The Hong Kong Polytechnic University

Department of Computing

Complex Event Detection in RFID and Wireless Sensor Networks

ZHU Weiping

A thesis submitted in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

September 2012

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

_____(Signed)

ZHU Weiping (Name of Student)

Abstract

Radio Frequency Identification (RFID) and Wireless Sensor Network (WSN) are important enabling techniques for data collection in many fields including mobile computing, pervasive computing, and internet of things. A great deal of useful information about surroundings is obtained through them, and then fed into upper-layer applications. Event detection is a data processing technique that is quite suitable for RFID and WSN applications. An event encapsulates raw data into a meaningful form that denotes a user-specified activity, and thus relieves the users from tedious underlying data processing. Compared with traditional event detection, new challenges are raised in RFID and WSN systems. The wireless communication used by them is error-prone and causes unreliable event detection results. Moreover, limited energy supply, computation capacity, memory and other resources in these systems demand more efficient and effective approaches of event detection. In this thesis, we investigate the event detection in RFID and WSNs to address these new challenges. We consider three important aspects of event detection: data collection, event aggregation, and event inference.

First, we study data collection, i.e. collecting data from surroundings to applications. We focus on reliable data collection in mobile RFID systems. One unique characteristic of RFID is that usually multiple RFID tags communicate with one reader simultaneously, which may cause collisions and unsuccessful identification of all the tags. This problem is even more serious in mobile RFID systems since the tags are moving and timely identification is required. Specially designed anti-collision protocols are needed to improve the identification rate of RFID tags. We propose a schedule-based RFID anti-collision protocol which, given a high identification rate, achieves the maximal tag moving speed. The protocol schedules an optimal number of tags to compete for the channel according to their identification deadlines, so as to achieve the optimal identification performance. Simulation results show that this approach can increase the moving speed of tags by 120% compared

with existing approaches, while achieving an identification rate of 99.999%.

Second, we study event aggregation, i.e. merging several sub-events into a composite event, and eventually the user required events. We focus on two energy-efficient event aggregation problems in WSNs. One problem is optimizing event aggregation utilizing complex relations in an event. The other problem is optimizing the event aggregation involving multiple events with different latency constraints and event relations (denoted by aggregation function). For each user specified event, a routing tree called event aggregation tree is usually built in a WSN to enable information exchange among sensor nodes for event aggregation. For the first problem, we utilize the complex relations included in an event to optimize the event aggregation tree. We propose principles of designing such an event aggregation tree. After that, we propose centralized and distributed algorithms to build this tree to achieve energy-efficient event detection. For the second problem, we first propose an approach to build energy-efficient event aggregation tree for individual events considering both latency constraint and aggregation function. We further optimize the routing structure for the aggregation of multiple events to save energy, by making some events share event aggregation trees instead of building their own. For both problems, simulation results show that our algorithms outperform existing approaches and save a significant amount of energy.

Third, we study event inference, i.e. infer the occurrence of an event through the information of other events. We focus on RFID reader localization, where detected RFID tags are used to infer the location of an RFID reader. It is a challenging task to achieve such an objective in the presence of long-lasting regional fault that means the RFID tags in a large region cannot response to the RFID reader for a long time period. We propose an effective localization approach which can tolerate such kind of fault, and define the quality index to measure the accuracy of a localization result obtained by our approach. Both 2D and 3D localization are discussed in our work. Our method also can be integrated into Multidimensional Scaling approach to solve network localization problem which involves multiple target objects to be located. We have taken extensive simulations and implemented an RFID-based localization system. In both cases, our approach outperforms existing approaches in localization accuracy and can provide additional useful quality information.

Publications

Journal Paper

1. **Weiping Zhu**, Jiannong Cao, Henry C. B. Chan, Xuefeng Liu, and Vaskar Raychoudhury, "*Mobile RFID with a High Identification Rate*", accepted by IEEE Transaction on Computers (TC), 2013

2. **Weiping Zhu**, Jiannong Cao, Yi Xu, Lei Yang, and Junjun Kong, "*Fault-Tolerant RFID Reader Localization Based on Passive RFID Tags*", submitted to IEEE Transaction on Parallel and Distributed Systems (TPDS)

3. **Weiping Zhu**, Jiannong Cao, Yi Xu, and Vaskar Raychoudhury, "*Efficient Detection of Multiple Composite Events in Wireless Sensor Networks using Event Aggregation*", to be submitted

4. Chao Yang, **Weiping Zhu**, Jia Liu, Lijun Chen, Daoxu Chen, and Jiannong Cao, "*Self-orienting the Cameras for Maximizing the View-Coverage Ratio in Wireless Camera Sensor Networks*", submitted to Pervasive and Mobile Computing

5. Junjun Kong, Jiannong Cao, **Weiping Zhu**, Tao Li, Yao Guo, and Weizhong Shao, "*Ubiquitous Interacting Object: A Distributed and Localized Approach to Building Ubiquitous Computing Applications*", submitted to ACM Transactions on Interactive Intelligent Systems

Conference Paper

1. **Weiping Zhu**, Jiannong Cao, Michel Raynaly, and Xuefeng Liu, "*Energy-efficient Composite Event Detection in Wireless Sensor Networks*", submitted to IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 2013

2. Xuefeng Liu, Jiannong Cao, Shaojie Tang, Zongjian He, and **Weiping Zhu**, "*Senet-SHM:Enabling Practical Structural Health Monitoring using Intelligent Sensor Networks*", submitted to EEE International Conference on Distributed Computing in Sensor Systems (DCOSS), 2013

3. Guanqing Liang, Jiannong Cao, and **Weiping Zhu**, "*CircleSense: Exploiting User's Physical Proximity For Social Activity Recognition Using Smartphones*", in Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom), 2013

4. **Weiping Zhu**, Jiannong Cao, Yi Xu, Lei Yang, and Junjun Kong, "*Fault-Tolerant RFID Reader Localization Based on Passive RFID Tags*", in Proc. of IEEE International Conference on Computer Communications (INFOCOM), pp.2183-2191, 2012

5. Lei Yang, Jiannong Cao, **Weiping Zhu**, and Shaojie Tang, "*A Hybrid Method for achieving High Accuracy and Efficiency in Object Tracking using Passive RFID*", in Proc. of IEEE International Conference on Pervasive Computing and Communications (PerCom), pp.109-115, 2012

6. Vaskar Raychoudhury, Jiannong Cao, **Weiping Zhu**, and Ajay D. Kshemkalyani, "*Context Map for Navigating the Physical World*", in Proc. of Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp.146-153, 2012

7. **Weiping Zhu**, Jiannong Cao, Yi Xu, and Vaskar Raychoudhury, "*Event Aggregation with Different Latency Constraints and Aggregation Functions in Wireless Sensor Networks*", in Proc. of IEEE International Conference on Communications (ICC), pp.1-5, 2011

Acknowledgements

I would like to express my gratitude to all those who helped me during my PhD. study. My deepest gratitude goes first and foremost to Prof. Jiannong Cao, my chief supervisor, for his systematic guidance and valuable suggestions. He has broad knowledge, keen insight, and enormous enthusiasm on the research, which inspire me a lot and encourage me to keep going in my study. He always trained me to be a good researcher, not only about the research methods and presentation skills, but also rigorous work attitude. I appreciate all these and will definitely benefit from him in my future work.

I would like to thank Dr. Henry Chan and Dr. Bin Xiao. They discussed with me from time to time, and gave me many constructive suggestions in my research and great help in the paper writing. I would like to thank Prof. Michel Raynal, for his kindly help about my research and living affairs during my six months visiting of his lab at France.

I would also like to thank my parents, my little sister, and my girl friend. They always encourage and support me when I encounter difficulties in different aspects. Their love is the most powerful motivation I can make progress in my work.

I want to thank my colleagues Dr. Xuefeng Liu, Dr. Vaskar Raychoudhury, Ms. Joanna Siebert, Mr. Lei Yang, Mr. Tao Li, Dr. Chao Ma, Mr. Chao Yang, Mr. Junjun Kong, Ms. Jingjing Li, Ms. Bingbing Zhou, Ms. Jie Zhou, Ms. Miao Xiong, Mr. Yang Liu, Mr. Gan Yao, Mr. Chisheng Zhang, Mr. Wei Feng, Ms. Yin Yuan, Mr. Zongjian He, Dr. Pen Guo, and all other members of our research group that I cannot enumerate here. Thank you for your help in these years. We learn from each other, share our joyfulness and sadness, and

have a unforgettable memory together. I wish all of you a brilliant future.

Last but not least, I also want to thank my roommates Dr. Qinjun Xiao, Mr. Yi Xu, Mr. Chen Ma, and Mr. Zhiwei Qing, and also my friends Mr. Yi Hong, Ms. Feng Wang and Dr. Lin Zhang, I am lucky to meet you in Hong Kong and appreciate your kindly help in my life.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ATI: Activated Tag Included Method

BES: Base Event Selection Algorithm

CDF: Cumulative Distribution Function

DBEA: Delay Bounded Event Aggregation Algorithm

EAT: Event Aggregation Tree

EDT: Event Definition Tree

EEAT-C: Energy-efficient Event Aggregation Tree Building Algorithm (centralized)

EEAT-D: Energy-efficient Event Aggregation Tree Building Algorithm (distributed)

EPC C1G2: EPC Class 1 Generation 2

FPTAS: Fully Polynomial Time Approximation Scheme

GDT: Group Definition Tree

GUI: Graphical User Interface

LFBES: Latency Constraint Fixed Base Event Selection

MDS: Multidimensional Scaling

MFST: Minimum Fusion Steiner Tree

RFID: Radio Frequency Identification

RSS: Received Signal Strength

SAC: Schedule-based Anti-Collision Protocol

SACO: Schedule-based Anti-Collision Protocol (optimal)

SACWR: Schedule-based Anti-Collision Protocol (without tag replenishment)

SDP: Semidefinite Programming

SPT: Shortest Path Tree

TED: Type-based composite Event Detection

TDOA: Time Difference of Arrival

TOA: Time of Arrival

TSP: Traveling Salesman Problem

WSN: Wireless Sensor Network

Chapter 1

Introduction

This research aims to investigate the issues and design novel algorithms, protocols, and system models for complex event detection in Radio Frequency Identification (RFID) and Wireless Sensor Networks (WSNs). In this chapter, we first describe the background knowledge of RFID and WSNs in Section 1.1. Then we introduce the event processing technique in Section 1.2. After that, we explain the motivation of our work in Section 1.3. In Section 1.4, we summarize the main contributions of this thesis. Finally, we outline the organization of this thesis in Section 1.5.

1.1 RFID and WSNs

RFID is a digital identification technology based on radio communication [Fin03]. A typical RFID system consists of an RFID reader and multiple RFID tags. The RFID tags, with pre-stored ID information, are attached to the target objects of interest. The RFID reader reads ID information from RFID tags to identify the objects. The area in which the RFID reader can communicate with RFID tags is called interrogation area. RFID tags can be classified into two major types: active tags and passive tags. The former operate using embedded batteries, while the latter are powered by radio waves sent from the RFID reader. In this thesis, we focus on passive tags that do not have the constraint of battery power supply and hence are widely used in many industry applications.

A WSN consists of a collection of sensor nodes interconnected through wireless communications [YMG08]. Each sensor node is equipped with one or more sensors of different

types. A WSN can sense, measure, gather information from the environment and, after proper processing, transmit the data to the user. It is self-organized and without human intervention, which makes it suitable for working in the hazardous and remote places.

Both RFID and WSN have a booming development in the last decade. They are important enabling techniques for data collection in many fields including mobile computing, pervasive computing, and internet of things. They have many common technical characteristics including wireless communication, and limited resources such as computation capacity, memory, energy supply, etc. In some applications, RFID reader is regarded as a kind of sensor and mixed used with traditional sensors. Therefore, we investigate RFID and WSN together in this research.

A large number of data are generated by RFID and WSN applications. In RFID applications, the data are the ID information of RFID tags. In WSN applications, the data are sensory data from various sensors, such as temperature, humility, speed, chemical substance content, etc. The data amount is large on one hand due to the large number of RFID tags and sensors, on the other hand due to frequent data collection. Usually, not all the data are useful for one specific user, and it is a tedious work for the user to extract the needed information from the data. An effective data processing technique is needed to simplify this work.

The data processing technique needs to not only handle the large amount of data, but also meet special requirements of RFID and WSNs. RFID and WSNs are based on wireless communication especially radio communication. It is error-prone and susceptible to environmental changes, and thus causes unreliable results. The access contention problem exists in both RFID and WSNs, but more serious in RFID, since there usually are many RFID tags communicating with one RFID reader. In a typical processing of an RFID system, the RFID reader sends out an identification request to the tags, and the tags reply with ID information. If multiple tags reply simultaneously, tag collision occurs and this leads to the unsuccessful identification of all the tags. For WSNs, more attentions are needed to pay to the limited energy supply, computation capacity, memory and other resources in

sensor nodes. All of these pose new challenges to design data processing approaches for RFID and WSNs.

1.2 Event Processing

Event processing is widely used to process data in many applications such as command and control systems, communication systems, and distributed computing systems etc. [Luc02]. An event is a record of an activity occurred in a system. Compared with raw data collected from the environment, an event encapsulates data in a user defined form hence more useful. The data in an application first form many low-level events and event processing is responsible for extracting useful information from these low-level events according to the user's requirements.

Generally speaking, event processing has several stages in its life cycle: event specification, event detection, event transmission, event storage and query.

At first, the user specifies the requirements in the form of events. The events can be primitive events or composite events. A primitive event can be directly detected by one sensor or one group of sensors. A composite event consists of multiple correlated subevents that are primitive events or other composite events. This definition is iterative so eventually a composite event can be decomposed into a set of primitive events. Composite event facilitates the user to define complex requirements. The event specifications are decomposed and injected into specific detecting nodes in the system. When an event occurs, it will be detected through the collaboration of detecting nodes. The detected events are further transmitted to other nodes for storage, which could be in current system or other external systems. The results are then offered to the user for query.

It can be seen that event detection is the base operation of other event processings after events are specified. In this thesis, we will focus on the event detection in different situation and different applications. Existing event specification approaches are adopted according to our requirements.

Event detection further includes data collection, primitive event detection, event aggregation, and event inference. When an event occurs, data collection is the first step of the processing of event detection. Primitive event detection directly follows data collection. Based on it, composite events are detected mainly by event aggregation and event inference. Event aggregation deduces high-level events from low-level events according to the relations included in the composite events. Since usually the data amount of high-level events is much smaller than that of low-level events, and only high-level events are useful for the user, event aggregation can reduce the data amount to be transmitted and then improve the system performance such as energy consumption, bandwidth, latency, etc. Event inference is another method to achieve event detection. Due to complex detection environment and user requirements, some events may not be directly detected but can be inferred from other related events.

Among these processings of event detection, primitive event detection is more related to signal processing and pattern recognition [GJV+05, WB09, ASMM12], which is not our focus. Except primitive event detection, we will investigate other aspects of event detection. More specifically, in this thesis, we aim to improve the complex event detection in data collection, event aggregation, and event inference for RFID and WSN applications.

1.3 Motivations of Our Work

When applying event detection technique to RFID and WSNs, we need to meet the requirements discussed in Section 1.1. Although some related issues are well addressed, there are still many problems lacking sufficient investigation. In this section, we identify the problems that need further investigation, and make them as our research topics in this thesis.

First, due to the characteristics of wireless communication, the reliability of data collection is critical for WSN and RFID systems. The reliability problem of sensory data collection is mainly related to node/link faults and environmental noise [CSR04, KI04, LDH06, DFDA11], which has been widely investigated. For RFID systems, besides overcoming the

impact of these factors, anti-collision identification is important to achieve reliable data collection. This problem is well solved in stationary environment but needs further research in mobile environment. In mobile environment, existing RFID anti-collision protocols cannot support high moving speed of tags and high identification rate simultaneously, which restricts the development of mobile RFID applications especially high-speed mobile RFID applications. The anti-collision problem we solved in mobile RFID applications can also help to solve the channel contention problems in WSN applications, considering some sensor nodes may move.

Second, energy efficiency is important for event aggregation in WSN applications. In RFID applications, RFID tags are usually passive RFID tags which are powered by radio waves sent from RFID readers, and RFID readers are usually equipped with wired power supply, so the energy consumption is not a major concern. However, in WSN applications, the sensors are usually powered by micro-batteries that cannot or hard to be recharged. Optimizing event aggregation to save energy can prolong the working time of WSNs. We have identified two important problems that still lack investigation. One is optimizing event aggregation utilizing complex relations in a composite event. The relations are specified by the user and can be quite complex. These complex relations on one hand facilitate the user to specify complex requirements, on the other hand make the optimization of event aggregation quite challenging. The other problem is optimizing the aggregation involving multiple composite events. Existing works consider the optimal event aggregation only for single composite event, but may have sub-optimal solutions when considering multiple composite events. The optimization of event aggregation may need to consider different factors, including the latency constraint specified by the user, and also the correlations included in the composite events.

Third, reliability is also important for event inference in RFID and WSN applications. Due to the errors introduced in data collection (and then the detection of some primitive events), additional information is usually needed to infer the correct results. One such application is RFID reader localization. In that application, an object carrying an RFID

reader is located by communicating with passive RFID tags deployed in the environment. The detected RFID tags are used to infer the location of an RFID reader. Frequent occurred RFID faults affect the localization accuracy. Specifically, complex localization environment (may include metal, water, obstacles, etc.) makes some tags fail to communicate with the reader, which makes the localization result deviate from the real location. Existing approaches can tolerate the faults occurred in individual tags and lasting for a short time period, but suffer serious localization error if the faults exist in a large region and last for a long time period. Moreover, existing approaches do not provide quality measurement of a localization result.

In this thesis, we will analyze aforementioned problems in detail and propose corresponding solutions for them.

1.4 Contributions of the Thesis

The contributions of this thesis mainly lie in designing novel algorithms, protocols, and system models for complex event detection in RFID and WSNs. As illustrated in Fig. 1.1, our contributions include three parts:

First, with respect to data collection, we design an anti-collision protocol for mobile RFID system to achieve high moving speed of tags while maintaining a high identification rate. Second, with respect to event aggregation, we design two algorithms to save energy consumption of WSNs when performing event aggregation. One algorithm optimizes the event aggregation utilizing the complex relations included in a composite event. Another algorithm optimizes the aggregation process of multiple composite events. Third, with respect to event inference, we design a fault-tolerant approach for RFID localization system, to infer an RFID reader's location through the identification of RFID tags deployed in the environment.

These three parts can be integrated into a complete process for complex event detection. Data collection is the first step of this process. Its results can be fed into event aggregation or event inference methods for the detection of complex composite events. All these processions

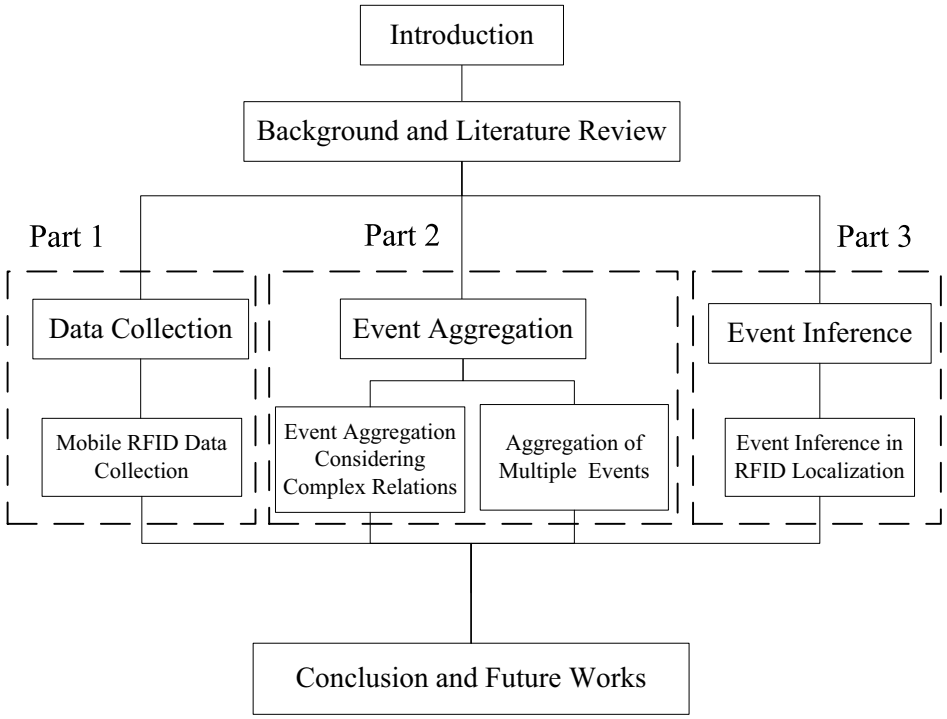


Fig. 1.1: An outline of the contributions in this thesis

may include both RFID data and sensory data. For example, in a mobile system that consists both RFID and sensor nodes, the results from our mobile RFID collection can be used for further event aggregation, together with other sensory data collected. In this situation, the method for building event aggregation trees still works. Take another example, in an RFID reader localization system, the tags may move in the region, and then the mobile RFID collection method can be used to get more reliable raw data for further inference of the location of the reader.

In the following, we will illustrate our contributions in detail one by one:

1.4.1 Contributions in Data Collection

Data collection is the first step of event detection. For the data collection in mobile RFID systems where RFID tags are moving, existing anti-collision protocols cannot support high tag moving speed and high identification rate simultaneously, because they are designed for stationary RFID systems and do not consider timely identification problem.

We propose a new RFID anti-collision protocol which, given a high identification rate, achieves the maximal tag moving speed. We first categorize different tag arrival models in mobile RFID systems. Different tag arrival models work for different purposes and require different protocol designs. Our work adopts a model that is suitable for an industry environment with dense tag placement and highs-peed moving tags. After that, we propose two principles for mobile RFID anti-collision protocols: workload optimal and the earliest deadline first. We argue that following these two principles can achieve optimal anti-collision identification in mobile environment. As a practice, we further propose the Schedule-based Anti-Collision Protocol (SAC) following these principles. Simulation results show that SAC achieves a 120% increase in tag moving speed compared with existing approaches, ensuring an identification rate of 99.999%.

1.4.2 Contributions in Event Aggregation

Event aggregation is an important method to achieve composite event detection. We solve two energy-efficient event aggregation problems in WSNs. One problem is optimizing event aggregation utilizing complex relations in a composite event. The other problem is optimizing the aggregation involving multiple events. In a WSN application, a routing tree is usually built to enable information exchange among sensor nodes for event aggregation. This tree is called event aggregation tree.

For the first problem, we aim to utilize the complex relations included in a composite event to optimize the event aggregation tree. The relations we considered are generic and compatible with event specification approaches. We explicitly utilize complex event relations to reduce the amount of data to be transmitted by aggregating low-level events into high-level events, at a cost of increased transmission distance. An optimal trade-off is made between the benefits and costs of such aggregations to minimize the overall energy consumption. We first propose principles to design such a tree. After that, we propose both centralized and distributed algorithms to build this tree to achieve energy-efficient event aggregation. Simulation results show that our proposed approach outperforms existing approaches and saves energy up to 20%.

For the second problem, we consider two factors impacting event aggregation: latency constraint and aggregation function. Latency constraint is a time limit specified by the user to finish the processing of event aggregation. Aggregation functions denotes the correlations among sub-events in a composite event. Existing works on event aggregation consider either latency constraint or aggregation function, but not both. They neither consider the optimal aggregation of multiple composite events with different latency constraints and aggregation functions. We first propose a new approach to build an energy-efficient event aggregation tree for individual composite events considering both latency constraint and aggregation function. We then consider the situation of multiple composite events. We optimize the routing structure in terms of energy consumption, by making some composite events share aggregation trees rather than building their own aggregation trees. To our best of knowledge, this is the first work to explore event aggregation in this aspect. Simulation results show that our approach outperforms existing approaches and saves significant amount of energy (up to 35% in our system).

1.4.3 Contributions in Event Inference

Besides event aggregation, event inference also can achieve composite event detection. We investigate a specific application of event inference: RFID reader localization, where the detected RFID tags are used to infer the location of an RFID reader. It is a challenging task considering that various faults may occur and then cause the detected tags different from those in an ideal environment. Our objective is to maximize the accuracy of localization and provide the accuracy measurement of each localization result.

We first formally categorize the RFID faults in RFID reader localization. Then we propose an effective localization approach which can tolerate the most challenging fault: long-lasting regional fault, which means the RFID tags in a large region cannot response to the RFID reader for a long time period. We also propose quality index to measure the accuracy of a specific localization result obtained by our approach. Both 2D and 3D localization are discussed in our work. Our method is further integrated into Multidimensional Scaling

(MDS) approach to solve network localization problem which involves multiple target objects. The locations and especially quality index provided by our approach are helpful to further improve the localization accuracy. We have taken extensive simulations and implemented an RFID-based localization system. In both cases, our solution outperforms existing approaches in localization accuracy and can provide additional useful quality information.

1.5 Organization of the Thesis

The structure of this thesis is shown in Fig. 1.1. Chapter 1 is the introduction to this thesis. Chapter 2 reviews related works in the literature. The main body of this thesis is divided into three parts from Chapter 3 to Chapter 6. The details are presented as follows.

In the first part, we mainly discuss our work in data collection. In Chapter 3, we design the Schedule-based Anti-Collision Protocol (SAC) for mobile RFID systems. Given a high identification rate, SAC can achieves the maximal tag moving speed. Different tag arrival models in mobile RFID systems are also discussed in this chapter.

