecture for sensor management in distributed sensor networks," *Proceedings of the SPIE - The International Society for Optical Engineering*, vol. 5813, no. 1, pages 281-9, March 2005.

[34] A. Viswanath, M. Tracy, H. David and T. Sudheer, "An experimental study on agent learning for market-based sensor management," *Proc. 2009 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making, MCDM*, 2009.

[35] M. P. Wellman, "A market-oriented programming environment and its application to distributed multi-commodity flow problems," *Journal of Artificial Intelligence Research*, vol. 1, pages 1 - 23, 1993.

[36] K. Jamieson, H. Balakrishnan and Y.C. Tay, "Sift: A MAC Protocol for Event-Driven Wireless Sensor Networks," *Third European Workshop, EWSN*, 2006.

[37] J. F. Nash, "Equilibrium points in N-person games," *Proc. of the National Academy of Sciences*, 1950.

[38] R. Rugin, A. Conti, G. Mazzini, "Experimental investigation of the energy consumption for wireless sensor network with centralized data collection scheme," *15th International Conference on Software, Telecommunications and Computer Networks*, 2007.

[39] T. S. Chandrashekar, Y. Narahari, R. H. Charles, C. H. Rosa, D. M. Kulkarni, J. D. Tew and P. Dayama, "Auction-based mechanisms for electronic procurement", *IEEE Transactions on Automation Science and Engineering*, vol. 4, no. 3, pages 297 - 321, July 2007.

[40] R. Bapna, P. Goes and A. Gupta, "Simulating online Yankee auctions to optimize sellers revenue", *Proc. of the 34th Annual Hawaii International Conference on System Sciences*, Jan. 2001.

[41] G. Illing and U. Kluh, "Spectrum Auctions and Competetion in Telecommunications," *The MIT Press*, London, England, 2003.

[42] S. Fatima, M. Wooldridge and N. R. Jennings, "Revenue maximising agendas for sequential English auctions," *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems,* pages 1422 1433, 2004.

[43] K. Sako, "An auction protocol which hides bids of losers," *In Proc. of International Conference on Practice and Theory in Public Key Cryptography PKC*, pages 422 - 432, 2000.

[44] J. Kalagnanam and D.C. Parkes,"Auctions, bidding, and exchange design, in Handbook of Quantitative Supply Chain Analysis: Modeling in the E-Business Era," *ser. Int. Series in Operations Research and Management Science*, 2005.

[45] L. Anand, D. Ghose and V. Mani, "ELISA: an estimated load information scheduling algorithm for distributed computing systems," *Computers and Mathematics with Applications*, version 37, no. 8, pages 57 - 85, April 1999.

[46] Q. Zheng, C.K. Tham and V. Bharadwaj, "Dynamic load balancing and pricing in grid computing with communication delay," *Grid Economics and Business Models,* September 2008.

[47] Y. Tian, E. Ekici and F. Ozguner, "Energy-constrained task mapping and scheduling in wireless sensor networks," I*EEE International Conference on Mobile Ad-hoc and Sensor Systems,* 2005.

[48] D. Schmidt, C. Shi, R. Berry, M. Honig and W. Utschick, "Distributed resource allocation schemes," *IEEE Signal Processing Magazine*, September 2009.

[49] M.C. Foo, H.B. Kim, Y. Zeng, V.T. Lam and R. Teo "Impact of distributed resource allocation in sensor networks," *Proc. of International Conference on Intelligent Sensors, Sensor Networks,and Information processing*, 2005.

[50] G. Mainland, D. C. Parkes and M. Welsh, "Decentralized adaptive resource allocation of sensor networks," *Proc. 2nd USENIX/ACM Symposium on Networked*, 2005.

[51] H.B. Lim, V.T. Lam, M.C Foo and Y. Zeng, "An adaptive distributed resource allocation scheme for sensor networks," *Mobile Ad-hoc and Sensor Networks. Second International Conference, MSN*, 2006.

[52] T. Hagres and J. Janecek, "A high performance, low complexity algorithm for compile-time job scheduling in homogeneous computing environments," *Proc.*

2003 International Conference on Parallel Processing Workshops, 2003.

[53] A. Wang and A. Chandrakasan, "Energy-efficient DSPs for wireless sensor network," *IEEE Signal Processing Magazine*, pages 68 - 78, July 2002.

[54] Z. Wang, Y. Yan, P. Jia and S. Wang, "Market-based adaptive task scheduling for sensor networks," *2006 International Conference on Wireless Communications, Networking and Mobile Computing*, 2006.

[55] A. A. Lazar and N. Semret, "The progressive second price auction mechanism for network resource sharing," *In 8 th Int. Symp. on Dynamic Games and Applications*, 1998.

[56] T.Mullen and M. P. Wellman, "A simple computational market for network information services," *Proc. of the First International Conference on Multi Agent Systems, MIT Press*, San Francisco, CA, pages 283 - 289, 1998.

[57] T.Mullen and J.Breese, "Experiments in designing computational economies for mobile users," *Decision Support Systems Journal*, no. 28, pages 21 - 34, 2000.

[58] A. Andersson and F. Ygge, "Managing Large Scale Computational Markets," *Proc. of the 31st Hawaiian International Conference on System Sciences, IEEE Computer Society*, 1998.

[59] W.E. Walsh, P. Wellman, P.R. Wurman, and J.K. MacKie-Mason, "Some economics of market-based distributed scheduling," *In International Conference on Distributed Computing Systems*, 612 - 621, 1998.

[60] M. Wellman, W. Walsh, P. Wurman and J. MacKie-Mason, "Auction protocols

for decentralized scheduling," *Games and Economic Behavior*, vol. 35, pp. 271 - 303, 2001.

[61] T. Mullen, A. Viswanath, D. Hall and A. Garga, "Customer-driven sensor management," *IEEE Intelligent Systems*, March/April 2006.

[62] F. Yao, A. Demers and S. Shenker, "A scheduling model for reduced CPU energy," *IEEE Annual Foundations of Computer Science*, pages 374 - 382, 1995.

[63] R.P. Dick, D.L. Rhodes, W. Wolf, "TGFF: Task Graphs for Free," *Proc. 6th International Workshop on Hardware/Software Codesign,*pages 97101 March 1998.

[64] A. Barberis, L. Barboni and M. Valle, "Evaluating energy consumption in wireless sensor networks applications," *Proc. 10th Euromicro Conference on Digital System Design Architectures, Methods and Tools, DSD* pages 455-462, 2007.