In the second part, we mainly discuss our work in event aggregation. This part consists of two chapters. In Chapter 4, we investigate event aggregation considering complex relations in a composite event. We propose a new event aggregation tree to achieve energy-efficient composite event detection. In Chapter 5, we investigate the optimal aggregation of multiple composite events with different latency constraints and aggregation functions in WSNs. The Delay Bounded Event Aggregation Algorithm (DBEA) is proposed to build the optimal event aggregation tree for individual composite events, considering both latency constraint and aggregation function. After that, a solution is proposed to optimize the routing structure for the aggregation of multiple composite events.

In the third part, we mainly discuss our work in event inference. In Chapter 6, we investigate fault-tolerant RFID reader localization based on passive RFID tags. We formally categorize the RFID faults in RFID reader localization. And then we propose an approach which can tolerate long-lasting regional fault, and define quality index to measure localization results in both 2D and 3D environment. Our approach is also useful to provide more

accurate information to MDS approach for network localization.

Finally, we conclude the thesis and discuss the directions of future works in Chapter 7.

Chapter 2

Literature Review

In this chapter, we review existing works about complex event detection in RFID and WSNs. As we have discussed, we focus on mobile RFID data collection, event aggregation in WSNs, and RFID reader localization in this thesis. We first review the existing works about mobile RFID data collection in Section 2.1. Then we review the existing works about event aggregation in WSNs in Section 2.2. Finally, we review the existing works about RFID reader localization in Section 2.3.

2.1 Existing Works about Mobile RFID Data Collection

RFID is a digital identification technology based on radio communication. The most important problem in RFID data collection is its reliability. This problem is even more critical in mobile RFID systems since the tags are moving and thus timely identification is needed.

Same with other radio communication based technologies, RFID is affected by various environmental factors. The factors include the distance between the tag and the reader, inter-tag distance, the orientation of the tag respective to the antenna of the reader, different materials of the attached objects, interference from other electronic devices, characteristics of environments, etc. [FLS06, CTTB06, RZHJ07]. Different approaches are proposed to mitigate the effects of these factors using redundancy technologies including multiple readers/antennas/tags and multiple readings [RZHJ07, WBBB07, FRL07]. Several works are also undertaken in mobile RFID systems. In [SDOS07], different configurations such

as tag placement and orientation, tag moving speed, tag type, package materials, and the distance between the antenna and the tags, were adjusted to improve the identification rate. In [RTWL09], extensive experiments were undertaken in a real conveyor belt system to analyze the relations between the identification rate and tag moving speed. However, fully solving this problem is still subject to the development of radio communication.

Besides the effects of environmental factors, the anti-collision protocol design is also important to achieve reliable RFID data collecting.

Existing anti-collision protocols are classified into two categories: tree-based protocols and ALOHA-based protocols. Tree-based protocols [LLS00, BSI06] recursively split a set of tags into two subsets until the tags in a subset respond without collision. The Smart Trend Traversal Tree Protocol [PW09] further controls the process of splitting according to online learned tag ID distribution. These protocols are designed only for stationary environments. Since recursive tag splitting causes long latency and assumes an unchanged tag population to be identified, tree-based protocols are unsuitable for mobile environments, especially for high-speed mobile environments. This conclusion is consistent with [SDR08], where a study showed that in mobile environments the identification performance of tree-based protocols is heavily affected by the moving speed of the tags and the ratio of staying tags to arriving tags in the interrogation area. However, that paper did not propose any solution to this problem.

ALOHA-based protocols do not have aforementioned disadvantages of tree-based protocols, and hence are more suitable for mobile RFID environment. Currently, most ALOHA-based protocols are frame slotted ALOHA protocols [FW06]. The identification process is divided into several frames and each frame is divided into several time slots. In each frame, every tag randomly selects a time slot to transmit ID information. The transmission gets successful if no other tags transmit at the same time slot, or collides with other tags' transmissions and fails. The tags that fail to transmit information repeat the process in the later frames until all of the tags are identified. The performance of ALOHA-based protocols depends on frame size and tag cardinality. Current research mainly focuses on

how to adjust frame size according to tag cardinality (or collision information that infers tag cardinality). Vogt's dynamic slot allocation approach [Vog02a, Vog02b] and Cha & Kim's C-ratio approach [CK05] compute the optimal frame size based on the collision information gathered in previous frames. In [Flo07], a Bayesian strategy is utilized to further refine the result, but this strategy suffers high computation cost. In the industry, the EPC Class 1 Generation 2 Protocol [EPC07] uses a Q algorithm to determine the frame size, which is added or subtracted by a constant value whenever a collision slot or empty slot occurs, respectively. A more detailed analysis of dynamic frame size adjustment can be seen in [WLZ+07]. In [LJL05], the authors proposed the EDFSA algorithm, in which the frame size has an upper bound due to hardware restrictions. EDFSA divides the tags into groups by simple random selection and allows only one group of tags to participate in the identification. Some works also just focus on the fast estimation of tag cardinality. In [KN06], collision-based estimator or probabilistic estimator are employed to estimate tag cardinality depending on whether or not the tag cardinality can be roughly known in advance. In [QNL08], a lottery frame scheme is proposed to estimate the tag cardinality taking into consideration that multiple readers work collaboratively in an environment.

All aforementioned protocols are designed for stationary RFID systems. They perform poorly in a mobile environment since tag cardinality is quite hard to estimate due to the mobility of tags and the stochastic characteristics of ALOHA-based protocols. An inaccurate estimation of tag cardinality causes unstable identification performance. More importantly, they do not distinguish the identification deadlines of tags.

With the emergence of mobile applications, in [SDR08] the authors discussed how to set the parameters in a mobile RFID system to meet a desirable identification rate. In [XST+10], the authors proposed a probabilistic model for mobile RFID systems and used dynamic programming to determine the optimal frame size. The above two works have not taken into consideration the different identification deadlines of tags in a mobile RFID system.

2.2 Existing Works about Event Aggregation in WSNs

Event aggregation is an important approach to detect composite events. It aggregates the sub-events into composite events iteratively according to event specifications, and finally determines if the events of interest occur. Moreover, since the data amount of a composite event is usually less than that of its sub-events, event aggregation also reduces the data amount to be transmitted and hence saves energy, which is quite important to WSN applications. In our work, the focus is posed on the energy-efficient event aggregation.

Composite events are usually defined using event specification languages. The early works about this are from active database research, including Ode [GJS92], SNOOP [CM94], SAMOS [GD94]. A composite event is defined based on primitive events and event operators (such as disjunction, conjunction, sequence, etc.). More complex event definitions consider interval timestamp of event occurrence [GA02], and processing policy when multiple sets of sub-events can deduce a composite event [CM94]. RAPIDE [Luc02], a generic event specification language, is proposed to facilitate the applications in various of fields. In WSN applications, the works [KRJ05, LAV+10] consider different types of primitive events and the cardinality of sub-events needed to be detected to achieve reliable composite event detection.

It is noticed that other forms of specification method are possible for specific kinds of events. In [XLCL06, LLC08], the events are defined using spatial-temporal patterns of the data map of a morning environment. In a data map, the sensory data with similar values are combined in the form of contours. Spatial and temporal relations among contours are used to characterize events. These works are quite suitable for detecting the events that are based on the global state of a continuous area and in a continuous time duration, such as gas leakage, oxygen-enriched spot monitoring, water seepage in a coal mining. These works are different from our work since the events in our work do not necessary represent the global state of a continuous area or in a continuous time duration. Instead, they can be defined by the sensory data from separate areas at different time. Moreover, the relations included in an event are not only spatial and temporal relations, but also various logic relations, which

meets diverse user requirements.

Most of the existing works on composite event detection utilize the encounter opportunities of sub-events to aggregate sub-events into composite events. In [KRJ05], a collection of sensor nodes is selected to form a routing tree to guarantee that each sub-event can be detected by multiple sensor nodes for fault tolerance. The sub-events can aggregate into composite events in the routing tree if having opportunity. In [LAV+10], the authors adopt a similar routing tree, but considering more performance requirements on latency and energy consumption. Energy efficiency is achieved by selecting the sensor nodes with higher residual energy into the routing tree. In [ZGC+09], the information about event occurrence region is aggregated around the routing tree. In [LCF11], primitive events are forwarded to pre-selected fusion nodes, and then merged into composite events if possible. In all aforementioned works, event relations are not explicitly used to optimize the structure of the routing tree to save energy.

On the other hand, existing approaches of data aggregation can be used to build energy-efficient routing trees for some special kinds of composite events by explicitly considering the event relations. In [MFHH02], to detect the maximum temperature in a given area, the sensor nodes independently report primitive temperature events, and any two such events can be merged (keep the event with larger temperature value). This is called *full aggregation*. Generally speaking, fully aggregation denotes the aggregation where two data packets with one unit data amount can merge into a packet with one unit data amount. It is also investigated by [IGE00, KEW02, DCX03, LW06]. A more generic model is adopted in [CBLV04, ZVPS08], where each primitive event brings a fixed amount of new information into the aggregated event. In [CBLV04], it is denoted by a correlation coefficient $\rho = 1-r/R$ where R is the original data amount of a data packet and r is the data amount of aggregated data. $\rho = 0$ denotes no relation among sensing data. $\rho = 1$ denotes full aggregation. When $\rho = 0$, the optimal routing tree is the shortest path tree (SPT). When $\rho = 1$, the authors proved that it is a multiple traveling salesman problem (TSP). When $0 < \rho < 1$, the authors proved that it is also a NP complete problem. Leaves Deletion Algorithm

and Balanced SPT/TSP approach are proposed as the heuristic algorithms. In [ZVPS08], the authors further formulated the event aggregation among multiple composite events. Assuming that there are n sub-events each of which has a data amount of m, and every two sub-events have the same correlation coefficient ρ, the aggregated event has a data amount of $m + (1 - \rho)m$. In [GE05], the relations are defined using *aggregation function*. Given j sensor nodes participating in the aggregation and each of them having one unit data amount, the data amount after aggregation is $f(j)$, where f is concave, non-decreasing, and $f(0) = 0$. The paper proposed a random algorithm to build the aggregation tree suitable for all this kind of aggregation functions. The paper has a strong assumption on the aggregation function which may not always hold. Moreover, it considers all the source nodes having equal data amount. In [LLD06], the aggregation function is defined in a relaxed way: if event u (with a data amount of $w(u)$) and event v (with a data amount of $w(v)$) aggregate, the result data amount is no less than $w(u)$ or $w(v)$. Minimum Fusion Steiner Tree (MFST) is proposed with similar idea with [GE05] but further takes the fusion cost into account.

Although these works are helpful to achieve energy efficiency for these special kinds of composite events, they cannot support composite events with generic relations. The relations included in a composite event can be quite complex, and hence the aggregation functions are not necessary no-decreasing, while no-decreasing aggregation function is a common assumption in data aggregation. In our work, one of the focus is to design a new kind of routing tree for energy-efficient event aggregation, in which generic event relations are supported and fully utilized.

Another concern of us is the event aggregation considering an additional latency constraint. An event aggregation considering both energy consumption and latency constraint is more desirable for many applications with service delivery deadlines. In [MCM+06], the authors studied the aggregation combining two objectives: delivery latency and energy consumption, with different priorities. Latency-oriented strategy and energy-oriented strategy are proposed. Latency-oriented strategy firstly minimizes the latency and secondly minimizes the energy consumption while energy-oriented strategy reverses the priority sequence.

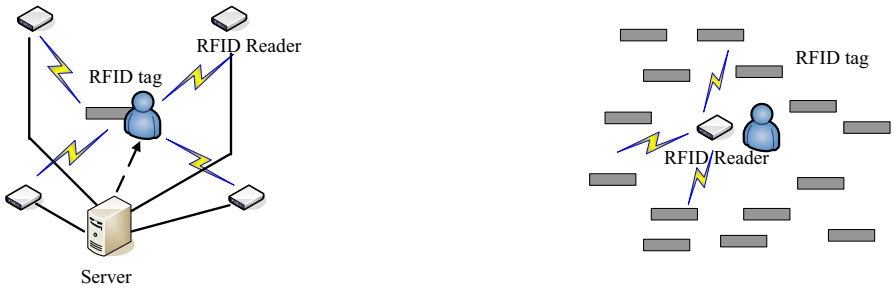
This paper only considered full aggregation. In [YKP04], the authors studied how to schedule packet transmissions to achieve energy-latency tradeoff. Given a latency constraint of the event aggregation tree, the paper proposed online and offline algorithm to schedule each packet's transmission(starting time and transmission time) with minimal energy consumption. Both latency constraint and aggregation function are considered there, however, the type of aggregation is simply full aggregation. The work of [BKMs+06] also only considered full aggregation. Further efforts are needed to investigate the event aggregation considering generic relations.

Moreover, to our best knowledge, all the existing works optimize the event aggregation base on individual composite events. It lacks the research on how to optimize the event aggregation in the presence of multiple composite events. Considering there are usually multiple composite events specified in an application to be detected [Luc02], further optimization can save more energy. One focus of our work is to consider the event aggregation of multiple composite events with different latency constraints and aggregation functions.

2.3 Existing Works about RFID Reader Localization

RFID reader localization is a specific application of event inference where the detected RFID tags are used to infer the location of the RFID reader.

With the growing use of RFID-based devices, many researchers are interested in utilizing RFID to provide localization service [HBF+04, LL06, WWT07, BP08, SN11]. RFID-based localization can be classified into *tag localization* and *reader localization* [SK08]. In the tag localization as shown in Fig. 2.1(a), each object to be located is attached with an RFID tag and RFID readers are scattered in the environment. A server gathers data from the readers, executes a localization algorithm and notifies the result to the object. On the contrary, in the reader localization as shown in Fig. 2.1(b), the object carries an RFID reader and a set of RFID tags are deployed in the environment. The object uses the reader to actively obtain its own location. Compared with tag localization, reader localization reduces infrastructure cost by using cheap tags instead of expensive readers. Moreover, unlike tag localization



(a) tag localization (b) reader localization

Fig. 2.1: RFID-based localization

with centralized computing, reader localization is inherently distributed and more scalable. We focus on reader localization in this work.

Most of the works related to reader localization utilize geometric and range-free methods to localize the target object.

In [BHE00], a low-cost outdoor localization method based on radio communication is proposed. There are a set of reference nodes deployed in the environment and transmitting beacon radio signals periodically. The target object receives the signals and locates itself to be the centroid of proximate reference nodes. In APIT [HHB+03], the target object interacts with neighboring objects to determine whether it is in a triangle formed by three reference nodes. Thought this, the target object narrows the possible region in which it resides, and finally locates itself as the centroid of the possible region. This method is useful in the scenario with multiple readers which can interact with each other. The virtual landmark [BP08] utilizes the connectivity information that an RFID reader can detect an RFID tag or not, and virtual reference tags, to determine the region that the tag may be in and then computes the centroid. In [SWJ+05], the centroid method is improved to the weighted centroid method by assigning a weight to each reference node based on RSS (Received Signal Strength). With a little difference, LANDMARC [NLLP03] deploys both readers and reference tags as the infrastructure. When a target object comes into an reader's identification region, k nearest tags are selected to calculate the weighted centroid as the localization result. The weight of a reference tag is inversely proportional

to the similarity in RSS between it and the target object. In [LL06], the authors also use weighted centroid method but the weights are defined by a Gaussian function. In [WWT07], the authors proposed a non-linear programming method to locate the reader based on the geometric knowledge of the reader's identification region. In [PLK+10], the effect of different tag placements on localization are discussed. Early works of the centroid method did not consider the fault-tolerance problem. Later works [NLLP03, SWJ+05, LL06] considered RFID faults and utilized spatial/temporal redundancy to select more reliable tags for localization. However, all existing works are not capable of coping with long-lasting regional fault during the localization process.

Rather than the geometrical method, machine learning based localization method is also developed. In [KK04], the RSS at different locations are first recorded as fingerprints. After that, the target object measures its RSS and match it with recorded fingerprints. The location of the most similar fingerprint is returned as the localization result. The probabilistic measurement model is further combined into the method to improve localization accuracy [HBF+04]. In [YTH+04], the authors use support vector machine to carry out the training and matching. The machine learning based method needs a time-consuming training process and a large database. For many applications, this may not be practical, so we do not consider this method in this research.

Few existing works provide quality information of localization results. For trilateration-based localization methods, Geographic Dilution of Precision is proposed to measure the error caused by geometric forms of multilateration [Spi01]. In [YL10], the quality of a localization result is defined based on geometric forms of trilaterations and the confidence of reference nodes. These two methods are for ranged-based localization and are not applicable to our problem. The work [BP08] narrows the region where a target may resides and measure the quality of a localization result as the volumes of that region. It is only a coarse-grained measurement for the ranged-free methods.

With regard to the network localization involving multiple target objects, multidimensional scaling (MDS) is a well-known method that utilizes the pairwise distances of objects

and some anchor objects to localize the objects of interest [CC01]. The basic idea is to find the locations for these objects that fit the measured pairwise distances as well as possible. Classical MDS needs the distances of any pair of objects, and then perform centralized computation [Gow66]. It is also used in RFID tag localization [SW11]. Distributed solution is proposed in [CPI04] considering that some objects only can measure the distances from neighboring objects. Besides anchor objects, this method utilizes uncertain objects whose initial locations are inaccurate subject to some accuracy degree. This method is iterative and hence time-consuming, and the authors do not mention how to obtain the uncertain objects. In [WZYB08], the authors solve a similar sensor localization problem using novel SDP (Semidefinite Programming) relaxation, which is validated much faster than existing algorithms. It considers only anchor objects but not uncertain objects. In this thesis, we aim to provide such uncertain objects to further improve the localization accuracy. For measuring the pairwise distances of objects, we can use the ranging techniques in sensor research including RSS (Received Signal Strength), TOA (Time of Arrival), and TDOA (Time Difference of Arrival)[WGD10], or utilize RFID readers that can adaptively control transmission power [AAHI10].

Chapter 3

Reliable Data Collection in Mobile RFID Systems

In this chapter, we investigate the reliable data collection problem in mobile RFID systems. We propose a new anti-collision protocol which, given a high identification rate, achieves the maximal tag moving speed. This chapter is organized as follows: Section 3.1 is the overview of this work. Section 3.2 describes the system model. Following this, Section 3.3 discusses the design principles of mobile RFID anti-collision protocols and Section 3.4 proposes our protocol in details. Simulation results are reported in Section 3.5. Finally, Section 3.7 concludes this chapter.

3.1 Overview

RFID is a rapidly developing digital identification technology based on radio communication. Reliable data collection problem is an important requirement of RFID systems.

In a typical working process of an RFID system, the reader sends out an identification request to the tags, and the tags reply with ID information. If multiple tags reply simultaneously, tag collision occurs and this leads to the unsuccessful identification of all the tags. Therefore, specially designed anti-collision protocols (e.g., tree-based protocols [LLS00, BSI06, PW09] and ALOHA-based protocols [Vog02a, Vog02b, CK05, LJL05, Flo07,

EPC07]) are employed to improve the success rate of tag identification, which is called *identification rate* in RFID technology.

Existing anti-collision protocols are designed only for stationary RFID systems. However, currently there is a growing interest in mobile RFID systems where tags keep moving. Consider a practical example of the RFID-based baggage processing system in Hong Kong International Airport, which deals with about 40,000 pieces of baggage every day [HKI08]. The baggage is attached with RFID tags and placed on a moving conveyor belt. Many other applications also deploy an RFID system with conveyor belts and have tags moving, including those in retail distribution (Wal-Mart [Rob04]), correspondence/parcels auto-sorting (China Post [Bac06], Australia Post [ZYW06]), pharmaceutical processing automation (Purdue Pharma [Imp09]), and food management (a Japanese sushi restaurant [NSL08]).

It is highly desirable that mobile RFID system can support a high moving speed of tags while maintaining a high identification rate. However, existing anti-collision protocols, including tree-based protocols and ALOHA-based protocols, are ill-suited for mobile tags.

The tree-based anti-collision protocols recursively split a set of tags into two subsets until there is only one tag in a set. They are unsuitable for mobile RFID systems because recursive tag splitting causes long latency and assumes an unchanged tag population in the interrogation area.

The ALOHA-based anti-collision protocol is the major focus of this work. It divides the identification process into several frames and each frame is divided into several time slots. In each frame, every tag randomly selects a time slot to transmit ID information and is successful at doing so only if no other tags select the same time slot. The performance of the protocols largely depends on *frame size* (the number of time slots in a frame) and *tag cardinality* (the number of unidentified tags in the interrogation area). Although many approaches have been proposed to dynamically adjust frame size according to estimates

of tag cardinality, they perform poorly in a mobile environment for the following reasons. First, it is hard to estimate tag cardinality due to the mobility of tags and the stochastic characteristics of ALOHA-based protocols. An inaccurate estimation of tag cardinality causes unstable identification performance. Second, even if tag cardinality can be well determined, the improvement is limited and a high moving speed of tags still cannot be supported. This is because existing approaches treat each tag equally, whereas in a mobile RFID system recognizing unidentified tags that reach the edge of the interrogation area is a priority. Depending on their positions in the interrogation area, tags have different identification deadlines.

In this work, we propose a new anti-collision protocol which, given a high identification rate, achieves the maximal tag moving speed. We schedule an optimal number of tags to compete for the channel according to their identification deadlines, without the need to estimate tag cardinality. We first distinguish different tag arrival models in mobile RFID systems. Our work adopts a model that is suitable for an industry environment with dense tag placement and high-speed moving tags. We propose two principles for mobile RFID anti-collision protocols: *workload optimal* and *the earliest deadline first*. The former is used to maintain an optimal number of tags competing for the channel so as to guarantee the identification rate. The latter is to assign a high identification priority to the tags that have tight identification deadlines. Following these principles, the Schedule-based Anti-Collision Protocol (SAC) is proposed as the main focus of this work. Extensive simulations are carried out to evaluate our protocol. So far, to the best of our knowledge, our protocol supports the highest tag moving speed while maintaining a high identification rate.

We also point out that the access contention problem exists in both RFID and WSNs due to commonly used radio communications. Therefore the idea discussed in our work,

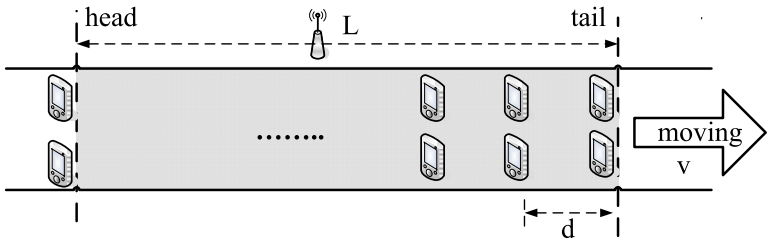


Fig. 3.1: A mobile RFID system on a conveyor belt

distinguishing the identification priorities of different nodes, can also be applied to WSN applications, considering some sensor nodes may move in some applications. Moreover, RFID devices are often considered as a special kind of sensor nodes and included in WSNs. The data collected by mobile RFID systems, will be used in WSNs for further event aggregation and event inference, just as other sensory data, which will be discussed in later chapters.

In summary, this chapter makes the following specific contributions.

- We distinguished different tag arrival models in mobile RFID systems.
- We proposed two principles for mobile RFID anti-collision protocols.
- We proposed the Schedule-based Anti-Collision Protocol (SAC), which achieves a 120% increase in tag moving speed compared with existing approaches, ensuring an identification rate of 99.999%.

3.2 System Model

We first describe the system model used in our work. In this model, how the tags arrive at the interrogation area is an important factor when designing an anti-collision protocol. Therefore, we further discuss tag arrival model in detail.

3.2.1 Mobile RFID System Model

Consider a general mobile RFID problem where there is a fixed RFID reader and a collection of mobile RFID tags. The tags move through the interrogation area. The identification rate is defined as a ratio of the number of identified tags to the total number of tags passing through the interrogation area. The goal is to achieve the maximal tag moving speed subject to an acceptable identification rate.

We use a conveyor belt application to demonstrate this problem, as shown in Fig. 3.1. A conveyor belt runs at a constant speed of v to transport items placed on it. The direction of movement is from the head to the tail. Each item is attached with an RFID tag. An UHF (860-960 MHz) RFID reader is deployed beside the conveyor belt to detect the tags. The reader has a fixed interrogation area, which is described using a length of L in the conveyor belt. n tags are put in two parallel lines in the conveyor belt. In each line, the distance between two neighboring tags is d. The orientation of the tags with respect to the antennas of the reader is tuned on-site to have the best identification performance. In each line, the distance between two neighboring tags is d, which is large enough to avoid the interference. The tag density p is defined as the number of tags per unit length of the conveyor belt (e.g., four tags per meter). Assuming there is no physical fault in the reader and the tags, the system needs an anti-collision protocol to achieve the maximal conveyor belt speed and maintain a high identification rate r (e.g., 99.9%, 99.999%, etc.).

3.2.2 Tag Arrival Model

In a mobile RFID system, tag arrival model describes how the tags enter the interrogation area and the relations among the tags. Tag arrival models are classified according to two factors: the number of tags in the interrogation area (NT) and the distance between consecutive tags (or sets of tags) (DT). In our system model, two tags in a row are grouped

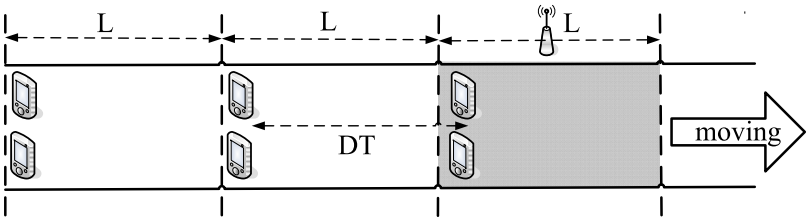


Fig. 3.2: The isolated constant arrival model

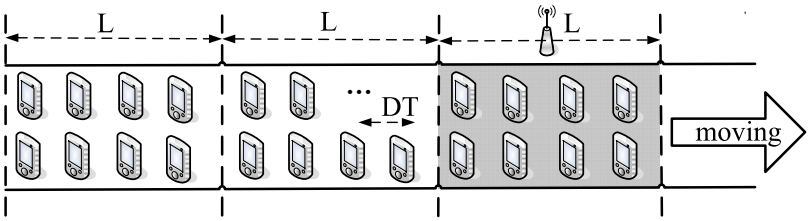


Fig. 3.3: The dynamic constant arrival model

as a set, so DT denotes the distance between consecutive rows of tags. Specifically, we have the following formal definitions where L denotes the interrogation area:

Constant arrival: if NT keeps a constant value at any time, this is called constant arrival.

Variable arrival: if NT varies with time, this is called variable arrival.

Isolated arrival: if $DT \geq L$, a new tag (or tag-set) enters the interrogation area only after the previous tag (or tag-set) has left. This is called isolated arrival.

Dynamic arrival: if $DT < L$, the tags in the interrogation area keep changing. This is called dynamic arrival.

With the combination of these two factors, we have four basic types of tag arrival model: *isolated constant arrival*, *isolated variable arrival*, *dynamic constant arrival*, and *dynamic variable arrival*. As shown in Fig. 3.2, in an isolated constant arrival model with $DT = L$, a row of tags come into the interrogation area, and then they are kept in the interrogation

Table 3.1: The research based on different arrival models

Research	**Dynamic arrival**	**Isolated arrival**
Constant arrival	In [SMSC08], [NSL08], [Imp09] and this work	In [SDOS07] [RTWL09] [XST+10]
Variable arrival	Open question	Open question

area and no other tags come in until they move out. If $DT < L$, the model changes to a dynamic constant arrival model, as shown in Fig. 3.3. Notice that constant arrival and dynamic arrival are not conflicting, since the number of tags in the interrogation area can remain the same but the tag instances can change constantly. As shown in Fig. 3.3, in each time unit, a row of tags moves out and another row of tags moves in, bringing no change to the number of tags in the interrogation area, but the tag instances are different.

The research based on different arrival models is summarized in Table 3.1. The work [XST+10] considered mobile RFID problem in the isolated constant model. In that model, there is no need to distinguish the identification deadlines of tags since the tag instances in the interrogation area do not change. The works [SDOS07] and [RTWL09] are experimental works in which showing the impact of a single system factor (e.g., tag placement, tag orientation, etc.) on the identification rate was made easier by the use of the isolated constant model. By contrast, the dynamic constant arrival model is preferred for real applications, as discussed in [SMSC08, NSL08, Imp09]. This model is more suitable for an environment with dense tag placement and high-speed moving tags. Our work also adopts this model. This model is more general so the proposed approaches also work in the isolated constant arrival model. The mobile RFID anti-collision problem in the isolated variable arrival model and the dynamic variable arrival model remains to be addressed in a future work.

3.3 Principles for Mobile RFID Anti-Collision Protocols

In this section, we propose two principles for mobile RFID anti-collision protocols: *workload optimal* and *the earliest deadline first*. We argue that following these two principles can lead to the simultaneous achievement of a high identification rate and a high tag moving speed. Probabilistic analysis is used to determine the optimal workload value. Finally, we suggest to use tag grouping technology to implement these two principles.

We begin our discussion from ALOHA-based protocols. Existing ALOHA-based protocols dynamically adapt frame size to tag cardinality. The difficulty is how to estimate tag cardinality before the adaptation. Usually the estimation is based on the information gathered from ALOHA-based protocols, and it is not accurate due to the stochastic characteristics of the protocols. In a stationary or low-speed environment, this is not a big concern since the protocol can iteratively refine the estimation when more information is gathered. However, it is unsuitable for a high-speed environment. The long latency of the estimation may lead to missed reading of tags. Here we consider the problem from a new angle to avoid this situation. We schedule an optimal number of tags to compete for the channel to guarantee the identification rate, so there is no longer any need to estimate tag cardinality.

Another important aspect of high-speed mobile RFID problem is the identification deadlines of tags. Existing protocols have not considered that the unidentified tags reaching the edge of the interrogation area should be accorded a higher priority.

Based on the above analysis, we suggest that mobile RFID anti-collision protocols follow two principles: *workload optimal* and *the earliest deadline first*. In this problem, *workload* is defined as the number of tags simultaneously competing for the channel. When a tag is moving through the interrogation area, the time point at which it leaves the interrogation area is defined as the *identification deadline* of this tag. Among the identification deadlines of all tags, the minimal one is defined as *the earliest deadline*. The workload optimal

principle is used to maintain a stable throughput. The earliest deadline first principle is to ensure that the tags near the tail of the interrogation area can be identified in time. In the following part, we use a probabilistic analysis to determine the optimal workload value for our problem.

The key point of the problem is maintaining a high identification rate. Two factors are critical for it: *single tag identification probability* (*sip* for short) and throughput. Single tag identification probability describes the requirement in view of a single tag, while throughput considers all tags as a whole. In a frame slotted ALOHA protocol, the identification process includes multiple frames. Single tag identification probability further depends on *single tag identification probability per frame* (*sipf* for short) and the number of frames.

Let us begin with single tag identification probability. Assume that in a frame slotted ALOHA protocol, an identification process has m frames and each frame has f time slots. There are n unidentified RFID tags in the interrogation area. Each tag selects a time slot randomly, hence causing the selected time slots to be uniformly distributed. Time slots are classified into *empty slots*, *singleton slots*, and *collision slots*, denoting the time slots selected by no tag, only one tag, and multiple tags, respectively. Each time slot has the same probability p to be a singleton slot:

$$p = n\frac{1}{f}(1-\frac{1}{f})^{n-1} \tag{3.1}$$

The expectation of the number of singleton slots is

$$f \cdot n\frac{1}{f}(1-\frac{1}{f})^{n-1} = n(1-\frac{1}{f})^{n-1} \tag{3.2}$$

In a frame, there are n tags involved in the identification, so the identification probability per frame for a single tag *sipf* is

$$sipf = n(1-\frac{1}{f})^{n-1}/n = (1-\frac{1}{f})^{n-1} \tag{3.3}$$

We also calculate the unsuccessful identification probability per frame for a single tag *usipf*:

$$usipf = 1 - sipf \tag{3.4}$$

The above analysis is based on a single frame. In a complete identification process, there are m frames that need to be considered. We also need to take into account the silence of identified tags. In an anti-collision protocol, if a tag is identified in a frame, it will keep silent during the following frames. If t tags are identified in one frame and another t tags participate in the next frame, *usipf* is kept equal to *1-sipf* for each unidentified tag. This is regarded as a necessary condition to achieve the maximal tag moving speed (otherwise, the tag moving speed can still be adjusted). Following the binomial distribution, we deduce the successful identification probability in m frames for a single tag, which is the single tag identification probability, *sip*:

$$sip = 1 - C_m^m usipf^m (1 - usipf)^0 = 1 - (1 - sipf)^m \tag{3.5}$$

$$= 1 - [1 - (1 - \frac{1}{f})^{n-1}]^m \tag{3.6}$$

As the optimal frame size, f is set to n to achieve the maximal throughput. We then have

$$sip = 1 - [1 - (1 - \frac{1}{n})^{n-1}]^m \tag{3.7}$$

According to Eq. 3.7, *sip* is determined by two factors: workload n and the number of frames m. Now let us correlate the interrogation area L with *sip*. In a frame slotted ALOHA protocol, before normal time slots the reader usually issues a "query" command to notify tags of some configurations (link frequency, data rate, etc.) that the identification process should be initiated [EPC07]. The tags then begin to transmit ID information to the reader, and the reader acknowledges the transmissions. Each acknowledgment serves as the beginning signal for the next time slot, with the exception of the last one. For the

sake of simplicity, we assume that all time slots are of the same length and there are *n+2* time slots in one frame[1]. The time duration of one frame, called *frame time* in this work, is $(n+2)T_0$, where T_0 is the time duration of a time slot. Given the interrogation area L, the conveyor belt speed v determines the number of frames m:

$$m = \frac{L}{(n+2) \cdot T_0 \cdot v} \tag{3.8}$$

Replacing m in Eq. 3.7 and combining the requirement of a high identification rate, we have:

$$sip = 1 - [1 - (1 - \frac{1}{n})^{n-1}]^{\frac{L}{(n+2) \cdot T_0 \cdot v}} \geq r \tag{3.9}$$

Single tag identification probability is just one side of the problem, while the other side is throughput. The protocol should consider not only the identification of a single tag but of all tags as a whole. From the above analysis, the average throughput in one frame is

$$n(1 - \frac{1}{n})^{n-1} \text{ singleton slots/frame time} \tag{3.10}$$

To achieve the maximal tag moving speed, the average throughput should be no less than the number of tags moving out of the interrogation area every frame time. We then have the following equation:

$$n(1 - \frac{1}{n})^{n-1} \geq (1 + \beta) \cdot p \cdot v \cdot (n + 2) \cdot T_0 \tag{3.11}$$

where β ($\beta \geq 0$) is the *over-provisioning factor* [HG05] describing the resilience of this system against unexpected workload. If $\beta > 0$, in a normal situation there are more identified tags than the tags moving out. This design allows sudden serious collisions to be tolerated. The value of β is application specific. p is the tag density in the conveyor belt.

Combining Eq. 3.9 with Eq. 3.11, we complete the constraints of this problem. Given an identification rate, we can solve these two equations to get the optimal workload and the maximal tag moving speed.

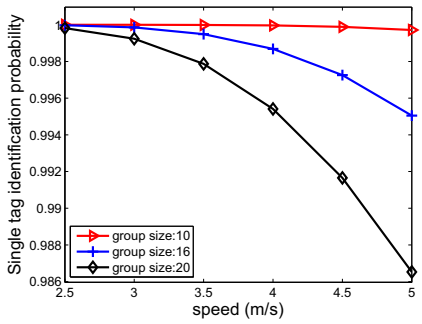


Fig. 3.4: Single tag identification probability in different speeds

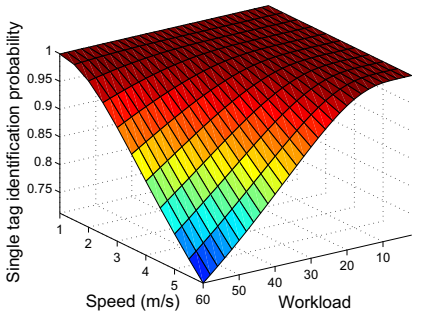


Fig. 3.5: Single tag identification probability with different workloads and speeds

We use Fig. 3.4-3.6 to show the relations between workload and speed in our problem more clearly. The parameters used to draw the figures are $L = 15m$, $d = 50cm$, $T_0 = 15ms$, and $r = 0.99999$. As shown in Fig. 3.4, when speed increases, if workload is large, *sip* decreases much more quickly than in the other cases. Fig. 3.5 shows *sip* with different workloads and speeds. Clearly, *sip* is high when workload is small or speed is low, and decreases when workload or speed increases. Fig. 3.6 shows the contours of *sip*. The throughput constraint of Eq. 3.11 is also added with different values of parameter β. The feasible solution space is below both the identification rate contour and the throughput constraint. For example, if β is 1.1, the optimal solution is about a workload of 6 tags and

[1]Readers can analyze empty slots, singleton slots, and collision slots of different lengths as in [XGWX10].

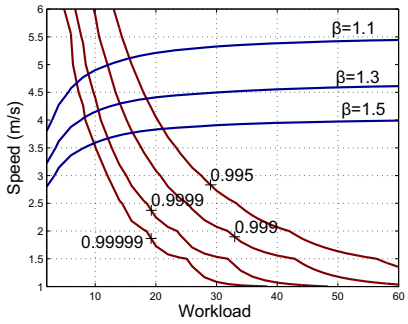


Fig. 3.6: Contours of single tag identification probability and throughput

a speed of about $4.7m/s$.

Tag grouping technology can be used to implement the proposed two principles. We organize the RFID tags in the interrogation area into groups and each time only some of the groups are allowed to compete for the channel. Corresponding to the two principles, we schedule the group with the earliest deadline to be identified first, and control the number of groups involved in the channel competition to provide proper workload. Through proper design of *group size* (the number of tags in one group) and *concurrent group number* (the number of groups involved in the channel competition at the same time), the optimal workload can be achieved. We use tag grouping technology to design our protocol.

3.4 Solution

In this section, we propose the Schedule-based Anti-Collision Protocol (SAC), which organizes the tags into groups and schedules tag identification according to the identification deadline. SAC follows the principles of workload optimal and the earliest deadline first. After a detailed explanation of SAC, we will engage in a further discussion on tag selection policy and RFID uncertainly handling in this protocol.

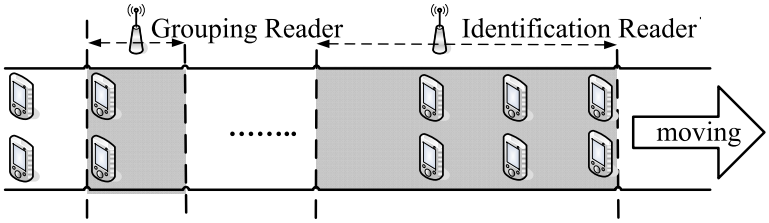


Fig. 3.7: The deployment of two readers in SAC

3.4.1 Schedule-based Anti-Collision Protocol

SAC utilizes tag grouping technology to organize tags and then schedule tag identification. Tag grouping approaches can be classified into two categories: tag control approach and reader control approach. In the tag control approach, most of the tasks are undertaken by the tag, while the protocol on the reader side is simple. In the reader control approach, the grouping is undertaken by the reader, while the protocol on the tag side is simple. The tag control approach is more suitable for active tags that have sufficient computational and power resources. In this work, we propose a reader control approach suitable for passive tags. Considering that passive tags are commonly used in many RFID applications, this approach is a practical one.

In SAC, we adopt a design of two RFID readers. One of them, called the *grouping reader*, is used to allocate group IDs to the tags. The other one, called the *identification reader*, is used to detect the tags. The deployment is shown in Fig. 3.7. The grouping reader is placed in front of the identification reader. The grouping reader's interrogation area is controlled so that only the tags of one group at a time can receive the allocated group ID. The identification reader is just set to the largest interrogation area as in a stationary environment.

Note that the grouping reader introduced here is used only for assigning group IDs to the tags, not for the purpose of identification. Therefore the grouping reader does not need

Table 3.2: Notations in SAC

Variable	**Definition**
f	frame size
frameId	ID of current frame
$s, s.id$	current time slot and its ID(from 1 to f)
N_0, N_1, N_2	the number of empty slots, singleton slots and collision slots, respectively
L	interrogation area of the identification reader
p	tag density in the conveyor belt
groupLen	the total number of groups in the conveyor belt
groupNum	the maximal number of groups which can co-exist in the interrogation area
groupSize	the optimal number of tags in a group
changeFreq *changeAmt*	system parameters to adjust the number of groups joining in the channel competition.
finish$[x]$	used to record the number of identified tags in group x
finishInFrame$[x]$	used to record the number of identified tags in frame x
curGroupId	ID of current group

to be as powerful as the identification reader in the functionalities. The interrogation area of the grouping reader can be much smaller, so the process of radio emitting/receiving can be greatly simplified. Some components in the reader, such as signal processing module and CRC verification module [EPC07] also can be tailored or eliminated. This kind of reader is much cheaper than a common reader; hence, little additional funding will be needed to make use of it. Moreover, if any prior knowledge can be directly used to group the tags, the grouping reader can be eliminated. For example, in a manufacturing application, if tag IDs are continuous and the optimal group size is eight, we can use the ID prefix (from the leftmost bit to the fourth rightmost bit) as the group's ID. For the sake of integrality of our work, we describe the algorithms of both the grouping reader and the identification reader as follows.

The details of SAC are shown from Algorithm 1 to Algorithm 3, including the operations

Algorithm 1: SAC Grouping Reader Operation

Variable : f

Function: initGroupingReader( )

1 $curGroupId = 1$
2 broadcast *curGroupId* and f using "groupIdAssign" command to assign a group id to the tags

Function: receiveGroupingReader(time slot s)

1 add N_0, N_1, N_2 correspondingly
2 **if** $s.id == f$ **then**
3 | **if** $N_1 + N_2 > 0$ **then**
4 | | *curGroupId*++
5 | **end**
6 **end**
7 goto *initGroupingReader* line 2

in the grouping reader, the identification reader, and the tag, respectively. In the grouping reader or the identification reader, there is a function for reader initialization and a function for received messages handling. The notations used in SAC are summarized in Table 3.2.

In Algorithm 1, the grouping reader increasingly allocates group IDs to the tags. In Function *initGroupingReader*, the reader broadcasts *curGroupId* and f to the tags using our defined "groupIdAssign" command. This command has the same functions as the "query" command. Having received this command, the tag sends a short message back. The content of the message is not important because the message is merely used to show whether any tag exists in the interrogation area. For the same reason, the collision caused by multiple messages sent back is also not a concern in this protocol. Both collision and singleton results show the presence of tags.

Function *receiveGroupingReader* is called at the end of each time slot s. Frame size f denotes the number of time slots in a frame. Therefore, the time slot id $s.id$ is numbered from *1* to *f*. N_0, N_1, and N_2 are used to record the number of empty slots, singleton slots, and collision slots, respectively (line 1). When a frame is finished ($s.id == f$, line 2), *curGroupId* increases by one if there are still some tags in the interrogation area ($N_1 + N_2 > 0$) (lines 3-5). On the contrary, *curGroupId* is retained and retried in the next frame if no tag is

Algorithm 2: SAC Identification Reader Operation

Input : *groupSize*, *groupLen*, L, p
Variable : *groupNum*, f, *finish*, *finishInFrame*, *frameId*, *curGroupId* *changeFreq*, *changeAmt*

Function: initIdenReader( )
1 *groupNum*=$(L \cdot p)/$*groupSize*
2 f=*groupSize*
3 *finish*$[1 :$ *groupLen*$] = 0$
4 *frameId*=1, *curGroupId*=1, *changeFreq*=2, *changeAmt*=1
5 *finishInFrame*[*frameId*]=0
6 **while** *curGroupId* $<$ *groupLen* **do**
7 | *candidateNum* = *groupSize* − *finish*[*curGroupId*]
8 | *maxGroupId* = *curGroupId*
9 | **while** *candidateNum* $<$ *groupSize* **do**
10 | | *maxGroupId*++
11 | | *candidateNum*+=*groupSize* − *finish*[*maxGroupId*]
12 | **end**
13 | **if** *candidateNum* $>$ *groupSize* && *frameId* % *changeFreq* == 1 **then**
14 | | *maxGroupId* = *maxGroupId* − *changeAmt*
15 | **end**
16 | broadcast *maxGroupId* and f using "query" command to begin the identification of a new frame
17 **end**

Function: receiveIdenReader(time slot s)
1 add N_0, N_1, N_2 correspondingly
2 **if** s is a singleton slot **then**
3 | extract *tagId* and *groupId* of the tag, store *tagId*
4 | *finish*[*groupId*]++
5 | *finishInFrame*[*frameId*]++
6 **end**
7 *curGroupId*=max(*curGroupId*, *groupId*-*groupNum*+1)
8 **while** *finish*[*curGroupId*] == *groupSize* **do**
9 | *curGroupId*++
10 **end**
11 **if** *s.id* == f **then**
12 | (optional) record *finishInFrame*[*frameId*] for further performance tuning
13 | *frameId*++
14 **end**
15 goto *initIdenReader* line 4

detected ($N_1 + N_2 = 0$).

The operations in the identification reader are shown in Algorithm 2. The protocol tracks the group with the earliest deadline and maintains the optimal workload. As the protocol input, *groupSize* is the optimal value of workload computed by Eq. 3.9 and Eq. 3.11. *groupLen* is the total number of groups in the conveyor belt if it can be known in advance.

Otherwise, *groupLen* should simply be set to a large enough value. L and p are defined in our system model.

In Function *initIdenReader*, lines 1-5 initialize some variables. First, the maximum number of groups that can coexist in the interrogation area, *groupNum*, is computed. The frame size f is set to be the optimal workload. The array *finish* is used to record the number of identified tags in each group and initially set to be 0. *frameId* and *curGrouopId* track the current frame and the group with the earliest deadline, respectively. *changeFreq* and *changeAmt* are two system parameters for adjusting the number of groups participating in the channel competition. They will be discussed later. The array *finishInFrame* is used to record the number of identified tags in each frame.

Following this is a loop from line 6 to line 17, where each iteration begins a new frame to detect the tags. In each frame, the protocol tries to involve a number of *groupSize* unidentified tags as candidates for the channel competition. These candidates are from multiple groups and *maxGroupId* denotes the maximal group ID of these groups. In lines 7-12, *maxGroupId* gradually increases to include more groups of tags as candidates. Notice that, as in line 11, the tags in a group are considered as a whole to be candidates, so that the final number of candidates may not be exactly equal to *groupSize*.

There are two policies for achieving an approximate optimal result: the *conservative policy* and the *aggressive policy*. The conservative policy rounds down *maxGroupId* so that the number of candidates is just less than *groupSize*, while the aggressive policy rounds up *maxGroupId* so that the number of candidates is just greater than *groupSize*. As in lines 13-15, these two policies are used in a cyclical way to achieve a desirable performance. The idea is that by adjusting *changeFreq* and *changeAmt*, we can control changes to *maxGroupId*. *changeFreq* controls the frequency of change. *changeAmt* controls the amount of change. We set *changeAmt* to 1, denoting that the deviation of *maxGroupId* from the optimal value

is bounded by 1. We set *changeFreq* to 2, denoting that in odd frame IDs we use the conservative policy, while in even frame IDs we use the aggressive policy.

Finally, the identification reader sends a "query" command with *maxGroupId* and *groupSize* to the tags (line 16).

Function *receiveIdenReader* is called at the end of each time slot *s*. The function records the identified tags of each group using *finish* (lines 2-6) and tracks the group with the earliest deadline using *curGroupId* (lines 7-10). Notice that at most *groupNum* groups can coexist in the interrogation area. Therefore, if one tag with the group ID of *groupId* is successfully identified, the group with the ID of $(groupId - groupNum)$ must have moved out of the interrogation area and hence does not need to be considered (line 7). Moreover, if all tags in a group are identified, *curGroupId* increases until a group that still has unidentified tags is found (lines 8-10). When a frame is at the end (line 11), the protocol increases *frameId* (line 13). The function also records the number of identified tags in the current frame for further performance tuning (line 12), which is optional. This will be discussed in later subsections. Finally, the execution of the protocol jumps to line 4 of Function *initIdenReader* to begin a new round of identification (line 15).

The tag operations are shown in Algorithm 3. When a tag receives a "query" command from the grouping reader, it records *curGroupId* as its group ID and then sends back a short message to the reader (lines 1-4). If the tag receives a "query" command from the identification reader, it checks whether it is allowed to compete for the channel (line 5). The check condition is whether the tag's *curGroupId* is less than or equal to the *maxGroupId* included in the command. If the condition holds, the tag competes for the channel in the same way as in a traditional ALOHA-based protocol (line 6).

To implement this protocol, only slight modifications are needed for standard EPC C1G2 tags [EPC07]. An additional memory is needed to record *curGroupId*, which is similar with

Algorithm 3: SAC Tag Operation

```
   Function: receiveTag()
1  if received "groupIdAssign" command (with curGroupId and f) from the grouping reader
   then
2  |  record curGroupId
3  |  reply an arbitrary short message
4  end
5  if received "query" command (with maxGroupId and f) from the identification reader
   && curGroupId ≤ maxGroupId then
6  |  reply as basic frame slotted protocol
7  end
```

the memory storing Q value (the frame size in EPC C1G2 protocol). The operations in line 2-3 of Protocol 3 have the same complexity with the operations when receiving Q value in EPC C1G2 protocol. The comparison in line 5 can be combined into the operations of "Query" and "QueryRep" commands in that protocol. Therefore, the power requirement of the new tags is similar with that of original tags.

3.4.2 Tag Selection Policy

In this and the next section, we hold further discussions about SAC. First, we consider the policy of selecting tags to compete for the channel. In this work, we called it the *Tag Selection Policy*.

We have known that workload optimal is one principle that leads to a high identification rate in mobile RFID applications. Due to the stochastic characteristics of ALOHA-based protocols, a group of tags usually needs multiple frames to detect. After the first frame, if no additional tags are replenished in the following frames, the workload will decrease, resulting in a decrease of the throughput. On the other side, if too many tags are involved in the channel competition, collisions will increase, again affecting the throughput. Therefore, SAC needs to maintain the optimal workload in each frame using a proper tag selection policy. Specifically, the preference is to always have *groupSize* tags to compete for the channel in each frame.

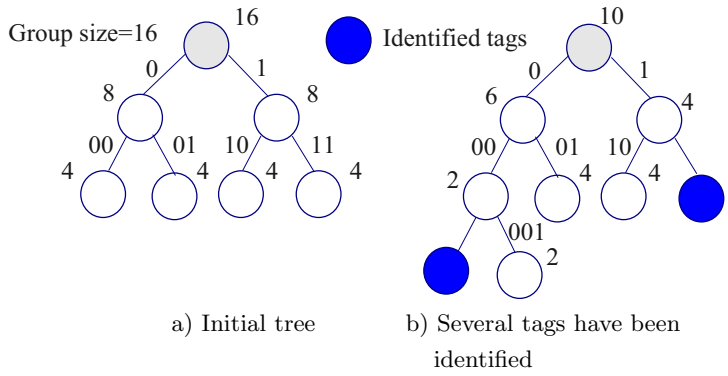


Fig. 3.8: Tree-based tag selection policy

In Section 3.4.1, the conservative policy and the aggressive policy are used in a cyclic way to achieve the optimal workload. In our scenario, the optimal group size is less than 10 and the number of tags with the group ID of *maxGroupId* is even less than that because some of the tags have already been identified. A simple cyclical approach can achieve the optimal workload in most frames.

Further consideration may be needed if the optimal group size is large in other scenarios. A fine-grained workload control is needed. A tree-based tag selection policy is introduced for this purpose. The tree-based tag selection is widely used in tree-based anti-collision protocols. Here, we modify it a little to select unidentified tags. It should be noted that here the tree-based approach is not used for identification, but only for tag selection. For the sake of simplicity, we assume that the distribution of tag IDs is continuous.

In tree-based tag selection, we generate the mask code according to the number of tags needed for the channel competition. As shown in Fig. 3.8(a), the distribution of tag IDs is described using a tree where the number on the edge denotes the tag ID prefix, and the number near the node denotes the number of tags in the sub-tree that matches the tag ID prefix. In SAC, if eight tags are needed to compete the channel, we use mask code 0 or 1 while if four tags are needed, we use a mask code of 00, 01, 10, or 11. The mask code is

broadcasted and only the tags matching the mask code can compete for the channel. We record the identified tags and re-compute the number of tags that are unidentified in each sub-tree, as shown in Fig. 3.8(b). The approach can then work correctly in the next frame.

This policy has some limitations. First, the distribution of tag IDs has to be prior knowledge. This is fine in closed-loop applications, but possibly not in other applications. Second, sometimes it is hard to generate a mask code that matches a specific number of tags. For example, as shown in Fig. 3.8(b), a tag-set with five tags cannot be found in the tree. In such a case, how to achieve a fine-grained tag selection is still an open question. When designing such a tag selection policy, one thing that needs to be kept in mind is the importance of not greatly increasing protocol complexity and message overhead, which is quite challenging. Alternatively, we can consider the deviations in workload to be a type of RFID uncertainty and address it as in the following sub-section.

3.4.3 RFID Uncertainty Handling

In RFID systems, the identification of tags is usually uncertain due to a complex RFID environment and the stochastic characteristics of ALOHA-based protocols. This problem commonly exists in all RFID anti-collision protocols and has not been well addressed yet. In this section, we discuss the impact of RFID uncertainty on our protocol.

The uncertainty in this problem arises from several aspects: 1) The irregular interrogation area of the grouping reader. It varies from time to time, hence making the number of tags in one group not exactly equal to *groupSize*. 2) The impact of environment on the identification reader. It is probable that a tag will be not identified due to environmental factors, even in the singleton slot. 3) The stochastic characteristics of ALOHA-based protocols. Serious collisions may occur occasionally, which is quite different from the process of identification in a normal situation.

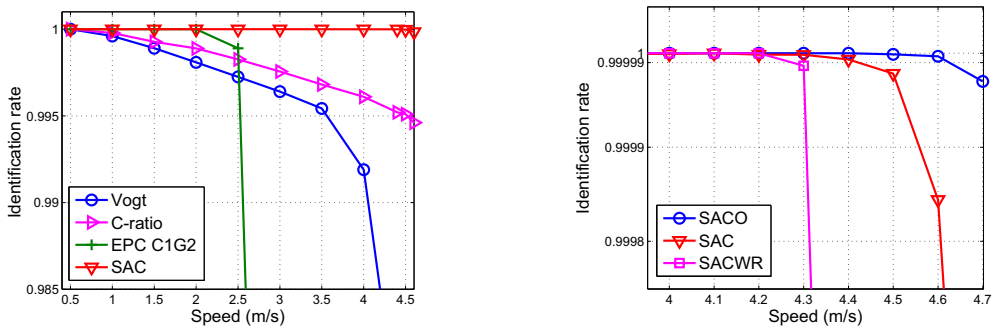
Uncertainty causes improper workloads and a serious degradation in performance. To

address this problem, we designed a feedback mechanism for the protocol. The basic idea is to adjust the workload via *maxGroupId* in Algorithm 2. If more than enough tags join in the channel competition, *maxGroupId* decreases. By contrast, if the workload is not sufficient, *maxGroupId* increases. The parameters *changeFreq* and *changeAmt* together control the change of *maxGroupId*. *changeFreq* controls the frequency of change and parameter *changeAmt* controls the amount of change (Function *initIdenReader*, lines 13-15). After the identification of a frame, we examine the identification performance as in Function *receiveIdenReader* line 12. We can then add some codes after line 12 to adjust *changeFreq* and *changeAmt*. Coming up with a detailed algorithm is beyond the scope of this work and is left for a future study. Here, we provide interested readers with a brief description of a possible method to address the problem of uncertainty

A statistical hypothesis test [Leh86] guides the adjustment of the workload. The steps are as follows. First, learn the distribution of the number of tags in one group through extensive experiments before the system really works. After the system works, a statistical hypothesis test is used to analyze whether or not the workload is maintained as expected. The null hypothesis is that the workload follows the original distribution and the alternative hypothesis is that it does not. We observe the number of identified tags in each frame (using *finishInFrame* in Algorithm 2). Based on the observations, we determine whether the null hypothesis is consistent with a specified first kind error p_1. If it is, the system changes *changeFreq* and *changeAmt* correspondingly. Otherwise, the configuration is kept as it is in the last frame, and the process of tag identification continues.

3.5 Simulations

This section presents the simulation results of SAC compared with existing approaches. We select Vogt's approach [Vog02a] and the C-ratio approach [CK05] for comparison, which



(a) The comparison of different approaches (b) The comparison of SAC, SACWR and SACO

Fig. 3.9: The comparison in the identification rate against tag moving speed

have reportedly produced a better performance in mobile environments than other existing approaches [WLZ+07]. We also compare our approach with the EPC Class 1 Generation 2 protocol (EPC C1G2) [EPC07], which is an ISO standard and widely used in the industry. SAC without tag replenishment (SACWR for short) and SAC optimal (SACO for short) are included to justify our protocol design. For SAC optimal, the number of tags competing for the channel is equal to the optimal workload in each frame. The simulation follows the scenario as shown in Fig. 3.1. The system parameters are set as follows: $L = 15m$, $d = 50cm$, $T_0 = 15ms$, $r = 0.99999$. 12000 tags are put on the conveyor belt. The over-provisioning factor β is set to be 1.2. Thus, according to Eq. 3.9 and 3.11, we get the optimal workload (also the optimal group size) of 8. In the simulation, 1000 rounds are run to obtain each result. The confidence level is 0.975.

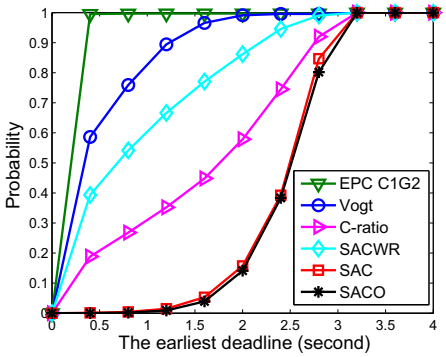
The results reveal four points that should be noted. First, existing approaches do not work well in high-speed mobile environments. SAC exhibits a significant increase in speed of up to 120%, given an identification rate of 99.999%. Second, SAC has better performance due to the control exerted over the earliest deadline and workload. Third, the optimal group size is critical to a high identification rate. Fourth, the concurrent group number can be used to measure the earliest deadline online.

3.5.1 Identification Rate

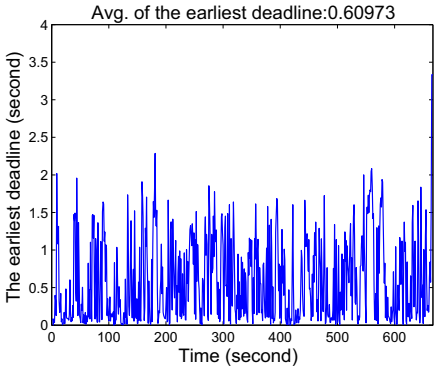
We compare the identification rates of the aforementioned approaches in different speeds. The results are shown in Fig. 3.9.

The results of the EPC C1G2 protocol, Vogt's approach, the C-ratio approach, and the SAC protocol are compared in Fig. 3.9(a). The EPC C1G2 protocol is a time-slot based algorithm, in which the frame size can be changed in the time-slot level. The frame size is increased or decreased if empty slots or collision slots are observed, respectively. By contrast, Vogt's approach and the C-ratio approach are frame based approaches in which the frame size is adjusted only at the end of a frame. In a low-speed environment ($< 2m/s$), the EPC C1G2 protocol converges to a desirable frame size more quickly, and so has a better identification rate, than Vogt's approach and the C-ratio approach. However, when the speed is greater than $2m/s$, the identification rate of the EPC C1G2 protocol is greatly affected. This is because compared with the frame based approach, the frame size changes more frequently in the time-slot based approach and more transition time is needed. This causes a decrease in identification time, and some tags may not be identified before moving out of the interrogation area. Among these four approaches, SAC has the best performance in a high-speed environment. Given an identification rate of 99.999%, Vogt's approach, the C-ratio approach, and the EPC C1G2 protocol only support speeds of less than $2m/s$, while SAC can support a speed of $4.4m/s$. From $2m/s$ to $4.4m/s$, SAC achieves a significant improvement in speed of up to 120%.

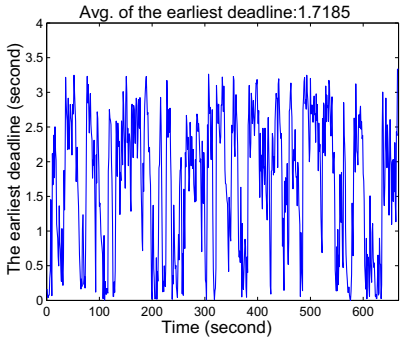
The performance of SACO, SAC and SACWR are compared in Fig. 3.9(b) to validate the rationality of our design. SACWR's performance is always no better than that of SACO and SAC. The maximal speed it can support is about $4.2m/s$ and the identification rate decreases dramatically with higher speed. This is because SACWR does not properly control the throughput. After identifying one group, it may not have sufficient time to



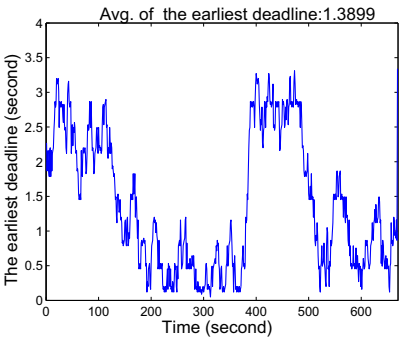
(a) CDF of the earliest deadline



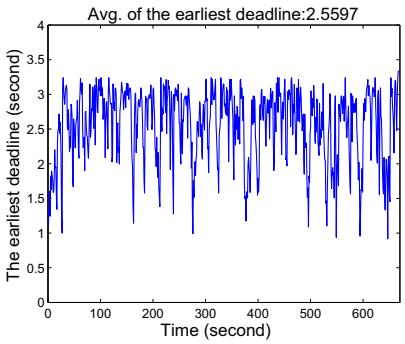
(b) Vogt



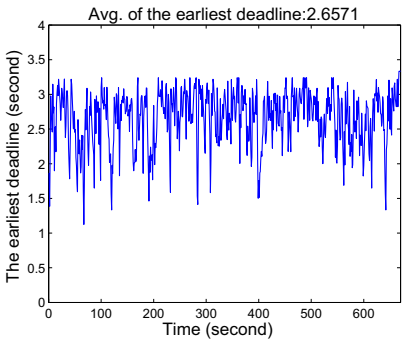
(c) C-ratio



(d) SACWR



(e) SAC



(f) SACO

Fig. 3.10: The earliest deadline in different approaches. a) cumulative distribution function of the earliest deadline in different approaches; b)-f) identification samples of different approaches

finish identifying the next group. SACO and SAC have quite similar performance, with a maximal speed of $4.6m/s$ and $4.4m/s$, respectively. This shows that the cyclical use of the conservative policy and the aggressive policy in SAC is effective.

3.5.2 The Earliest Deadline and Throughput

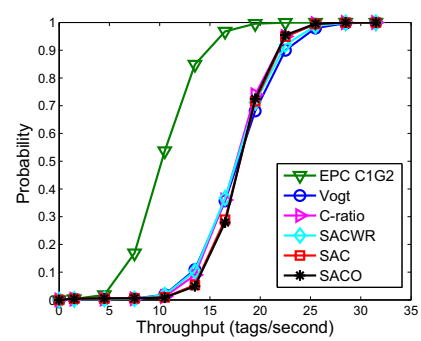
In this section, the earliest deadline and throughput in different approaches are examined. At any time, if a reader takes a longer time than the earliest deadline to identify a tag, the tags with the earliest deadline will be missed. The anti-collision protocol should keep the earliest deadline as long as possible. A protocol offering a longer earliest deadline can support a higher tag moving speed. Throughput is also studied here, as another aspect of identification rate. Simulations are carried out at a speed of $4.4m/s$. The results are shown in Fig. 3.10.

The cumulative distribution function (CDF) of the earliest deadline in different approaches is shown in Fig. 3.10(a). According to the figure, at almost any time the earliest deadline in the EPC C1G2 protocol is less than 0.5 seconds. 40% and 90% cases of the earliest deadline in Vogt's approach and the C-ratio approach, respectively, are less than 1.5 seconds. By contrast, there is less than 5% cases of SAC where the earliest deadline is less than 1.5 seconds. Therefore, on average, SAC has more time to read each tag. This is because SAC properly schedules the reading of tags. A higher priority is given to the tags that reach the edge of the interrogation area. We also can see that the performance of SAC and SACO is quite close. SARWR underperforms SAC, SACO, and the C-ratio approach. This underperformance is because SARWR does not optimize the workload.

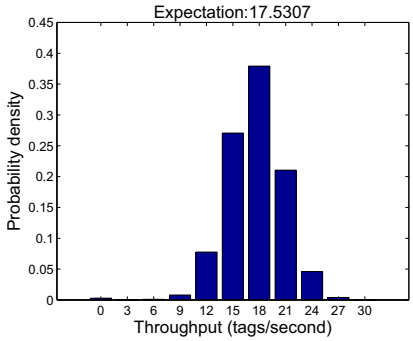
For a clearer view of the earliest deadline in different approaches, we show some samples of the identification performance in Fig. 3.10(b) to 3.10(f). A sample of the EPC C1G2 protocol is not shown here because its value is almost always 0. As shown in Fig. 3.10(b), it is clear that most cases of the earliest deadline in Vogt's approach is near the value of 0. As shown in Fig. 3.10(c), the average earliest deadline in the C-ratio approach is greatly improved compared with that in Vogt's approach. However, there are still many cases in which the earliest deadline is near 0. This means that many tags are likely to be lost in

the identification. SACWR's result is shown in Fig. 3.10(d). The average earliest deadline of SACWR is greater than that of Vogt's approach, but less than that of C-ratio approach. The problem with SACWR is that it does not properly control the workload. If a group needs more than the normal time to be identified, the earliest deadline in the following groups will gradually decrease, and finally the tags cannot be identified in time. SACWR's performance quite depends on the stochastic characteristics of ALOHA-based protocols. As shown in the figure, its earliest deadline dramatically changes from $0s$ to $3.5s$. Among Vogt's approach, the C-ratio approach, and the SAC approach, SAC has the best performance, as shown in Fig. 3.10(e). In SAC, a stable high earliest deadline is maintained and the average earliest deadline is further increased to 2.5597, which is 1.5 times and 4.2 times that of the C-ratio approach and Vogt's approach, respectively. The earliest deadline of SAC is quite close to the one of SACO, which is shown in Fig. 3.10(f).

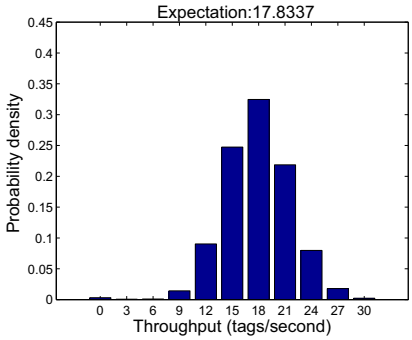
The throughput is further evidence of the advantages of SAC. The results are shown in Fig. 3.11. Fig. 3.11(a) shows the cumulative distribution function of the throughput in different approaches. We can see that the EPC C1G2 protocol has the worst throughput among these approaches in a high-speed environment. Again, this is because the EPC C1G2 protocol is quite aggressive in adjusting the frame size, hence increasing the transition overhead and decreasing the whole identification time. The throughputs of other methods only have slight differences from each other. Fig. 3.11(b) to Fig. 3.11(f) show the probability density of the average throughput in different approaches. We can see that the throughput of the EPC C1G2 protocol is around 9 tags/sec, which is smaller than the 18 tags/sec of other approaches. The expectation of the throughput is 17.5307 for Vogt's approach, 17.8337 for the C-ratio approach, 10.3875 for the EPC C1G2 protocol, 17.6454 for SACWR and 17.91 for SAC. According to these results, SAC has the largest expectation of the throughput, which means that SAC is more likely than the other approaches to identify all



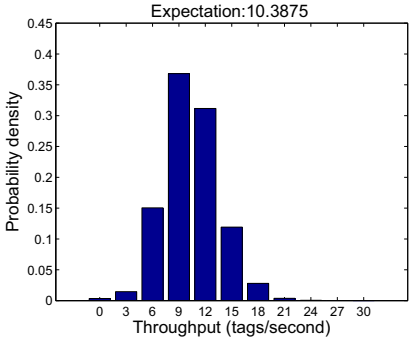
(a) CDF of throughput



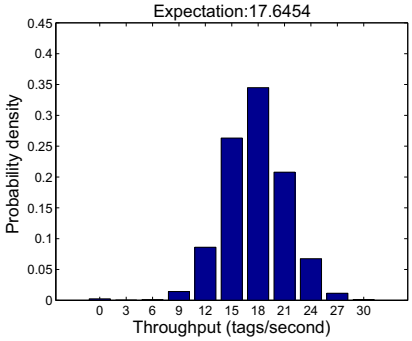
(b) Vogt



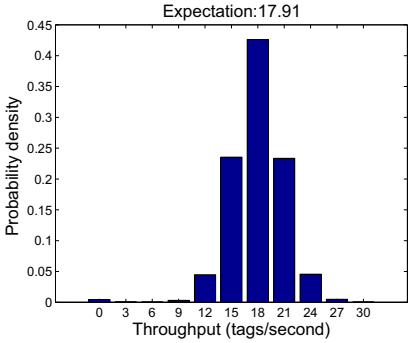
(c) C-ratio



(d) EPC C1G2



(e) SACWR



(f) SAC

Fig. 3.11: Throughput in different approaches. a) cumulative distribution function of the throughput in different approaches; b)-f) probability density of the throughput in different approaches

RFID tags successfully.

3.5.3 Group Size in SAC

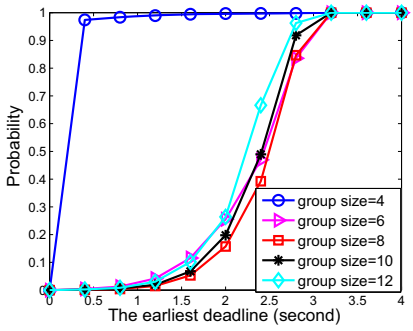
In this section, we analyze the impact of group size on SAC. We change the group size from 4 to 12 to examine the change in performance. Throughput and the earliest deadline

are used as performance metrics.

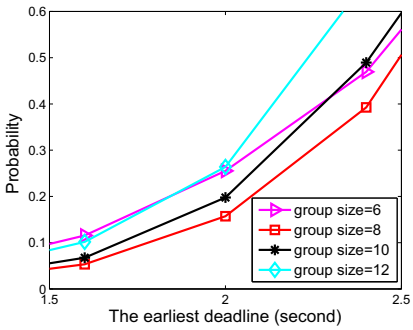
The results for the earliest deadline are shown in Fig. 3.12, where the speed is $4.4m/s$. Fig. 3.12(a) shows the cumulative distribution function of the earliest deadline in SAC with different group sizes. Fig. 3.12(b) shows an enlargement of Fig. 3.12(a) when the earliest deadline is between $1.5s$ and $2.5s$. From these two figures, we can see that SAC with a group size of 8 has the best performance in terms of the earliest deadline. The earliest deadline when the group size is 8 is larger than in the other cases. If the group size increases or decreases from 8, the probability of having small values of the earliest deadline is greater. In the figures, we can see the curves for these other group sizes are always above the curve for ($group\ size = 8$). Fig. 3.12(c) shows the change in the average earliest deadline with the group size. Clearly, the largest average earliest deadline is achieved at the group size of 8.

Similar to the discussion in Section 3.5.2, we give some samples of SAC with different group sizes. The figures are shown from 3.12(d) to 3.12(f). The group size is critical to the earliest deadline since when the group size is small (group size < 8, as shown in Fig. 3.12(d) or large (group size > 8, as shown in Fig. 3.12(f)), the earliest deadline decreases from the optimal value (group size = 8, as shown in Fig. 3.12(e)). The group size needs to be carefully selected.

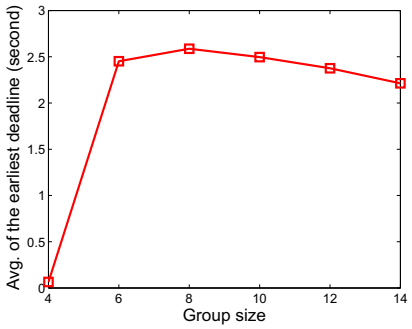
The results for throughput are shown in Fig. 3.13. According to the cumulative distribution function shown in Fig. 3.13(a), the throughputs corresponding to different group sizes are quite similar. That is because SAC can control the number of tags competing for the channel in a fine-grained way. One exception is the group size of 4, which deviates too much from the optimal value and hence introduces additional transition time. The probability density of the average throughput when the group size is 4 and 8 are shown in Fig. 3.13(b) and Fig. 3.13(c), respectively. The figures for other group sizes are similar to those for a group size of 8 and are therefore omitted.



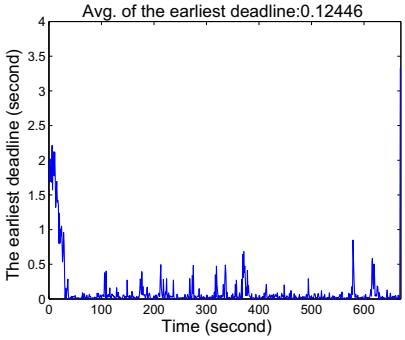
(a) CDF of the earliest deadline of SAC



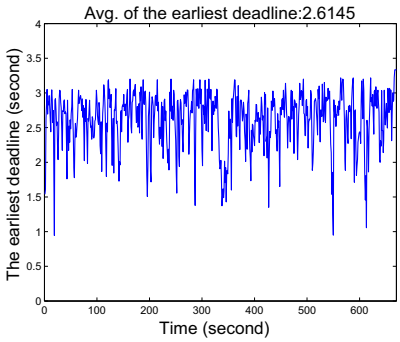
(b) CDF of the earliest deadline of SAC (closer view)



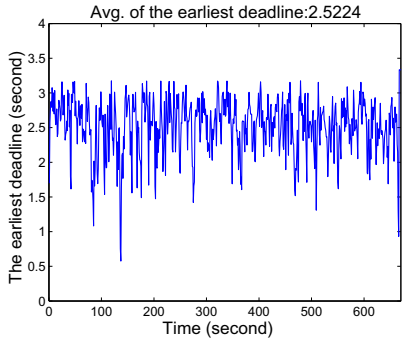
(c) Average earliest deadline vs. group size



(d) SAC (group size=4)

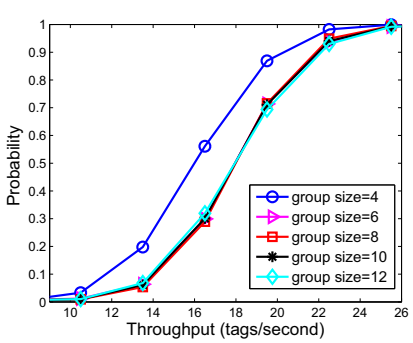


(e) SAC (group size=8)

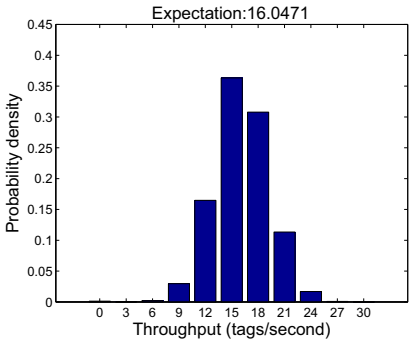


(f) SAC (group size=10)

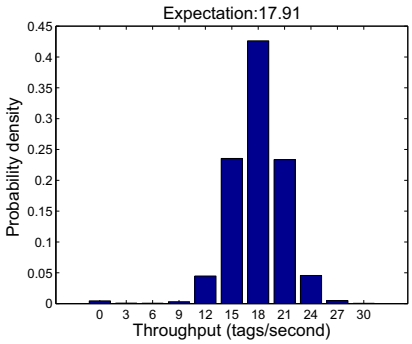
Fig. 3.12: The earliest deadline in SAC with different group sizes. a)-b) cumulative distribution function of the earliest deadline in SAC with different group sizes; c) the change of average earliest deadline with group size; d)-f) identification samples of SAC with different group sizes



(a) CDF of the throughput of SAC



(b) SAC (group size=4)



(c) SAC (group size=8)

Fig. 3.13: Throughput in SAC with different group sizes. a) cumulative distribution function of the throughput in SAC with different group sizes; b)-c) probability density of the throughput in SAC with different group sizes

3.5.4 Concurrent Group Number and the Earliest Deadline

Although in the simulation, we have the global knowledge so we can easily calculate the earliest deadline, how to do so in real applications is unclear. We observed that *concurrent group number* (the number of groups involved in the channel competition simultaneously) can be used to measure the earliest deadline. For convenience of comparison, we define *the nearest distance* to describe the earliest deadline. The nearest distance is the minimal physical distance from any unidentified tag to the tail of the interrogation area, which is a one-one mapping to the earliest deadline given a constant speed.

We go into the frame level to examine the relation between the concurrent group number

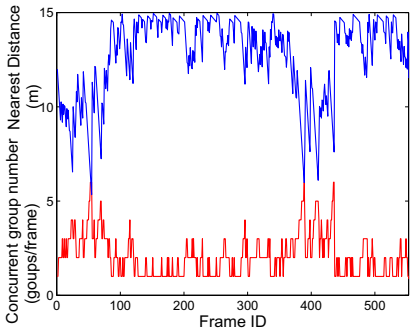


Fig. 3.14: The relations between concurrent group number and the nearest distance/the earliest deadline

and the nearest distance. Fig. 3.14 shows a result of SAC at a speed of $4.4m/s$, which clearly denotes the relation between the two measures. When the concurrent group number increases, the nearest distance decreases at the same time, and vice versa. We repeated the simulations and always obtained similar results. An RFID application can monitor the concurrent group number online to estimate the nearest distance and then the earliest deadline as well.

3.6 Experiment Results

We conducted experiments at Shanghai RFID Test Center to further compare the performance of SAC with the commercially used EPC C1G2 protocol. The results are shown in Fig. 3.15.

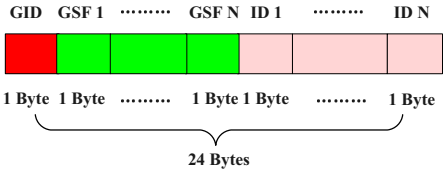
A small conveyor belt test platform was used for the experiments. The experiment configuration is shown in Fig. 3.15(a). The platform has two straight tracks each of which has a length of $160cm$ and two circular tracks each of which has a diameter of $50cm$. The speed of it can be adjusted between $0m/s$ to $1.25m/s$. A number of 120 Alien HD900AL-09 UHF RFID tags (size: $2.25cm \times 2.25cm$) were attached to 8 cartons (size: $19.2cm \times$

$13.2cm \times 10.7cm$). Each carton had a 3×5 grid of tags. An Alien 9900+ UHF RFID reader and two Impinj IPJ-A0311-USA Threshold-FS Antennas were set up in the platform. The interrogation area along the track was tuned to $1m$. In that configuration, the optimal group size was 12 and the maximum speed was $0.5m/s$ given an identification rate of 0.97, according to Eq. 3.9 and 3.11. In the experiments, the tags passed the interrogation area 100 times to have the average result.

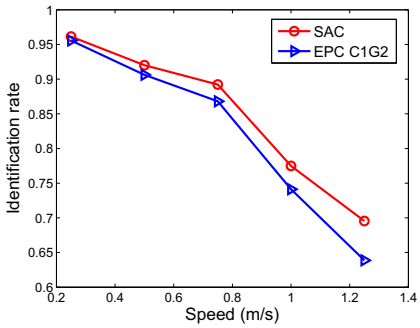
Due to limited hardware, we cannot directly modify the air interface between the tag and reader, which is also the main reason that most of anti-collision protocols are evaluated using only simulations. To overcome this difficulty, we designed a novel approach to simulate SAC's execution utilizing the "mask" operation in EPC C1G2 protocol. The "mask" operation is used to select proper tags at the beginning of an RFID identification process, which is somehow similar to SAC, which selects proper groups of tags at each frame. Before the experiments, we encoded the tags using specially designed codes, and during the experiments, we used multiple identification processes (each process executed one frame's reading) to simulate the multiple frames of SAC. The code format in tags is shown in Fig. 3.15(b). Group ID and tag ID in a group were assigned to the tags according to their physical locations, which replaced the work of grouping reader (see Section 3.4.1). For the group selection flag i, the tags of group i, $i+1$, $i+2$, $i+3$ were assigned the values of "1111", "0111", "0011", and "0001", respectively. This flag was set to "0000" for other groups. When only group i was needed, the groups with "1" in the first bit of this flag were selected, and when group i and $i+1$ were needed, the second bit was used, and so on. During each identification process, some masks were also generated to exclude the tags that had already been read.

It is noticed that the performance of this is worse than the real implementation since the transition overhead in multiple identification process is larger than that in one identification

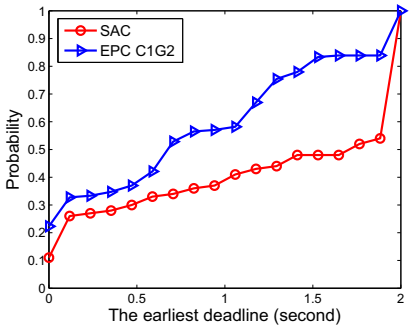
(a) Experiment configuration



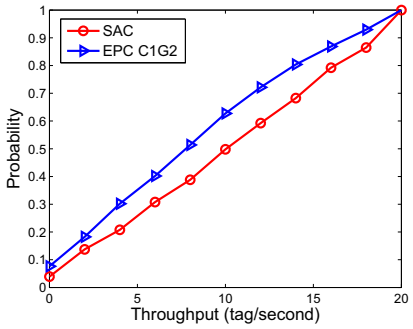
(b) The code format in tags



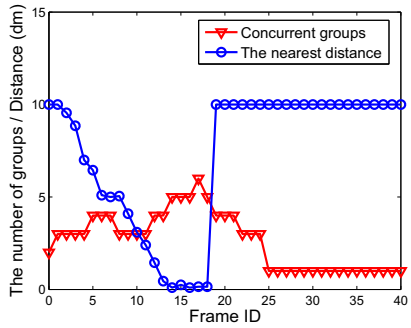
(c) Identification rate vs. speed



(d) CDF of the earliest deadline



(e) CDF of throughput



(f) the number of concurrent groups vs. the earliest distance

Fig. 3.15: The experimental comparison of SAC and EPC C1G2 protocol

process. Anyway, with this design, it is possible for us to compare the performance of SAC with that of EPC C1G2 protocol.

In Fig. 3.15(c), it can be seen that the identification rate of SAC is always better than that of EPC C1G2 protocol, and the improvement is more obvious when the speed increases.

This is consistent with the simulation results in Section 3.5.1. The earliest deadline and throughput of SAC are further checked in a speed of 0.5m/s. As shown in Fig. 3.15(d), there are more than 80% cases of the earliest deadline in EPC C1G2 protocol are less than 1.5 seconds while there are only 48% cases in SAC are less than that. This improvement is because SAC schedules the identification of tags according to their different deadlines. The throughput of SAC also outperforms that of EPC C1G2 protocol as shown in Fig. 3.15(e). This is because SAC controls the workload to an optimal value. Finally, we get into an identification result of SAC in Fig. 3.15(f) (the speed is 0.5m/s) to show that the number of concurrent groups increases/decreases when the earliest distance decreases/increases. Therefore the number of concurrent groups can be used to measure the earliest distance (and then the earliest deadline) effectively, which is consistent with the discussion in Section 3.5.4.

3.7 Summary

In this chapter, we studied the mobile RFID problem to achieve the maximal tag moving speed while maintaining a high identification rate. It is a challenging task considering the tags are moving and thus timely identification is required. We began our work from categorizing the arrival models of tags, according to the combinations of constant/variable arrival and dynamic/isolated arrival. Our work focused on the dynamic constant arrival model, which is more suitable for an industry environment with dense tag placement and high-speed moving tags. Two principles were proposed for mobile RFID anti-collision protocols: workload optimal and the earliest deadline first. Following these principles, we proposed the Schedule-based Anti-Collision Algorithm (SAC) to meet our requirement. Our work is different from existing works, which focus on how to adjust frame size dynamically according to estimates of tag cardinality. Instead, we control the workload and distinguish the identification deadlines of tags. There is no need for our protocol to estimate tag cardinality.

Simulation results show that SAC can increase the moving speed of tags by 120% compared with existing approaches, given an identification rate of 99.999%.

Chapter 4

Energy-efficient Composite Event Aggregation in WSNs Considering Complex Relations

In this chapter, we investigate energy-efficient composite event aggregation in WSNs considering complex relations. We propose an energy-efficient event aggregation tree for this purpose, based on generic composite event definitions that can have arbitrarily complex relations. This chapter is organized as follows: Section 4.1 is the overview of this work. Section 4.2 describes the system model and the problem formulation. Following this is the theoretical analysis of event aggregation in Section 4.3. Section 4.4 and Section 4.5 propose the centralized and distributed algorithms to solve this problem, respectively. Section 4.6 reports the simulation results, and finally Section 4.7 concludes this chapter.

4.1 Overview

Due to complex user requirements, an event to be detected in a WSN application is often a composite event that consists of multiple correlated sub-events. A generic event relation is defined as a function mapping from sub-events to a composite event. It describes the conditions needed to comply with to aggregate the sub-events into the composite event, and also the reduction of the amount of data during the aggregation. The relations are

specified by the user and can be quite complex, making event detection a challenging task.

In a typical process of composite event detection in a WSN, the user specifies an event to be detected, and then the information about the event is disseminated to the sensor nodes for monitoring and detection. A routing tree is usually built for efficient information transmission among the sensor nodes to collaboratively detect the composite event. Due to limited power supply of sensor nodes, energy efficiency of the routing tree is very important.

Most of the existing works on composite event detection [KRJ05, ZGC+09, LAV+10, LCF11] focus on utilizing the encounter opportunities of sub-events in the routing tree to aggregate sub-events into composite events. Event relations especially complex event relations are not explicitly used to optimize the structure of the routing tree to save energy. The data aggregation approaches [CBLV04, GE05, LLD06, ZVPS08] can be used to build energy-efficient routing trees for composite event detection, but only support the event relations derived from data redundancy. These relations have a common feature that the amount of data of a composite event is no less than that of any its sub-event. This does not necessary hold in a generic form of event relation. The support of generic relations is critical to not only the specification of complex user requirements, but also the power of a WSN in practical event detection applications. To overcome these weaknesses, we propose a new kind of routing tree for energy-efficient composite event detection, in which generic event relations are supported and fully utilized. Low-level events are aggregated when data about the events are transmitted in this routing tree, and only the deduced high-level events are sent to the applications. Thus the amount of data to be transmitted can be reduced to save energy. We call this kind of routing tree *event aggregation tree*.

In this chapter, we aim to detect composite events in an energy-efficient way. Our major work is to build an energy-efficient event aggregation tree for the event detection, based on generic composite event definitions that can have arbitrarily complex relations. We

first compare event aggregation with the approach of data aggregation, and then propose the principles of designing an event aggregation tree. After that, both centralized and distributed algorithms are proposed to build such a tree. Extensive simulations have been taken to validate the effectiveness of the proposed algorithms. In summary, this chapter makes the following specific contributions:

- We propose event aggregation tree that utilizes complex event relations to save energy for composite event detection in WSNs.
- We propose principles of designing an event aggregation tree, based on the comparison between data aggregation and event aggregation.
- We propose centralized and distributed algorithms to build event aggregation tree to achieve energy-efficient composite event detection. Simulation results show that they outperform existing approaches and save a significant amount of energy.

4.2 System Model and Problem Formulation

We first describe the system models used in this work, which include event definition tree and event aggregation tree. Then we formulate the problem.

4.2.1 Event Definition Tree

Event Definition Tree (EDT) is used to define events of interests in this work, which is compatible with event specification languages [GJS92, CM94, GD94, Luc02] and can support generic form of relations. An example of EDT is shown in Fig. 4.1. A typical data structure of EDT node includes event ID, sub-event list, relation, event description, and data amount.

A tree denotes an event. The ID of tree root corresponds to the event ID. This definition is applied to all sub-trees iteratively. In this example, A denotes the composite event while B

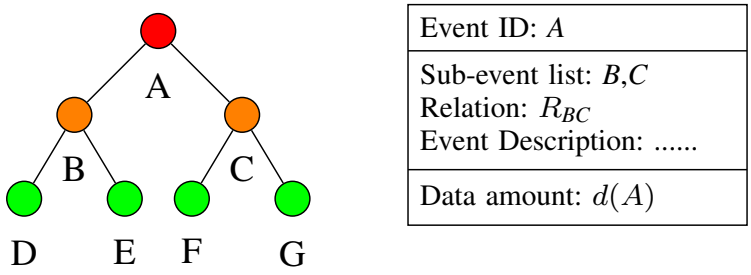


Fig. 4.1: An example of event definition tree

and C denote the sub-events of A. According to the relations of child nodes and parent nodes in the tree, corresponding events are called *child events* and *parent events*, respectively. In this example, B and C are the child events of A, denoted as $child(A) = \{B, C\}$. A is the parent event of B, denoted as $parent(B) = A$.

The relation included in an event is defined as a function mapping from child events to the parent event. Given event A with two child events B and C, the relation is referred to as R_{BC} and defined as $R_{BC} = \{B, C\} \rightarrow A$. The function is user-specific. Specially, if considering only the data amount change from child events to the parent event, the function is called *aggregation function*. Finally, EDT also defines the data amount of events. $d(A)$ denotes the data amount of event A where $d(.)$ is a function mapping from an event type to a data amount value.

4.2.2 Event Aggregation Tree

After specifying an event using an EDT, we need to determine which sensor nodes are required for the event detection and how to connect these sensor nodes through a routing structure. For energy efficiency, the routing structure is usually a tree. In this work, we fully utilize the event relation information to build an energy-efficient routing tree and call it Event Aggregation Tree (EAT). Fig. 4.2 shows one possible EAT corresponding to the EDT shown in Fig. 4.1. The annotated event IDs denote the events the sensor node detects.

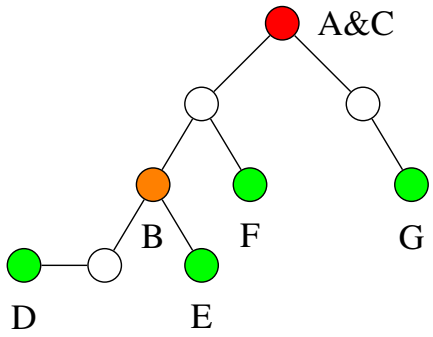


Fig. 4.2: An event aggregation tree corresponding to the event definition tree shown in Fig. 4.1.

The primitive events (i.e. event D, E, F, G) are directly detected by individual sensor nodes. So it is straightforward to determine corresponding sensor nodes in EAT, which are called *source nodes*. The composite events (i.e. event A, B, C) are more complex and can only be detected by aggregating the detection results of source nodes at intermediate sensor nodes. Therefore, additional sensor nodes other than source nodes are involved in the EAT to facilitate the aggregation. In an EAT, a sensor node forwards its detection results to its parent node, and the parent node determines whether a corresponding composite event happens based on event definition. The process that low-level events aggregate into high-level events is called *event aggregation*.

Let EAT be a tree $T(V, E)$ where node set V denotes sensor nodes and edge set E denotes communication links. r is the root node. Each edge $e \in E$ has a weight $l(e)$ denoting the transmission distance between the two nodes that are incident to edge e. The data amount of node $v \in V$ is denoted by $d(v)$. Then the total energy consumption of the EAT for event detection is defined based on the first order radio model:

$$\sum_{v \in V \setminus \{r\}} d(v)[\delta \cdot l(v, parent(v))^{\gamma} + \varepsilon]. \tag{4.1}$$

where ε is the energy consumption per bit to run transmitter/receiver circuit, and γ and δ

are two parameters of radio transmission. The parameter setting follows [LLD06]: $\gamma = 2$, $\delta = 100pJ/bit/m^2$, $\varepsilon = 40nJ/bit$. It is noticed that this is just one metric of energy consumption. Further analysis may consider fusion cost [LLD06], sensing cost [LAV+10], etc.

An energy-efficient EAT facilitates event aggregation to reduce the data amount to be transmitted, and also reduce data transmission distance. As these two goals are contrary frequently, a tradeoff of them is needed to obtain the optimal solution. As an example, event F is supposed to aggregate with event G directly according to the EDT (Fig. 4.1), whereas their aggregation is postponed in the actual built EAT (Fig. 4.2). This is because the source nodes corresponding to event F and G are far away from each other. Although following EDT to aggregate these two events can achieve early aggregation to reduce the data amount to be transmitted, it may greatly increase the transmission distance and then the energy consumption.

4.2.3 Problem Formulation

Composite event detection is divided into two phases. In the first phase, an EAT is built in the WSN. In the second phase, based on the EAT, the composite event is detected via in-network processing, similar with data aggregation [LLD06, ZVPS08, LAV+10]. Building EAT is the key of the whole event detection. We formulate it as follows and investigate it in the rest of this work.

Given:

a. A WSN that consists of a collection of sensor nodes and a sink node t.

b. A composite event defined by an EDT, denoted by *edt*.

assume:

According to *edt*, all source nodes *srcNodes* are determined.

find an EAT connecting *srcNodes* to t such that the total energy consumption of this EAT

is minimized.

This problem is a generalization of data aggregation [CBLV04] and hence is NP-complete.

4.3 Design Rationale

In this section, based on the comparison of event aggregation and data aggregation, we propose principles of designing an event aggregation tree. After that, we illustrate how to revise a data aggregation approach to build such a tree.

4.3.1 Data Aggregation and Event Aggregation

Event aggregation undertaken in EATs is different from the comprehensively investigated *data aggregation* [CBLV04, GE05, LLD06, ZVPS08]. We first distinguish these two techniques, which is helpful for us to understand our problem.

Data aggregation eliminates redundant information based on the analysis of data itself. For example, when monitoring the temperature of an area via a set of sensors, the results in the overlapped sensing areas can be aggregated due to redundancy. In event aggregation, the information is redundant with respect to relations. For example, the user may want to know the maximum temperature of an area [MFHH02]. The temperature values at different places (i.e. primitive events) are not redundant considering the temperature values themselves. However, there exists an implicit relation specified by "maximum temperature": the larger temperature subsumes the smaller temperature. In this context, some primitive temperature events are redundant and can be eliminated. Relations can be simple relations or complex relations. The relation of above example is simple and called full aggregation, where every two primitive events can aggregate into a composite event. In general, the relation can be arbitrary complex, where sub-events need to comply with special rules to aggregate into a composite event. It is noticed that the processing of event aggregation based on simple relations (e.g. full aggregation) is quite similar with that of data aggregation, and hence

existing works [CBLV04, GE05, LLD06, ZVPS08] do not distinguish them. It is necessary to clarify these two concepts, especially for the event aggregation based on complex relations that are arbitrarily specified by the user. This is to say, a generic event aggregation needs to support complex event definition.

In a specific aspect, event aggregation differs from data aggregation in aggregation function. In existing data aggregation works [CBLV04, GE05, LLD06, ZVPS08], the aggregation function is no-decreasing. It means that given two data with an amount of d_1 and d_2, respectively, the data amount of the aggregation result is greater than $\max(d_1, d_2)$. This does not hold in event aggregation. Complex relations introduce semantics from the user. A large number of primitive events may aggregate into a composite event with a small data amount. This makes difficult to use existing data aggregation algorithms for event aggregation.

According to above discussion, two principles need to be followed to achieve event aggregation and then event aggregation tree: one is to support complex event definition, and the other is to support generic aggregation function.

4.3.2 Revising Data Aggregation into Event Aggregation

Following the principles proposed in the last sub-section, we look into a specific data aggregation approach, MFST [LLD06], and try to revise it to build the event aggregation tree. We choose this approach since it has a good approximation ratio ($\frac{5}{4}\log(n+1)$ where n is the number of source nodes) to the optimal data aggregation solution. It is a multi-round 3-step process as follows:

1) Source nodes as well as the sink node are paired up. For each pair of nodes, a node is randomly selected as the *transmitter* and the other node as the *receiver* (for the node pair including the sink node, the sink node acts as the receiver). The transmitter transmits the data to the receiver through the shortest path, and then does not participate in the algorithm execution in further rounds. The pairing up process aims to minimize total

transmission cost of node pairs.

2) The receiver conducts aggregation once receiving data. The aggregation is based on a non-decreasing aggregation function.

3) Repeat step 1 and 2 until only the sink node is left.

This approach does not support complex event definition. To overcome this drawback, we revise the approach to distinguish relations of events. In step 2, the aggregation should be based on EDT, allowing the events to aggregate only if they satisfy specific aggregation conditions.

The other issue is the aggregation function. In an EDT, each node is attached with an aggregation function. The nodes with a decreasing aggregation function are called *checkpoints*. If there is no checkpoint, we can directly use above revised method to achieve event aggregation. However, if some checkpoints exist, further considerations are needed. Notice that in step 1 the pairing up process only minimizes the transmission cost, so it may lose the opportunity to achieve early aggregation that can reduce the data amount to be transmitted. One reasonable method to trade off data transmission and event aggregation is: try the best to follow EDT to conduct early aggregation unless the transmission cost is larger than the benefits of the aggregation in terms of energy consumption.

The benefits of an early aggregation correspond to the opportunity loss due to not achieving early aggregation (aggregation opportunity loss for short), which can be estimated as follows. Assuming that early aggregation cannot be achieved at a checkpoint c, the potential loss is at most

$$[\Sigma_{i \in child(c)} d(i) - d(c)] \cdot [\delta \cdot l(c,t)^{\gamma} + h(c,t) \cdot \varepsilon] \tag{4.2}$$

where $l(c,t)$ denotes the distance between c and the sink node t, $h(c,t)$ denotes the number of hops between c and the sink node t when achieving $l(c,t)$, and δ, γ and ε follow our model in Section 4.2.2. The problem followed is how to compute $l(c,t)$ and $h(c,t)$, which needs to

know where event c is detected. As a composite event, event c is detected at an intermediate node but its exact location is difficult to predicate. We compute its approximate place as the weighted centroid of its child nodes. This catches the essences of early aggregation that when all child nodes transmit their events to this place, the total energy consumption should be minimized. Since the transmitter is randomly selected (step 1), this approximation is reasonable. As an iterative definition, all places of composite events can be computed from the source nodes. Based on this, we can modify step 1 to take into account both transmission cost and aggregation opportunity loss.

In the following two sections, we describe the detailed algorithms following above ideas.

4.4 Centralized Algorithm

In this section, we propose a centralized Energy-efficient Event Aggregation Tree Building Algorithm (EEAT-C, where C denotes the centralized version). In this algorithm, there exists a central server connected to the sink node, which keeps the event definition and also the global information of WSN (sensor node types, sensor node locations, etc.). The event aggregation tree is computed at the central server, and then disseminated to the sensor nodes in the WSN.

4.4.1 Data Structure

There are two data structures maintained at the central server: Event Definition Tree (EDT) and Group Definition Tree (GDT).

An EDT denotes a composite event. The nodes of EDT are numbered with increasing IDs from the root node to the leaf nodes, level by level. Checkpoints are also marked in the EDT. One example of EDT is shown in Fig. 4.3. GDT is used to facilitate the building of event aggregation tree. For each checkpoint in an EDT, it prefers to have its child events to achieve early aggregation. A checkpoint and all its descendent nodes (considering

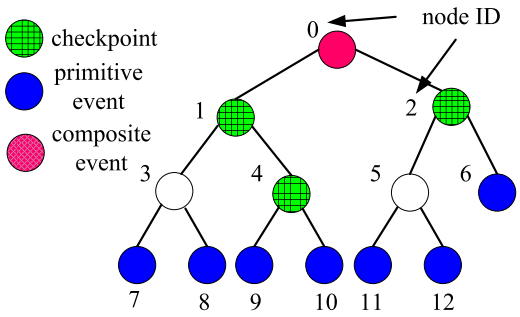


Fig. 4.3: An example of EDT kept in the central server

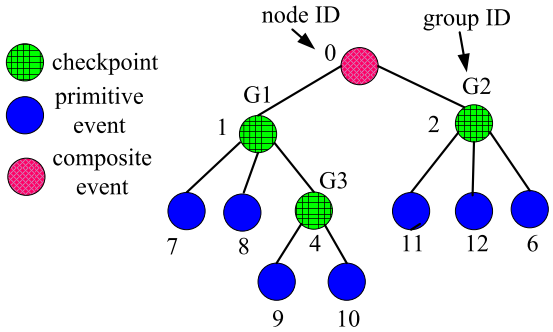


Fig. 4.4: The GDT corresponding to the EDT shown in Fig. 4.3

only source nodes and checkpoints) form a *group* for ease of processing. The checkpoint is called *group node* and represents the group. There are multiple groups due to multiple checkpoints. According to the locations of group nodes in EDT, these groups form the tree structure of GDT. Similar with EDT, the nodes of GDT are numbered. One example of GDT is shown in Fig. 4.4. We define a group as *the most matching group* of two nodes if the group node is the nearest common ancestor of those two nodes. For example, in Fig. 4.4, the most matching group of node 8 and node 9 is group G1.

4.4.2 The Algorithm

The detailed algorithms of EEAT-C are shown in Algorithm 4 and Algorithm 5. Algorithm 4 is the main algorithm, for building the energy-efficient event aggregation tree.

Algorithm 4: Build energy-efficient EAT

Input : *edt*, *t*, *srcNodes*
// *edt*: event definition tree; *t*: sink node of the WSN;
srcNodes: source nodes of the WSN
Output: E^* //edge set of EAT
Function: buildEAT(Tree *edt*, Node *t*, NodeSet *srcNodes*)

1 determine the checkpoints in *edt* and put them in *checkpointList*
2 Tree *gdt* = buildGDT(*edt*, *checkpointList*, *srcNodes*)
3 initialize round index $i = 0$, node set to be considered (in the first round) $S_0 = srcNodes \bigcup t$, and $E^* = \varnothing$, let $w_0(v)(v \in srcNodes)$ be the data amount of v, $w_0(t) = 0$
4 define the transmission cost of each node pair (u, v) $(u \in S_i,\ v \in S_i)$, which is denoted by $k_i(u, v)$: $k_i(u, v) = \alpha \cdot (w_i(u), w_i(v))c(u, v) \qquad (u \in S_i, v \in S_i)$
$k_i(u, v) = w_i(u)c(u, v) \qquad (u \in S_i, v = t)$
where $\alpha = \frac{w_i(u)w_i(v)(w_i(u)+w_i(v))}{w_i^2(u)+w_i^2(v)}$, $c(u, v)$ is the energy consumption when transmitting one unit of data between u and v
5 calculate the aggregation opportunity loss of node pair (u, v): $optl_i(u, v) = \max(u.optl, v.optl)$
6 define the total cost of node pair (u, v): $tc_i(u, v) = k_i(u, v) + optl_i(u, v)$
7 find the minimum cost perfect matching based on the total cost, let the matched node pairs be $(u_{j1}, v_{j1})....(u_{jm}, v_{jm})$ $(m = \lceil |S_i|/2 \rceil)$
8 add the edges corresponding to (u_{jp}, v_{jp}) $(1 \le p \le m)$ into E^*
9 **foreach** node pair (u_{jp}, v_{jp}) **do**
10 if $v_{jp} \neq t$ and $v_{jp} \neq t$, choose node u_{jp} as the transmitter with a probability of $\frac{w_i^2(u_{jp})}{w_i^2(u_{jp})+w_i^2(v_{jp})}$, otherwise choose t as the transmitter. The receiver is determined correspondingly.
11 the transmitter sends its events to the receiver
12 **endfch**
13 **foreach** node q having received data **do**
14 conduct event aggregation and update the data amount to $w_{i+1}(q)$
15 **endfch**
16 copy S_i to S_{i+1} excluding the transmitters
17 if $S_{i+1} = \{t\}$, stop the algorithm, otherwise $i = i + 1$, go to step 4
18 **return** E^*

Algorithm 5 is used to build the GDT.

Algorithm 5 fulfills two tasks: building the GDT structure (line 2-7), and initiating the variables of GDT nodes (line 8-14). According to the definition of GDT, the inner nodes of EDT which are neither checkpoints nor source nodes are removed so as to form the GDT (line 3-6). After that, the algorithm numbers the checkpoints in the GDT (line 8), and then calculates the aggregation opportunity loss of each inner node of the GDT when corresponding early aggregation cannot be achieved (line 9-14). Here the opportunity loss is computed in a conservative way, using the upper bound of the opportunity loss as the

result, as discussed in Section 4.3.2.

Let us now return to Algorithm 4. The algorithm first determines the checkpoints in the EDT and initializes the GDT (line 1-2). From line 3 to line 17, there is a multi-round process. In each round, the nodes pair up and transmit data from/to their partner nodes.

In line 3, some local variables are initialized, which are self-explained. From line 4 to line 6, the total cost of each node pair is calculated. The cost includes two parts. One is transmission cost (line 4), and the other is aggregation opportunity loss (line 5). Transmission cost is determined by transmitted data amount and transmission distance. Since the data transmission in this algorithm is a random process (see line 10-11), the data amount to be transmitted is calculated using the mathematic expectation (i.e. $\alpha(w(u), w(v))$). Aggregation opportunity loss is also considered as a kind of cost, aiming to promote the aggregation in the most matched groups unless the aggregation causes large overhead. In line 7, the minimum cost perfect matching of nodes is calculated based on the total cost of all node pairs. The detailed approach can be seen in [MMP00]. The perfect matching aims to minimize the sum of the total cost of the finally selected node pairs. In line 8, the edges involved in the perfect matching are added into E^* as a partial result of EAT. In line 9-12, for each node pair, we randomly choose one node as the transmitter and the other as the receiver. The rationale behind it is that a node with a larger data amount should have a lower probability to transmit data. After that, we compute the aggregated event and its data amount (line 13-15). Finally, the transmitters are removed for further processing (line 16), and the next round begins if there are still some nodes having not been considered (line 17).

4.4.3 Discussion

In each round of Algorithm 4, the size of S_i is half reduced, so the algorithm terminates after $\log(n)$ rounds where n is the number of source nodes. In each round, the basic

Algorithm 5: Build the GDT

```
   Function: buildGDT(Tree edt, List checkpointList, NodeSet srcNodes)
 1 copy edt to gdt
 2 foreach inner node i of gdt do
 3 |   if i ∉ checkpointList && i ∉ srcNodes then
 4 |   |   delete node i
 5 |   |   make the child nodes of i as the child nodes of parent(i)
 6 |   end
 7 endfch
 8 number the checkpoints in gdt
 9 for i = max(gdt.nodeIDs) to 1 do
10 |   determine the weighted centroid of child nodes of node i, say c
11 |   find the shortest path between c and the sink node t, and record the path's length l(c,t)
   |   and the number of hops h(c,t)
12 |   calculate the opportunity loss of node i according to Eq. 4.2
13 |   define the location of i as the same with the location of c
14 end
15 return gdt
```

operations include building the GDT (line 2), computing the total cost of node pairs (line 4-6), calculating the perfect matching of nodes (line 7), and conducting aggregation (line 13-15). Building the GDT and conducting aggregation are based on tree traversal, hence have the complexity of $O(n)$. Computing the total cost of node pairs has the complexity of $O(n^2)$. And the perfect matching of nodes needs $O(n^2)$ time [MMP00]. Therefore, the complexity of the whole algorithm is $O(\log(n) \cdot n^2)$. This algorithm can support generic event definition that can have arbitrarily complex relations. When restricted to certain types of events, we can have more performance results. For example, if the event has decreasing aggregation functions, the algorithm has the approximation ratio of $\frac{5}{4}\log(n)$ to the optimal energy-efficient event aggregation tree, using similar proof in [LLD06].

4.5 Distributed Algorithm

In this section, the proposed algorithm is distributed and called EEAT-D. It is suitable for the applications that want to avoid the performance bottleneck and single point failure brought by the centralized algorithm. In EEAT-D, there is no central server keeping the

global information of WSN. The EDT is disseminated to sensor nodes, and the sensor nodes autonomously determine whether to participate in the detection process and then build the aggregation tree in a distributed way.

4.5.1 Data Structure

In this algorithm, EDT is stored in sensor nodes. Based on the EDT, each related sensor node builds the GDT accordingly. It is not a complete GDT as in the centralized algorithm but restricted to the parts necessary for this node's operations.

4.5.2 The Algorithm

This algorithm is also a multi-round process. In each round, the execution is divided into two phases: one is the pairing up phase, and the other is the data transmission phase. According to the operations executed, in the pairing up phase, the nodes are classified into *initiators* and *responders*; in the data transmission phase, the nodes are classified into *transmitters* and *receivers*. The detailed algorithm is shown in Algorithm 6, and the state transition diagram in the pairing up phase is shown in Fig. 4.5. Function *initiate* and Function *respond* are executed by initiators and responders, respectively. Function *transmit* and Function *receive* are executed by transmitters and receivers, respectively. Function *notify* is executed by the sink node.

At the beginning of the paring up phase, the nodes are in "new" state. Each node receiving EDT acts as an initiator and executes Function *initiate*. It sends an "information request" message to its k-hop neighbors to query their data amount and their distances from it. Then the initiator goes into "waiting for information" state (line 1). If any response is received, the initiator puts it in local storage temporarily. After all the responses are received, the initiator goes into "pairing up reservation" state and computes the initiator-partner node pair having minimum total cost (line 2). Then the initiator sends "pairing up

Algorithm 6: Build energy-efficient EAT in a distributed way

Function: initiate(Node *initiator*)

1 *initiator* broadcasts an "information request" to its k-hop neighbors and goes into "waiting for information" state.
2 once receiving all the feedbacks from its neighbors, *initiator* goes into "pairing up reservation" state. A neighbor is selected as *partner* such that the total cost of *initiator-partner* node pair is minimized (see Algorithm 4 line 6).
3 *initiator* sends a "pairing up reservation request" to *partner*, and then goes into "waiting for reservation result" state.
4 when the request is confirmed, *initiator* goes into "ready to aggregate" state. When the request is rejected, another neighbor is chose as *partner*, achieving the second minimal total cost of node pair, and then go to step 3.
5 reach a consensus between *initiator* and *partner* that one of them acts as *transmitter* and the other acts as *receiver*.

Function: respond(Node *responder*, Node *initiator*)

1 when receiving an "information request" request, *responder* sends back its data amount and also the distance between it and *initiator*.
2 when receiving a "pairing up reservation request", *responder* stores the request locally.
3 after timeout of receiving "pairing up reservation request", *responder* confirms the *initiator* that leads to the minimal total cost, and reject the others.

Function: transmit(Node *transmitter*, Node *receiver*)

1 *transmitter* transmits its events and GDT to *receiver*.
2 *transmitter* goes to the state of "stop".

Function: receive(Node *transmitter*, Node *receiver*)

1 *receiver* receives events and GDT from *transmitter*.
2 *receiver* conducts aggregation locally.
3 change the role to *initiator* unless the node is the sink node.

Function: notify(Node *sinkNode*)

1 if all child nodes of *sinkNode* are detected, notify all the nodes in EAT to begin the event detection.

reservation request" to the optimal matched neighbor and goes into "waiting for reservation result" state (line 3). If this neighbor replies with "reservation rejected", the initiator goes back to "paring up reservation" state and try the second optimal matched neighbor (line 4). Eventually, the initiator receives a "reservation confirmed" result, then it goes into 'ready to aggregate" state (line 4). On the other side, the responder begins its processing in "ready to response" state and executes Function *respond*. If "information request" is received, the responder answers with its data amount and its distance from the initiator (line 1). If "paring up reservation request" is received, the responder records the request locally (line 2). After a timeout of receiving "paring up reservation request", the responder confirms the

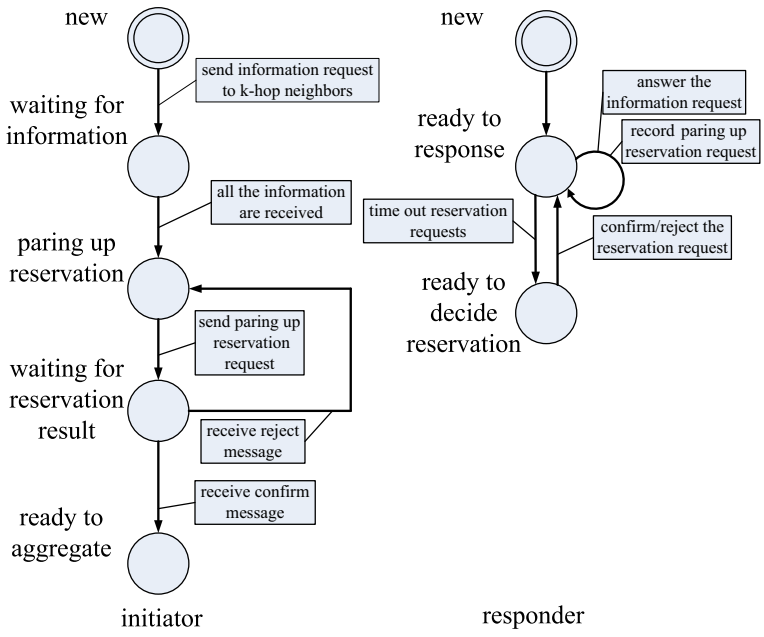


Fig. 4.5: States of source nodes in the distributed algorithm to build energy-efficient event aggregation tree

request achieving minimal total cost and rejects the others (line 3).

After the pairing up phase, the data transmission phase begins. The processing is similar with the centralized algorithm but shows distributed features. The two nodes in a node pair need to reach a consensus which sensor node is the transmitter and which sensor node is the receiver (Function *initiate*, line 5). The transmitter sends events and GDT, while the receiver receives this information and conducts event aggregation (Function *transmit* and Function *receive*).

The process of pairing up and data transmission is repeated until all the child events of the sink node are detected. In this situation, the sink node notifies the nodes in WSN that the event aggregation tree is successfully built and the event detection can be started (Function *notify*).

4.5.3 Discussion

The minimum cost perfect matching used in this algorithm is a distributed version of the approach proposed in [MMP00]. The idea in that work is to select the node pair with the minimum cost and removed them for further consideration, and then continue to select the node pair with the minimum cost, and so on. In this work, we use a reservation-confirmation process. We consider that in the distributed processing, a node may be chosen by multiple initiators as their partners. In our algorithm, each initiator first sends its partner a reservation request that may be confirmed or rejected. If it is rejected, it means that the partner is chosen by other nodes. Then the initiator needs to find another partner and repeats the process. If all the nodes finish the paring up process, the pairing up phase terminates.

In the rest of this section, we prove the correctness and termination of our algorithm.

Theorem 1. (Correctness) *The node pairs generated by Algorithm 6 correspond exactly to the result of minimum cost perfect matching approach in [MMP00] .*

Proof. This can be proved by contradiction. Assume that a "paring up reservation request" of node u is confirmed by node v but they are not a node pair generated by [MMP00]. In this case, there must exist another node w which has smaller distance from v than u. According to line 3 of Function *respond*, if such w exists, node v will reject node u's reservation request since w has a smaller distance. □

Theorem 2. (Termination) *The algorithm terminates with at most $O(log(n)\lceil n/2\rceil)$ message communication where n is the number of source nodes.*

Proof. Since nodes are paired up in a distributed way, additional message communication is needed. However, in each reservation-confirmation/rejection process (Function *initiate* line 3-4, Function *respond* line 2-3), at least the node pair that has the minimum total cost among all node pairs can be formed successfully. So through at most $\lceil n/2\rceil$ message

communication, all the node pairs are formed. After that, transmitters send events to receivers and stop (Function *transmit*), so half of the nodes are eliminated in the next round. At most $log(n)$ rounds are needed to finish the whole processing. Overall, the total message cost is $O(log(n)\lceil n/2 \rceil)$. □

4.6 Simulations

Simulations are carried out to validate the effectiveness of the proposed approaches. We compare our approaches (EEAT-C and EEAT-D) with the centralized approach (composite event detection is undertaken at a central server after all primitive events are collected), MFST (Minimum Fusion Steiner Tree) [LLD06], and TED (Type-based composite Event Detection) [LCF11] in different situations. The results reveal two points. First, our approach outperforms existing approaches in terms of energy consumption in a wide range of parameters. Second, our distributed algorithm has similar performance with its centralized version and incurs low communication overhead.

4.6.1 Simulation Setup

In the simulation, 100 sensor nodes are uniformly distributed in a $100m \times 100m$ area. The sink node is placed at the center of the area. Each sensor node has a communication range of $rc = 10m$. The parameter setting of energy consumption model is the same with Section 4.2.2. The EDT generated for the simulation is a binary tree with a depth of L=5. It means that there are 32 leaf nodes (corresponding to primitive events) and 30 intermediate nodes in the EDT. The data amount of primitive evens is randomly chosen in a range of $[400, 500]$ bits. 8 intermediate nodes are randomly chosen as checkpoints. Given the data amount of an event ap and the maximum data amount of one its child event as, *data reduction rate* is defined as ap/as. The data reduction rate of checkpoints in the simulation follows a normal distribution with the center of drr. The average distance from a

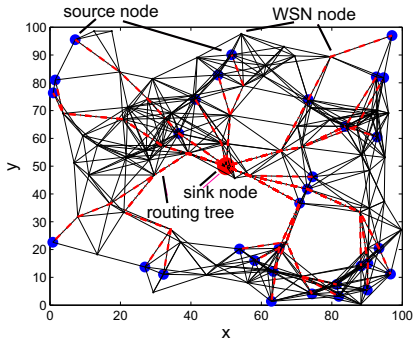


Fig. 4.6: The routing tree of the centralized approach

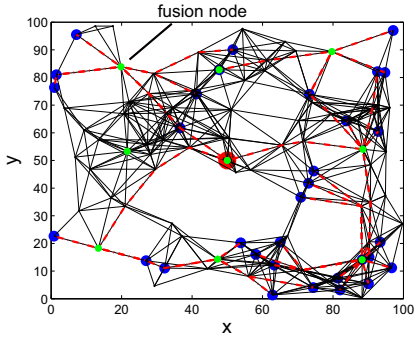


Fig. 4.7: The routing tree of TED

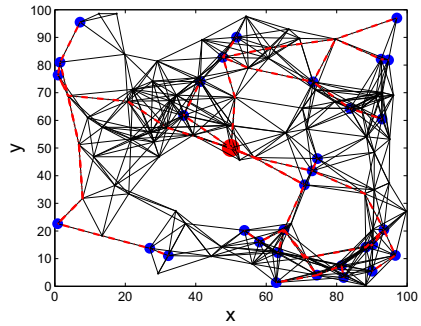


Fig. 4.8: The routing tree of MFST

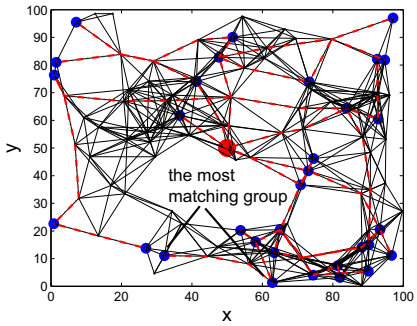


Fig. 4.9: The routing tree of EEAT-C

source node to the sink node is denoted by ds, and the average distance between two source nodes is denoted by dp. We vary these parameters to check the performance of different approaches. 1000 run simulations are repeated to get each data point of Fig. 4.10-4.15 with the confidence level of 0.95.

4.6.2 An Instance of EAT Built by Different Approaches

To have an intuitive understanding, Fig. 4.6-4.9 show some instances of routing trees built by different approaches. We can see that the centralized approach connects all the source nodes to the sink node using the shortest path (Fig. 4.6). TED generates a collection of fusion nodes in the network and the events from source nodes aggregate there first (Fig. 4.7). Its drawback is that the fusion nodes are fixed and hard to determine. MFST allows every sensor node to be a fusion node (Fig. 4.8). However, it does not support

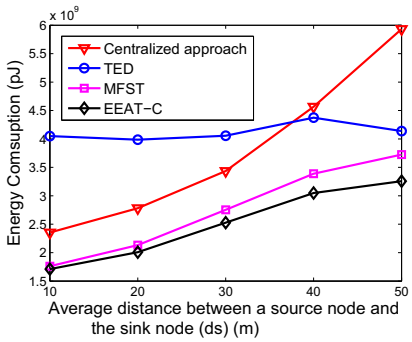


Fig. 4.10: Energy consumption comparison of different approaches varying the distance between a source node and the sink node

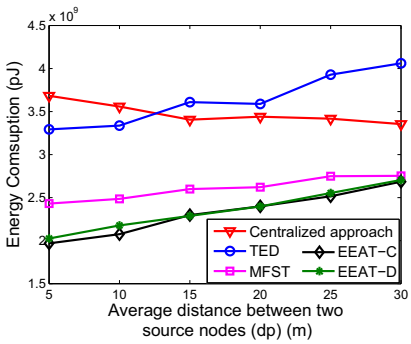


Fig. 4.11: Energy consumption comparison of different approaches varying the distance between two source nodes

composite event definition. Our approach goes one step further, supporting composite event definition and promoting the aggregation in the most matching groups to achieve high energy efficiency (Fig. 4.9).

4.6.3 Impact of Average Distance between a Source Node and the Sink Node

The average distance between a source node and the sink node (ds) is an important factor impacting the energy consumption of event detection. We vary ds to compare the energy consumption of different approaches in Fig. 4.10, where $dp = 20$ and $drr = 0.5$.

The energy consumed in the centralized approach linearly increases. This is because the transmission distance linearly increases when ds increases. TED is worse than the centralized approach when $ds \leq 38$ since the transmission cost dominates the energy consumption, which is reserved when $ds > 38$ since the benefits of aggregation in TED is large enough. MFST and EEAT-C are always better than the centralized approach and TED due to fine grain control of the routing tree. EEAT-C outperforms MFST since the event relations are utilized to optimize the routing tree. The energy saved of EEAT-C increases with the increase of ds, compared with MFST.

4.6.4 Impact of Average Distance between Two Source Nodes

The average distance between two source nodes (dp) is another important factor for energy consumption. Event aggregation is to make correlated events aggregate at the cost of transmitting data from one node to another. If dp is large, the overhead of aggregation increases. We vary dp to compare the energy consumption of different approaches in Fig. 4.11, where $ds = 30$ and $drr = 0.1$.

It is shown that the energy consumption of TED is less than that of the centralized approach when $dr \leq 12$ since the events aggregate at the fusion nodes, making the data amount to be transmitted reduced. However, when $dr > 12$, the result is reversed since the transmission cost to reach fusion nodes increases. MFST and EEAT-C are always better than the centralized approach and TED. The advantage of EEAT-C is more obvious when dp is small and unapparent when dp increases due to additional transmission cost. When $dp = 5$, EEAT-C saves 20% energy compared with MFST.

4.6.5 Impact of Event Relation

The last factor we consider is event relation. We first consider the EBT as described in 4.6.1 and change drr to check the performance of different approaches. drr denotes the change of the data amount in the event aggregation. The result is shown in Fig. 4.12 where $ds = 40$ and $dp = 20$.

The centralized approach and MFST are not so sensitive to the change of drr compared with TED and EEAT-C. The centralized approach collects all primitive events to the sink node, without considering the reduction of data amount due to aggregation. MFST assumes a non-decreasing aggregation of any two events and does not utilize data reduction rate explicitly. TED utilizes this information so has a better performance than that of the centralized approach when $drr \leq 0.62$. When drr is larger (means less aggregation), the

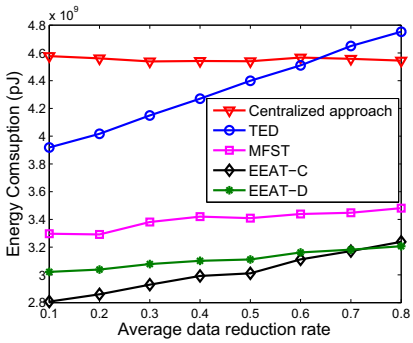


Fig. 4.12: Energy consumption comparison of different approaches varying data reduction rate

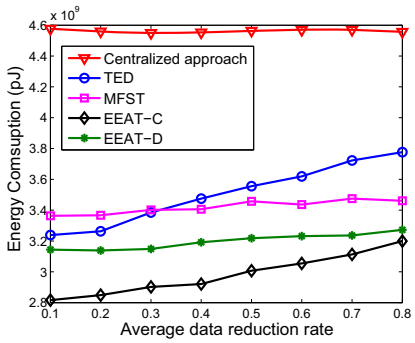


Fig. 4.13: Energy consumption comparison of different approaches varying data reduction rate (revised EBT structure)

benefit through aggregation is relatively limited compared with the overhead to achieve aggregation. TED is inferior to MFST since the transmission from the source nodes to the fusion nodes is not optimized. With full utilization of event relations, EEAT-C always consumes the least energy among these approaches. The energy saved is 7% to 15% that of MFST. Considering in real applications, simple predicate (yes or no) can be used to summarize a number of underlying events. In that situation, drr can be quite close to 0 (e.g. 0.001). Therefore, EEAT-C can achieve more significant energy saving.

We change the EBT by adjusting its depth from 5 to 4 and then 3, and got similar results. We further make the nodes of EBT having different data reduction rates to check the performance. Specifically, the data reduction rate of half of the nodes at level 4 follows the normal distribution $N(drr, 1)$. Half of the nodes at level 2 or 3 have the data reduction rate of 0.1. All other nodes have the data reduction rate of 1. The result is shown in Fig. 4.13. The result is similar with previous discussion except that TED outperforms MFST when $drr \leq 0.3$. This is due to the utilization of event relations in TED. The energy saved of EEAT-C is between 8% ($drr = 0.8$, compared with MFST) and 16% ($drr = 0.1$, compared with TED).

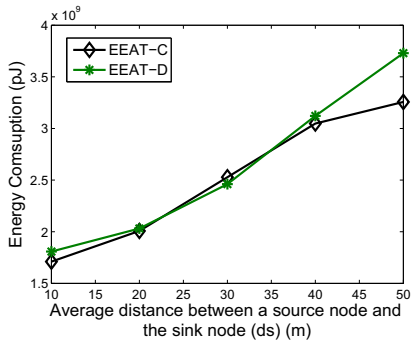


Fig. 4.14: Energy consumption comparison of our centralized alg. and distributed alg. varying the distance between a source node and the sink node

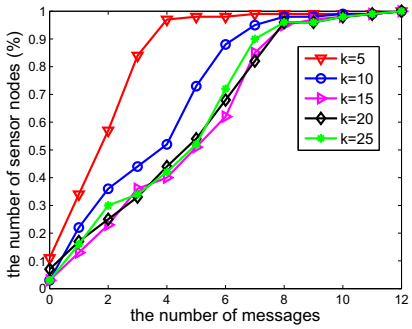


Fig. 4.15: The communication overhead of our distributed alg. when considering k-hop neighbors

4.6.6 The Performance of Our Distributed Algorithm

We compare the performance of our distributed algorithm EEAT-D with the centralized algorithm EEAT-C. Similar with above simulations, we change *dr*, *ds* and *drr* to check the energy consumption in event aggregation. The result with respect to *dr* is shown in Fig. 4.14, and the results with respect to *ds* and *drr* are shown in Fig. 4.11-4.13. In all cases, EEAT-D has quite similar performance with EEAT-C and the difference is less than 15%, which shows the effectiveness of the distributed mechanism used in our approach. The communication overhead distribution of sensor nodes in EEAT-D is shown in Fig. 4.15 when $L = 3$, $ds = 30$ and $dp = 10$. It shows that 90% sensor nodes need less than 8 messages to build the event aggregation tree even though 25-hop (sufficient large) neighbors are considered, which is quite energy-efficient.

4.7 Summary

In this chapter, we investigated energy-efficient event aggregation to achieve composite event detection. Complex relations in a composite event are considered. We first built a composite event detection model based on event definition tree and event aggregation tree. Event definition tree supports generic event definitions that may have arbitrarily complex

relations, meeting complex requirements of the user. Event aggregation tree is an energy-efficient routing tree for event detection, which is optimized by considering event relations. We compared event aggregation with data aggregation, and then proposed two principles for building event aggregation tree: supporting complex composite event definition and supporting decreasing aggregation function. Following these principles, we proposed both centralized and distributed approaches to revise a specific data aggregation algorithm named MST to build energy-efficient event aggregation tree. Simulation results show that our approach saves up to 20% energy than existing approaches.

Chapter 5

Energy-efficient Aggregation of Multiple Composite Events in WSNs

In this chapter, we investigate the energy-efficient aggregation of multiple composite events in WSNs. We make some composite events share the event aggregation trees to save the energy. This chapter is organized as follows: Section 5.1 is the overview of this chapter. Section 5.2 introduces the system model in this work and formulates the problem. Section 5.3 describe the structure of the solution. Section 5.4 and Section 5.5 illustrate the detailed solutions of this problem. Simulation results are reported and discussed in Section 5.6. Finally, Section 5.7 concludes this chapter.

5.1 Overview

In this chapter, we consider the event aggregation in multiple events situation. We assume that each of the events has a latency constraint and aggregation function (denoting event relations). We aim to optimize the event aggregation process considering both the latency constraint and aggregation function.

This work is different from the work in Chapter 4, which considers only a single composite event without latency constraint. The purpose of the work in that chapter is to minimize

the energy consumption of an event aggregation tree considering generic event relations. In this chapter, we need to consider both event relation (denoted by aggregation function) and latency constraint, in the context of multiple composite events. The purpose of the work in this chapter is to minimize the overall energy consumption of all these composite events.

In existing works, some analyzed the impact of aggregation function on the aggregation process [CBLV04, GE05, LLD06], where an aggregation function describes the relations included in an event. Others considered the latency constraint to meet time sensitive requirements [YKP04, MCM+06]. An aggregation considering both factors is more general but has hardly been studied yet especially for an event with complex relations. More importantly, there is no existing work having studied the optimal aggregation of multiple user-specified events. It is an important problem, considering nowadays many applications are composed of multiple such events. Taking a WSN-based intelligent transportation system [CCCT06] for example, many events are requested by different users for different purposes. Some users may have interests in the average speed of vehicles, while some others may want to know the speed of individual vehicles. The events have different correlations among their low-level events (hence different aggregation functions) and different latency requirements. Existing approaches have not investigated the diversity of the requirements in event aggregation.

In this chapter, we investigate the optimal aggregation of multiple events with different latency constraints and aggregation functions in WSNs. We propose a three-phase solution to solve this problem. In the first phase, we select several events as base events. In the second phase, we build the optimal event aggregation trees for these base events considering both latency constraint and aggregation function. And in the last phase, each event selects a proper tree from them to perform aggregation. Different from existing works which build an event aggregation tree for each event, our approach makes some events share

event aggregation trees rather than build their own trees. Through proper design of the approaches, we can achieve optimal event aggregation for these events. Simulation results show that our approach outperforms existing approaches and saves a significant amount of energy. To our best of knowledge, this is the first work to explore event aggregation in this aspect. In summary, this chapter makes the following specific contributions.

- We propose the Delay-bounded Event Aggregation Algorithm (DBEA) to consider both latency constraint and aggregation function for individual events.
- We investigate the optimal aggregation of multiple events with different latency constraints and aggregation functions. Simulation results show that our approach outperforms existing approaches and saves a significant amount of energy (up to 35% in our system).

5.2 System Model and Problem Formulation

We first describe the system models used in this work. Based on these system models, we conduct preliminary analysis and then formulate the problem.

5.2.1 Event

An event is a record of an activity occurred in a system. Due to complex user requirements, an event is usually consists of multiple correlated sub-events. These sub-events can be further decomposed into their sub-events in a iterative way, and eventually into a set of primitive events that can be directly detected by one detection node such as a sensor. In this work, we consider a number of n composite events specified by the users to be detected. Each event is specified a latency constraint that denotes the time limit to finish the process of event detection. Each event also has an aggregation function which denotes the relations included in the event. These user-specified composite events are referred to as just

composite events if no ambiguity.

5.2.2 Wireless Sensor Network

A WSN is modeled as a graph $G(V, E)$, where V denotes the set of sensor nodes and E denotes the set of feasible communication links between a pair of sensor nodes. A subset S of V are sources nodes which monitor the environment and generate primitive events. A sink node $r \in V$ issues event detection requests and receives the results.

5.2.3 Event Aggregation in Wireless Sensor Network

Event aggregation is a process of combining several sub-events into a composite event, and eventually the user-specified events. It is an important approach to achieve event detection, and also help to eliminate redundant information to be transmitted to save energy. In a WSN, a routing structure is usually built among sensor nodes to facilitate the information exchange of event aggregation. For the sake of simplicity, we only consider tree as the aggregation structure, referring it as *event aggregation tree*. When an event is ready in a node, it forwards the event to its parent node in the event aggregation tree.

Latency constraint and aggregation function are two important factors affecting the structure of event aggregation tree. When latency constraint is tight, the aggregation tree trends to be the shortest path tree and little opportunity to perform event aggregation. By contrast, if latency constraint is loose (One extreme case is no latency constraint), optimal aggregation can be achieved. Aggregation function also affects event aggregation. A larger reduction in data amount means that the event is more sensitive to event aggregation.

5.2.4 Preliminary Analysis

There are two issues needed to be addressed to achieve a desirable event aggregation. First, existing approaches have not considered both aggregation function and latency constraint. Second, existing approaches are designed for single event aggregation but cannot

directly used for multiple events with different latency constraints and aggregation functions. For the first issue, we will propose a solution in Section 5.4 and so defer the discussion about it. Even when the first issue is addressed, the second issue is still very challenging because of the characteristics of WSNs.

In a traditional way, an optimal event aggregation tree is built for each event. These aggregation trees are determined and then transferred to corresponding sensor nodes through multi-hop communications. However, this is not always feasible for several reasons. First, the communication overhead of transferring the information of event aggregation trees from the sink node to other sensor nodes is large, which is increased with the number of events and the number of sensor nodes involved in each tree. Second, if a new event needs to be detected, a new tree needs to be built before detection, which causes additional latency. Third, a large number of event aggregation trees do not only consume limited memory of sensor nodes (only 3500 bytes are available for Mica2 and Micaz [Cro06]), but also increase the maintenance overhead in the sensor nodes such as communication link maintenance, routing adaptation, duty cycling control, etc.

To overcome these drawbacks, it is reasonable to make some events share the event aggregation trees rather than build their own. By building several proper event aggregation trees and make them serve for the detection of all events, a desirable performance can be achieved. We assume that at most a number of m such event aggregation trees can be built. The specific value of m depends on different consideration to overcome above mentioned drawbacks including communication overhead, latency, and maintenance overhead. We first assume this value as a precondition to design our solution, and then discuss how to choose m if considering only the communication overhead. Further discussions about other factors are left as future work.

For the sake of simplicity, we select m of n events as base events to build the aggregation

trees, while others will choose one of them to perform event aggregation. The aggregation trees of base events are called *base event aggregation trees*. Correspondingly, We call the latency constraints of base events *base latency constraints*, and the aggregation functions of base events *base aggregation functions*. Each event chooses a base event as its *home base event*, and perform the event aggregation using the event aggregation tree of that home base event.

It is noticed that since some events use a shared event aggregation tree other than the optimal aggregation tree of them, the performance in terms of traffic overhead may be degraded. It weakens the effectiveness of event aggregation. For a better management of the performance in multiple events environment, we hope to bound the performance degradation of an event aggregation to some degree, with reference to the optimal performance it can achieve. We call this requirement *performance constraint*. The constraint can be posed on all events or only some important events. In this work, we consider the events with base latency constraints as such important events, which will be suggested to the users to promote the usage of base latency values. Other constraints are summarized in Section 5.2.6 and discussed in Section 5.5.5.

5.2.5 Problem Formulation

We then formulate the problem as an Event Aggregation with Different Latency Constraints and Aggregation Functions Problem (EALA):

Given:

a. A WSN $G(V, E)$. There is a set of source nodes $S \subseteq V$ and a sink node $r \in V$.

b. A number of n composite events $ce_i (i = 1...n)$ which need to be detected. Each composite event ce_i has a latency constraint l_i and an aggregation function af_i.

find an aggregation structure in the WSN which consists of at most m base event aggregation trees, such that the total message cost of event aggregation for all the composite events is minimized.

subject to

a.(latency constraint) The event aggregation of each composite event meets its latency constraint.

b.(performace constraint) Each composite event performs the aggregation with the performance at most $p\%$ worse than the optimal performance it can achieve in the individual event aggregation scenario.

We observed that the performance degradation is from the difference (also called gap) in latency constraint and aggregation function between base events and other events, so the problem is equivalent to minimize these differences for all composite events. We further describe the problem how to select base events using the following formula where $HB(ce_i)$ denotes the home base event of event ce_i, $LG(a,b)$ and $AG(a,b)$ denote the difference in latency constraint and aggregation function between event a and b, respectively, and $\alpha_i, \beta_i (i = 1...n)$ are weights assigned to latency constraint and aggregation function of event ce_i.

$$
\begin{aligned}
&\min \sum_{i=1}^{n} (\alpha_i \cdot LG(l_i, l_{HB(ce_i)}) + \beta_i \cdot AG(af_i, af_{HB(ce_i)})) \\
&st.\ \alpha_j \cdot LG(l_j, l_{HB(ce_j)}) + \beta_j \cdot AG(af_j, af_{HB(ce_j)}) < \Delta, \\
&\quad j : l_j \in \{l_{HG(ce_1)} ... l_{HG(ce_n}\} \\
&\quad l_{HB(ce_i)} < l_i,\ i = 1...n \\
&\quad HB(ce_i) \in \{1...n\}
\end{aligned}
\tag{5.1}
$$

Several problems need to clarified before solving this formulation, such as the determination of home base events, and the representation of differences in latency constraint and aggregation function. Moreover, we need to determine the building of event aggregation tree if base events are selected.

5.2.6 Problem Variants

There are several variants of the problem with different constraints and objective functions. Here we summarize the family of this problem as follows.

For the constraints, we have strong performance constraint, weak performance constraint, and multi-level performance constraint:

(Strong Performance Constraint) The performance degradation of every event is bounded.

(Weak Performance Constraint) The performance degradations of some events are bounded.

(Multi-level Performance Constraint) The performance degradations of different events are bounded differently.

For the objectives functions, we can minimize overall gap, or maximize desirable event population:

(Minimize Overall Gap) Minimize the overall gap in latency constraint and aggregation function between the events and their home events, assuming that all the events can satisfy the constraints.

(Maximize Desirable Event Population) Maximize the number of events that can satisfy the constraints.

Different combinations of constraints and objective functions generate different problems, which adapt to diverse application requirements. In this work, we mainly focus on

weak performance constraint and minimizing overall gap. However, we will also briefly discuss other problems in Section 5.5.5 for the reader's reference.

5.3 Solution Framework

We propose a three-phase solution framework to solve this problem, which is illustrated as follows:

Phase 1(Base Event Selection Phase): Among n composite events, select at most m of them as base events.

Phase 2(Base Aggregation Tree Building Phase): Each base event builds the optimal event aggregation tree based on its latency constraint and aggregation function. All these event aggregation trees are put in an event aggregation tree repository and distributed stored in the sensor nodes. This is the aggregation structure for this problem.

Phase 3(Aggregation Phase): Each event chooses the most appropriate base event as its home base event. After that, the event aggregation tree of this base home event is retrieved from the repository for performing event aggregation.

Based on this solution framework, we will illustrate the detailed solution as the following order: first, we propose the solution for Phase 2 in Section 5.4, which constructs the optimal event aggregation tree for individual composite events. Then we propose the solution for Phase 1 and Phase 3 in Section 5.5, which completes the solution for multiple composite events.

5.4 Base Aggregation Tree Building

In this section, we propose the Delay Bounded Event Aggregation Algorithm (DBEA) to build an event aggregation tree for single composite event, considering both latency

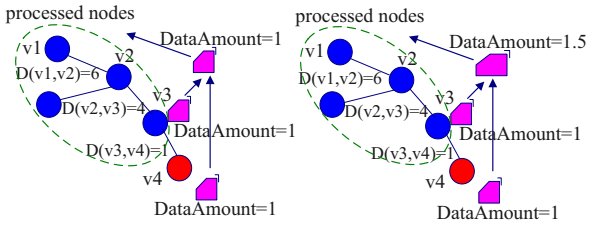


Fig. 5.1: Fully aggregation (left) and patrial aggregation (right)

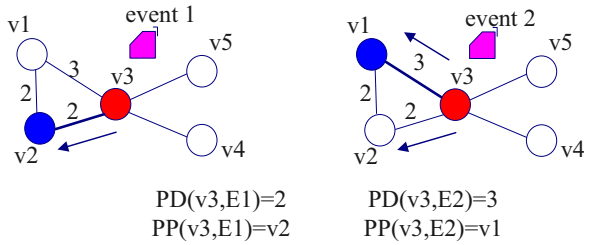


Fig. 5.2: Possible aggregated distance and conflicting optimal parent candidates.

constraint and aggregation function.

The algorithm builds the event aggregation tree starting from just the sink node, and then calculates the distance and latency of its one-hop neighbors from the sink node. This process is repeated with increasing hop number and stopped if the latency constraint is violated. When a source node is processed, this source node and its proper parent nodes are added into the event aggregation tree. Compared with existing approaches, the distance calculated in this algorithm has two differences. First, it considers partial aggregation rather than just fully aggregation in [TM80, SS97]. When a source node is added into the tree, the distance increased is not the distance between this node and the processed node set (defined as the minimal distance between this node and any node in that set), but the distance between this node and the sink node. One example is shown in Fig. 5.1 where $v1$, $v2$ and $v3$ are already processed. The distance between $v4$ and the processed node set is

1, which is exactly the increased distance in fully aggregation since when a primitive event traverses from $v4$ to $v3$, it merges into the primitive event of $v3$ hence does not increase the cost from $v3$. This does not hold in partial aggregation. The primitive event traversing to $v3$ may aggregate into an event with the data amount more than 1 (eg. 1.5), then causes additional cost from $v3$ to the sink node (eg. $(1.5-1)(6+4)=5)$). Second, the distance is not simply the physical distance, but with respect to different types of primitive events. One processed node may have conflicting optimal parent candidates. We introduce the concept of *possible aggregation distance* to handle this problem. As shown in Fig. 5.2, given $v1$, $v2$ and $v3$ that are already processed, when primitive event $e1$ transmits to v_3, the optimal distance is 2 subject to the parent of $v2$, while for primitive event $e2$, the optimal distance is 3 subject to the parent of $v2$. We record all these possible aggregation distances and defer the final decision until the primitive events traversing this node are determined.

The details of DBEA are shown in Algorithm 7 and Algorithm 8. Function *buildTree* returns an event aggregation tree T for a composite event with latency constraint $maxLatency$ and aggregation function Ag. ET is the set of primitive events involved in the composite event. $PD[v][et]$, $PP[v][et]$, $PL[v][et]$ are possible aggregation distance, parent pointer, and latency, respectively, when an event $et \in ET$ transmits from node v to r. After considering all event types, $D(v)$, $P(v)$, $L(v)$ record current optimal values in the node v. $phyD(v)$ is the physical distance from v to r. Line 1 initiates a latency shortest path tree $LSPT$ as the reference structure. Line 2-8 set initial variable values and firstly put only r in T. Q stores the nodes which have not been considered. Followed this is a loop in line 9-23, each iteration chooses a nodes with minimum $D(v)$. If the latency cannot be further increased (minimal $D(v)$ is inf), using $LSPT$ to add v_j into T as in line 11-13. Otherwise, if v_j is a source node, as in line 17, the nodes in $Path(v_j, T)$ are added into T where $Path(v_j, T)$ means the path from v_j to T. All affected nodes during above process (eg. v_j and the nodes in

Algorithm 7: Delay Bounded Event Aggregation Algorithm (DBEA)

Global: V, E, S, r, ET
//Global variables for Alg.7 and Alg.8. V, E: sensor nodes and possible communication links for a WSN; S: source nodes; r:sink node; ET: the set of composite event types;

Input : *maxLatency*, *af*
//an event with latency constraint *maxLatency* and aggregation function *af*
Output: aggregation tree T
Function: buildDelayBoundedAggTree(*maxLatency*, *af*)

```
1  build the latency shortest path tree LSPT
2  foreach v ∈ V do   //Enumerate all sensor nodes
3  |  D(v) = inf, P(v) = null, L(v) = inf, phyD(v) = inf
4  |  foreach et ∈ ET do   //Enumerate all event types
5  |  |  PD[v][et] = inf, PP[v][et] = null, PL[v][et] = inf
6  |  end
7  end
8  D(r) = 0, T = {r}, Q = V
9  while Q ≠ ∅ do
10 |  find v_j such that D(v_j) = min_{v_i∈Q}(D(v_i))
11 |  if D(v_j) == inf then
12 |  |  use LSPT to add v_j into T
13 |  end
14 |  Adjust = {v_j}
15 |  if v_j ∈ S then
16 |  |  add the nodes in Path(v_j, T) to Adjust
17 |  |  add the nodes in Path(v_j, T) to T
18 |  |  adjust PD, PP, PL, P, D, L, phyD in Path(v_j, T)
19 |  end
20 |  foreach node u such that (t, u) ∈ E, t ∈ Adjust do
21 |  |  modifyAggDistance(t,u)
22 |  end
23 end
24 return T
```

$Path(v_j, T)$) are added into a set named *Adjust* and then call Function *modifyAggDistance*, as in line 14, 16, 20-22.

Function *modifyAggDistance* diffuses distance and latency measurements from node t to a neighbor node u. $D(t, u)$ and $L(t, u)$ denote the distance and latency between t and u, respectively. $D(t)$ and $L(t)$ denote the distance and latency between t and r, respectively. $E(t)$ returns the types of primitive events that may traverse t. There are three kinds of nodes in the function: tree node (already in T), intermediate node (considered but not in T) and source node(not considered). We distinguish three cases of t to u: tree node

Algorithm 8: DBEA-Distance Modification

```
   Input : t, u    //t, u: two sensor nodes
   Result: updated PD, PP, PL, D, P, L
   Function: modifyAggDistance(t, u)
 1 if t ∈ T then   //tree node to other nodes
 2     foreach et ∈ ET do
 3         let maxFac = max_et af(E(t), et)
 4         if min(phyD(t) × (1−maxFac), PD(t, et)) + D(t, u) <
           PD(u, et) && L(t) + L(t, u) < maxLatency then
 5             update PD(u, et), PP(u, et), PL(u, et)
 6             if u ∈ S && E(u) == et then
 7                 update D(u), P(u), L(u)
 8             else
 9                 if min(phyD(t), D(t)) + D(t, u) < D(u) && L(t) + L(t, u) < maxLatency
                   then
10                     update D(u), P(u), L(u)
11                 end
12             end
13         end
14     end
15 end
16 if t ∉ T && u ∉ S then   //inter node to inter node
17     foreach et ∈ ET do
18         if PD(t, et) + D(t, u) < PD(u, et) && PL(t, et) + L(t, u) < maxLatency then
19             update PD(u, et), PP(u, et), PL(u, et)
20         end
21     end
22     if D(t) + D(t, u) < D(u)&&L(t) + L(t, u) <maxLatency then
23         update D(u), P(u), L(u)
24     end
25 end
26 if t ∉ T && u ∈ S then   //inter node to source node
27     dist = min(PD(t, E(u)), D(t)) + D(t, u)
28     latency = PD(t, E(u)) < D(t)?PL(t, E(u)) : L(t)
29     if dist< D(u) && latency +L(t, u) < maxLatency then
30         update the data in PD(u, et), PP(u, et), PL(u, et), D(u), P(u), L(u)
31     end
32 end
```

to other nodes, intermediate node to intermediate node, and intermediate node to source node. $PD(u, et)$, $PP(u, et)$ and $PL(u, et)$ are firstly calculated from tree nodes as in line 4-5, then accumulated between intermediate nodes as in line 14-18, finally reached source nodes as in line 24-28. For intermediate node, $D(v)$ is simply accumulated in terms of the physical distance, as in line 8-9 and 19-21. For a source node, it is determined according to the aggregation distance with respect to the primitive event type of that source node, as in

line 6-7.

5.5 Optimal Aggregation of Multiple Composite Events

After we discussed the solution of Phase 2, we turn to the solutions of Phase 1 and Phase 3, which are unique for the event aggregation of multiple composite events. We have observed that some composite events may suffer performance degradation since only m of n composite events can be selected as base events to build the aggregation structure. The performance degradation is from the difference between non-base events and base events in latency constraint and aggregation function. To solve this problem, we will first discuss how to describe the difference of two events in latency constraint and aggregation function. After that, we will discuss how to select a proper home base event for a composite event. Based on this, we will finally discuss how to select the optimal m base events to achieve a good performance.

5.5.1 Latency Gap

The latency constraint of a composite event is usually represented by a double value, eg. 2.5 minutes, 50 seconds, etc. The difference in latency constraint between two composite events is easy to understand. We have the following definitions:

Definition 1: The difference in latency constraint between event A and event B is defined as the *latency gap between A and B*, denoted by $LG(A, B)$.

The latency gap between two events is a non-negative double value. For example, event A with a latency constraint of 25.5 minutes and event B a the latency constraint of 15.1 minutes, the latency gap between A and B is 10.4 minutes.

Definition 2: The difference in latency constraint between event A with and its home base event is defined as the *latency gap of A*, denoted by $LG(A)$.

5.5.2 Aggregation Function Gap

The difference in aggregation function between two composite events is relatively hard to describe. This is due to the descriptive forms of aggregation function. Some previous works consider only fully aggregation, which is not suitable for general aggregation case. Some other works use abstract aggregation function [LLD06, YKP04], which takes two events as the input and an aggregated event as the output. It can handle general aggregation case but not easy to compare two aggregation functions in this form. Here we propose a simple approach called *relation matrix* to describe aggregation functions. We focus on the data reduction due to the duplicate information in the primitive events. This approach will simplify the comparison between two aggregation functions.

Given primitive events $e_i(0 \leq i \leq n)$, a relation matrix $A_2 = [a_{ij}](0 \leq i, j \leq n, 0 \leq a_{ij} \leq 1)$ denotes the similarity between two primitive events, where a element a_{ij} denotes the similarity between event i and event j. Similarly, relation matrixes $A_3 ... A_n$ denote the similarity among multiple primitive events. The following is an example of A_2:

$$A_2 = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}$$

$a_{11} = 1$ means that two events of type i_1 are 100% similar hence they can merge into one event of the same size. $a_{12} = 0.5$ means an event of type i_1 has 50% similarity with an event of type i_2 so they can share 50% information.

We give the formal rules of information reduction using relation matrix A_2 when event e_1 and e_2 aggregate into an event e_{result}. For the sake of simplicity, we assume all primitive events having the same data amount $d(e_p)$. $d(e_1)$, $d(e_2)$ and $d(e_{result})$ denote the data amount of event e_1, e_2, and e_{result}, respectively.

Rule 1: If e_1 and e_2 are of primitive event type i and j respectively,

$$d(e_{result}) = 2d(e_p) - d(e_p) \times a_{ij}$$

Rule 2: If e_1 is a composite event derived from a set of primitive event types $S = \{i_1, i_2...i_m\}$, e_2 is of primitive event type j,

$$d(e_{result}) = d(e_1) + d(e_p) - d(e_p) \times a_{jj} \qquad (j \in S)$$
$$d(e_{result}) = d(e_1) + d(e_p) - d(e_p) \times (\sum_{t \in s} a_{tj} - O_{Sj}) \qquad (j \notin S)$$
$$O_{Sj} = a_{c_2^s j} - (|C_2^S| - 1)a_{c_3^s j} + ...(|C_{|S|-1}^S| - 1)a_{Sj}$$

where $|.|$ denotes the cardinality of a set, C_i^S denotes the i-combination set of S. For example, assuming that $S = \{A, B, C\}$, then $C_2^S = \{AB, BC, AC\}$ and $|C_2^S| = 3$. $\sum_{t \in s} a_{tj}$ is the sum of shared data amount between event type j and every event type t in S. O_{Sj} denotes the overlap amount of these shared data amount which needs to be cut off in the computation. For example, among the items in O_{Sj} , $a_{c_2^s j}$ denotes the data amount of overlapped information among three events in which two of them are from C_2^S and the other is the event j. It can be computed by that $a_{c_2^s j} = \sum_{pq \in C_2^S} a_{pqj}$ where a_{pqj} is a element of relation matrix A_3.

Rule 3: If e_1 is a composite event derived from a set of primitive event types $S = \{i_1, i_2...i_m\}$, e_2 is a composite event derived from a set of primitive event types $T_1 \cup T_2$, $T_1 = \{j_1, j_2...j_{n1}\}$, $j_1, j_2...j_{n1} \in S$, $T_2 = \{k_1, k_2...k_{n2}\}, k_1, k_2...k_{n2} \notin S$

$$d(e_{result}) = d(e_1) + d(e_2) - d(e_p) \times (\sum_{i \in T_1} a_{ii} - O_{T_1} + \sum_{i \in T_2} \sum_{t \in S} a_{ti} - O_{ST_2} - O_{T_1 T_2})$$

where O_{T_1} denotes the shared data amount of all event types in T_1, and O_{ST_2} , $O_{T_1 T_2}$ are similarly defined to eliminate the overlap of shared data between S and T_2, T_1 and T_2, respectively.

Fig. 5.3 shows an example in aggregation using our rules. Two event type i and j are involved in this example, whose relation matrix is as follows:

$$A_2 = \begin{pmatrix} 0.5 & 0.2 \\ 0.2 & 0.5 \end{pmatrix}$$

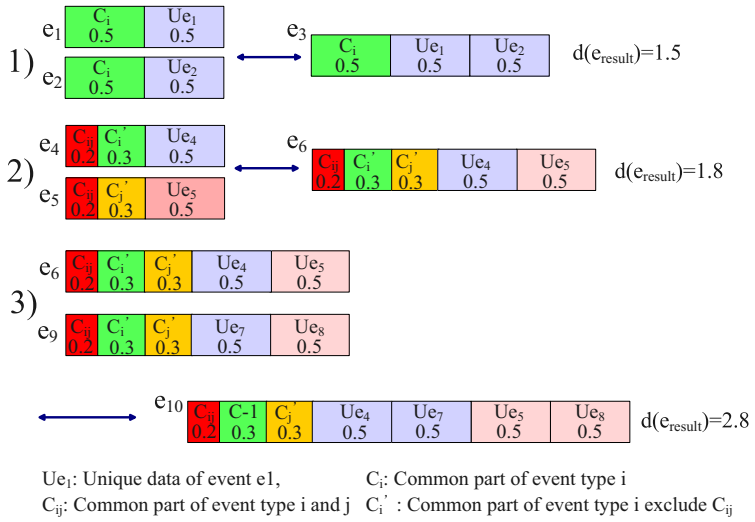


Fig. 5.3: Different kinds of event aggregation. 1) aggregation of two primitive events of the same event type. 2) aggregation of two primitive events of different event types. 3) aggregation of two composite events.

The case 1 is the aggregation of two primitive events e_1 and e_2 of the same event type i. C_i denotes the common data in all events of this event type, while U_{e_1} and U_{e_2} denote the unique data in individual event e_1 and event e_2, respectively. The numbers annotated in the figure denote the data amount of corresponding information. When event e_1 and event e_2 aggregates, the result shown in the right has a data amount of 1.5. This matches the result using our rule 1 in which the computation is $1 + 1 - 0.5 \times 1 = 1.5$.

The case 2 is the aggregation of two primitive events e_4 and e_5 of event type i and j, respectively. C_{ij} denotes the common data in event type i and j, and C_i' denotes the common data in the events of event type i, excluding C_{ij}. That is $C_i' = C_i - C_{ij}$. C_j' is similarly defined. The aggregated event e_6 has a data amount of 1.8, which matches the result using rule 1 in which the computation is $1 + 1 - 0.2 \times 1 = 1.8$.

The case 3 is used to validate rule 3. e_6 is the result of the case 2. e_9 is a similar aggregation result with e_6, but from event e_7 and e_8. The ground truth of the aggregation

result is event e_{10} shown in the right. According to rule 3, $S = \{i, j\}, T_1 = \{i, j\}, T_2 = \emptyset$, the computation result is $1.8 + 1.8 - 1 \times ((0.5 + 0.5) - 0.2)$=2.8. It matches the ground truth quite well.

Now we discuss the gap in aggregation function of two composite events. We have the following definitions:

Definition 3: The difference in aggregation function between event A and event B is defined as *aggregation function gap between A and B*, denoted by $AFG(A, B)$.

Definition 4: The difference in aggregation function between event A and its home base event is defined as *aggregation function gap of A*, denoted by $AFG(A)$.

We have used relation matrix to describe the aggregation functions. Therefore we can use the norm [GVL96] of the relation matrix to compare the aggregation functions, which maps a relation matrix into a real number. There are different forms of norm including Manhattan norm, Euclidean norm, p-norm etc. [GVL96]. All these forms are equivalent to describe the difference of matrixes. We adopt Manhattan norm of A_2 in this work. If different events have different effects on the event aggregation, weighted norm can be used. If more accurate aggregation result is needed, the relation matrix A_3, A_4... can further be taken into account.

For example, the aggregation functions of composite event ce_1 and ce_2 are as follows, we then compute the aggregation function gap between A and B.

$$A_2^{ce_1} = \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix}; A_2^{ce_2} = \begin{pmatrix} 0.5 & 0.2 \\ 0.2 & 0.5 \end{pmatrix}$$

$$AFG(A, B) = (1 + 0.5 + 0.5 + 1) - (0.5 + 0.2 + 0.2 + 0.5) = 1.6$$

Algorithm 9: Event Aggregation Tree Determination Algorithm (EATD)

Input : ce_j, $T = \{T_1....T_m\}$, $BE = \{ce_{v_1}....ce_{v_m}\}$ //event aggregatiopn tree repository T, base event set BE, T_i is built from ce_{v_i}, ce_j is an event need to determine the event aggregation tree

Output: event aggregation tree for event ce_j

Function: deterEventAggTree(ce_j, T, BE)

```
if ce_j ∈ BE then //base events
    find the base event ce_{v_i} such that ce_{v_i} = ce_j
    T_result = T_i
else //non-base events
    find the base event sets BE' such that l_k ≤ l_j (ce_{v_k} ∈ BE').
    find a base event ce_{v_i} such that α_i(l_{v_i} − l_j) + β_i|af_{v_i} − af_j| (ce_{v_i} ∈ BE') is minimized
    T_result = T_i
end
return T_result
```

5.5.3 Home Base Event Selection

In this section, we discuss the solution of home base event selection for Phase 3. For base events, there is not a concern because its home base event is itself and the event aggregation tree is already optimal, which is guaranteed by DBEA. The difficulties exist in the non-base events. For each non-base event, we need to determine its home base event and conduct the aggregation.

Assume that the base events are already determined and for each base event an event aggregation tree is built using DBEA. Each non-base event need select one of them to perform event aggregation and ensure that the latency constraint and performance constraint are satisfied. The selection needs to consider both latency and aggregation function.

An example is used to demonstrate the selection. We assume that there are three base events ce_1, ce_2, and ce_3, with latency constraint l_1, l_2, and l_3, respectively. $l_1 < l_2 < l_3$. The aggregation functions of them are af_1, af_2 and af_3, respectively. The aggregation function is described using the Manhattan norm of relation matrix A_2. An non-base event ce_i with latency constraint l_i and aggregation function af_i needs to determine its home base event.

($l_2 \leq l_i < l_3$). Both the weight α_i and β_i of ce_i are 1 (see Eq. 5.1).

We consider latency constraint first. To meet the latency constraint, ce_i should choose ce_1 or ce_2 as its home base event. Then we take the aggregation function together into account. According to our objective in Eq. 5.1, we need to minimize the gap in latency constrain and aggregation function between ce_i and its home base event. In this example, we choose ce_1 or ce_2, depending on which one minimizes the objective function $(l_i - l_j) + |af_i - af_j|, (j = 1, 2)$.

We summarize the policy to select home base events as follows:

Given event e_i with latency constraint l_i and aggregation function af_i, its home base event is the base event with latency constraint l_j and aggregation function af_j, such that 1) $l_j \leq l_i$, and 2) $\alpha_i(l_i - l_j) + \beta_i|af_i - af_j|$ is minimized among all base events.

Based on our analysis, we propose Algorithm 9 for Phase 3, to determine a proper event aggregation tree for any composite event. According to the algorithm, each base event uses its own event aggregation tree to conduct the aggregation (line 1-3). Each non-base event uses the aggregation tree of its base home event to conduct the aggregation (line 4-7). With this home base event selection, the latency constraint is guaranteed. This algorithm also try the best to minimize the performance degradation. However, the optimal value is still subjective to the selection of base events. Moreover, the performance constraint may not be satisfied without proper base events. We then further discuss the selection of base events in the next section.

5.5.4 Base Event Selection

In this section, we discuss the solution of Phase 1, to select the base events.

We group all the events according to their latency constraints and in each group further group them according to their aggregation functions. Without loss of generality, we assume that there are a number of n different latency constraints $l_i(i = 1...w)$. For events with

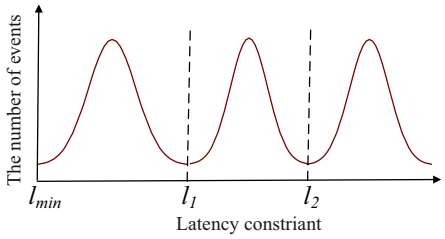


Fig. 5.4: Event distribution graph grouping by latency constraints

latency constraint l_i, the number of different aggregation functions is afn_i. A such example can be seen in Fig. 5.4.

We propose a two-level dynamical programming approach to select base events. In the outer dynamical programming process, the latency constraints are considered as stages. In each stage, we try to determine how many base events are needed. The objective is to minimize the weighted sum of latency gap and the aggregation function gap of all the events. The constraints are that the total number of base events is m and all events should satisfy the performance constraint. We can see if one stage is determined, the further computation is not related to the previous stages according to our non-base event aggregation policy, which is suitable for dynamical programming.

In stage l_i, we need determine how many base events needed and also which events are selected as base events. We called this sub-problem *Latency Constraint Fixed Base Event Selection* (LFBES). Supposing we begin to select a number of s of m composite events $(e_1, e_2, ...e_n)$ as base events in this stage, we need minimize the total aggregation function gap subjective to the performance constraint. Generally, the base events of these s composite events can also be the events with a latency constraint less than l_i, according to our home base event selection. However, if we exhaust all of them, the state space will increase dramatically. We use an approximation approach to overcome this problem: the base events only select their home base events having a latency constraint l_i. We solve this

problem also by dynamic programming [Ber05], inspired by video summary [LSKG05].

We use D_t^k to denote the minimum total aggregation gap in the state that k base events are selected after considering composite events $e_1, e_2, ...e_t$. P_t^k denotes the base event with maximum ID corresponding to D_t^k. We introduce two virtue composite events e_0 and e_{n+1} for the ease of discussion. $af_0 = 0$ and $af_{n+1} = \infty$. Composite events e_0 is assumed as a additional base event.

The solution is started from:

$$D_1^0 = \sum_{i=1}^{m} af_i, \quad P_1^0 = 0 \tag{5.2}$$

$$D_1^r = \sum_{i=1}^{m} (af_i - af_1), \quad P_1^r = 1 \quad (r \geq 1) \tag{5.3}$$

After we considered D_t^k, when we choose one more event e_j as the base event, the increased profit is

$$e^{t,k,j} = \sum_{i=P_t^k+1}^{m} [|af_i - af_{P_t^k}| - \min_{s \in \{P_t^k, j\}} |af_i - af_s|] \tag{5.4}$$

It is noticed that when e_j is added as the base event, the home base events of $e_{P_t^k}$,..., e_j are determined, which is either $e_{P_t^k}$ or e_j. We must guarantee $e_{P_t^k}$,..., e_j satisfy the performance constraint. We defined $v(D_t^k, t) = true$ if all these events satisfy the performance constraint, and $false$ otherwise.

Then the following is the recursive function:

$$D_t^k = \min (\min_{\substack{1 \leq j \leq t-1 \\ v(D_j^{k-1}, t) = true}} (D_j^{k-1} - e^{j,k-1,t}),\ D_{t-1}^k) \tag{5.5}$$

$$P_t^k = \begin{cases} P_{t-1}^k, & D_t^k = D_{t-1}^k \\ \arg\min_{1 \leq j \leq t-1} (D_j^{k-1} - e^{j,k-1,t}), & \text{otherwise} \end{cases} \tag{5.6}$$

For any specific m, we can reversely traverse the P_{n+1}^m to get the base events selected to meet the performance constraint. In this process, we observed that among all aggregation functions, the most effective ones are selected firstly and make the aggregation function

gap decrease quickly. With s increases, more aggregation functions are selected to further decrease the gap but the improvement is marginal. The property can be analyzed by a convex non-decreasing function. To meet the performance constraint, the minimal number of base events needed for stage l_i is denoted by mn_i.

After we solved LFBES, we continue our base event selection. $S^r_{l_i}$ denotes the state covering from l_1 to l_i and considering a number of at most r base events. Correspondingly, $D^r_{l_i}$ describes the objective value of $S^r_{l_i}$, $P^r_{l_i}$ records the maximum base latency constraint in $S^r_{l_i}$, and $Q^r_{l_i}$ denotes the number of base events with $P^r_{l_i}$. For latency constraint l_j, if s events are selected as base events, the total gap between them and all the events with $l_k(l_k \geq l_j)$ is defined as $g(l_j, s, l_k)$. mn_i is the minimum number of base events for latency constraint l_i, which is calculated by LFBES.

The starting function is

$$D^r_{l_1} = \begin{cases} \sum_{i=1}^{n}(\alpha_i l_i + \beta_i af_i), & r \in [0, mn_1) \\ \sum_{t=l_1}^{l_w} g(l_1, r, t), & r \in [mn_1, \min(m, afn_1)] \\ D^{\min(m,afn_1)}_{l_1}, & r \in (\min(m, afn_1), m] \end{cases} \tag{5.7}$$

$$P^r_{l_1} = \begin{cases} 0, & r \in [0, mn_1) \\ 1, & r \in [mn_1, m] \end{cases} \tag{5.8}$$

After we consider latency constraint l_j and s base events, if a number of k base events with constraint l_i are selected, the increased profit is

$$e^{l_j,s,l_i,k} = \sum_{t=l_i}^{l_w} d(l_j, s, t) - d(l_i, k, t) \tag{5.9}$$

Then the following is the recursive function:

$$D^r_{l_i} = \begin{cases} D^r_{l_{i-1}}, & r \in [0, mn_i) \\ \min\limits_{mn_i \leq k \leq \min(m,afn_i)} \{D^r_{l_{i-1}}, D^{r-k}_{l_{i-1}} - e^{P^{r-k}_{l_{i-1}}, Q^{r-k}_{l_{i-1}}, l_i, k}\}, & \\ & r \in [mn_i, m] \end{cases} \tag{5.10}$$

$$P_{l_i}^r = \begin{cases} P_{l_{i-1}}^r, & D_{l_i}^r = D_{l_{i-1}}^r \\ \underset{mn_i \le k \le \min(m, afn_i)}{\arg\min} (D_{l_{i-1}}^{r-k} - e^{P_{l_{i-1}}^{r-k}, Q_{l_{i-1}}^{r-k}, l_i, k}), & \text{otherwise} \end{cases} \tag{5.11}$$

The final result is obtained from $D_{l_w}^m$ and corresponding values.

Since the state space of the solution is still very large, we want to further decrease it. We adopt a FPTAS (Fully Polynomial Time Approximation Scheme) proposed in [HKL$^+$08, HKM$^+$09] to get a K-approximation result. In traditional DP, all the states need to be exhausted while in this approach, a sub set of the states called weak K-approximation set represents the whole space.

For a nondecreasing function $\varphi : [0, ...U] \rightarrow Z^+$, its weak K-approximation set is an ordered set $W = \{i_1 < ....i_r\}$ such that

1) $\{0, U\} \subseteq W \subseteq [0, ...U]$; and

2) for each $k = 1$ to $r - 1$, if $i_k + 1 > i_k + 1$ then $\varphi(i_k + 1) \le K\varphi(i_k)$.

Based on this weak K-approximation set, we can define a K-approximation function $\varphi' : [0, ...U] \rightarrow Z^+$ such that $\varphi'(x) = f(i_{k+1}), i_k < x < i_{k+1}$. In this way, the states of a function is polynomially bounded by the input size.

For our problem, in each step of calculation $D_{l_i}^r$, we need all the values of $D_{l_{i-1}}^r$, which needs large amount of computation. We observed that $D_{l_i}^r$ in fact is a nondecreasing function of r, we can use a K-approximation function to reduce the computation complexity and get a K-approximation result. The detailed approach is shown in Algorithm 10.

This algorithm follows the stages of dynamic programming to calculate the results. In each stage, Function *selectBaseEvents* first obtains the weak K-approximation set (line 2) and then builds the approximation function (line 3-5). After that, the algorithms records the results in D, P, Q for the computing of the next iteration. Function *weakSet* is called to build the weak K-approximation set of D_i^r. Different functions are used to generate the set according to if the stage is the first stage (line 1-5). After that, a binary search is used to

Algorithm 10: FPTAS Base Event Selection Algorithm (BES)

Global: l_i, $afn_i(i = 1...w), m, K$, D, P, Q
$//l_i$, $afn_i(i = 1...w)$: the composite events form w groups according to latency constraint. In group i, the events have the same latent constraint l_i and a number of afn_i aggregation functions. m: the max number of base events. K the approximation ration. D, P, Q: Results of the dynamic programming.

Output: selected base event in form of $D^m_{l_w}$,$P^m_{l_w}$ and $Q^m_{l_w}$
Function: selectBaseEvents(m)

```
for i = 1 to w do
    W_{l_i} = weakSet(l_i, K)
    Let D̄_{l_i} be the K-approximation function of W_{l_i}
    foreach r ∈ (min(m, afn_i), m] do
        add r to D̄_{l_i} , with the function value of D̄^{min(m,afn_i)}_{l_i}
    end
    for r = 0 to m do
        calculate and record D^r_{l_i} ,P^r_{l_i} and Q^r_{l_i} using D̄_{l_i}
    end
end
return D^m_{l_w} ,P^m_{l_w} and Q^m_{l_w}
```

Input : l_i $//l_i$: latency constraint
Output: the weak K-approximation set of the event group with the latency constraint l_i
Function: weakSet(l_i, K)

```
if l_i = l_1 then
    set function f as Eq.5.7, the number of base events r is the input variable
else
    set function f as Eq.5.10, the number of base events r is the input variable
end
generate the weak K-approximation set W_{l_i} of D^r_{l_i} using function f where
r ∈ [0, min(m, afn_i)]. The LFBES algorithm is called to calculate the detailed base event
selection.
return W_{l_i}
```

get the weak approximation set. The detailed algorithm can be see in [HKL+08]. During this process, the LFBES is called to determine the detailed selection of base events. It is clear that D_{l_i} and LFBES are not called only for several possible cases, and hence reduce the computation greatly.

Combining the solution of base event selection with base aggregation tree building (Section 5.4) and home base event (Section 5.5.3), we complete the solution of EALA.

5.5.5 Discussion

In this section, we discuss some extension of our solution and also possible solutions for the problem variants discussed in Section 5.2.6.

In LFBES, we take a simplification to consider the performance constraint for the events with base latency constraints. The home base events of them are just selected among the events with the same latency constraint. If considering this problem more strictly, we need consider the events with a latency constraint less than that. This can be implemented when taking the outer dynamic programming which considers $D^r_{l_i}$ and $D^k_{l_i}$, eliminate the events whose performance constraints are already guaranteed by the base events with the latency constraint less or equal to $P^r_{l_i}$. Similar works can be applied to all the latency constraint groups and then the solution is revised to satisfy the strong performance constraint. For the multi-level performance constraint, we assign different performance constraint values for different composite events. With respect to an alternative objective function of maximizing desirable event population, we can replace the objective value with the the number of events satisfy the constraints.

5.6 Simulation

This section presents the simulation results of our approach. The purpose of the simulation has two parts. The first one is to confirm our claims in previous sections. The second one is to compare our approach with existing approaches in different situations. The result reveals three issues. First, DBEA strictly meets the latency constraint and fully utilizes the aggregation to reduce the energy consumption. Second, our approach outperforms existing approaches for multiple events in a wide of situations. Third, the FPTAS in our approach leads to a desirable performance.

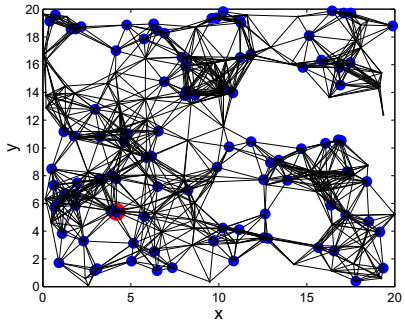


Fig. 5.5: An example of WSN in the simulation

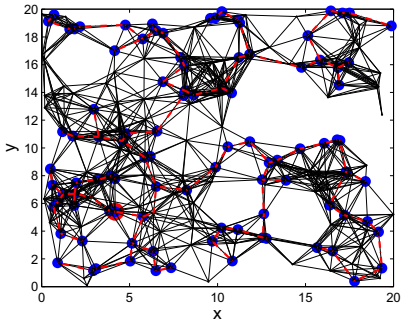


Fig. 5.6: An example of DBEA's result

5.6.1 Simulation Setup

We generate WSNs for the simulation based on the random graph model proposed by Waxman [Wax88], which is one of the most common models used in WSN research. We modified it a little by adding a max communication range constraint. In that model, the sensor nodes are randomly distributed in the area and the edges are added between any pair of sensor nodes. A edge between node u to node v exist with a probability as following:

$$P(\{u,v\}) = \begin{cases} \beta e^{\frac{-d(u,v)}{\alpha L}}, & d \in [0, rc] \\ 0, & d \notin [0, rc] \end{cases} \tag{5.12}$$

where d is the Euclidean distance from node u to node v, L is the maximum distance between any two nodes, rc is the maximum communicate distance of a node, and α and β $(0 < \alpha, \beta \leq 1)$ are two parameters to control the network structure. A larger value of β generate a graph with larger average node degree and a larger value of α makes a larger ratio of long edges relative to shorter edges for a node. In our simulation, we put 200 nodes into a $20 \times 20(m^2)$ space. A edge between a pair nodes denotes a possible communication link between these two nodes. The parameters are set as $\alpha = 0.3, \beta = 0.4$.

We use energy consumption as the performance metric in event aggregation. We follow the energy consumption model used in [LLD06]. Assuming the packet with data amount I and transmission distance d, the transmission cost is $I(\delta d^{\gamma} + \epsilon)(0 < d \leq rc)$, where ϵ is the

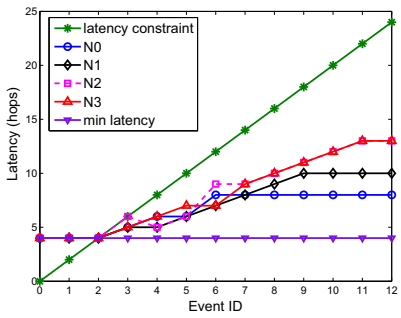


Fig. 5.7: The performance of DBEA

energy consumption on the transmitter and receiver circuit for one bit information and γ and δ are two parameters to reflect the radio transmission characterizes. In our simulation, the parameters are set that $\gamma = 2, \delta = 100(pJ/bit/m^2), \epsilon = 40(nJ/bit)$. The number of hops are used to describe the latency.

Assuming there are a number of n senor nodes in the WSN, we randomly choose one of them as the sink node and a number of θn senor nodes as the source nodes. $\theta = 0.33$ in our simulation. The primitive events are evenly distributed in the source nodes. A example of the WSN in our simulation is shown as Fig. 5.5. The sink node is in red color and largest marker width, the source nodes is in blue color and a large marker width, normal node is in black color and only a point.

We run 100 runs to get each data point in Fig. 5.7 to Fig. 5.13. The confidence level is 0.95.

5.6.2 The Performance of Delay Bounded Event Aggregation Algorithm

DBEA is used in our solution to construct the optimal event aggregation trees for individual composite events. In this section, we check the effectiveness of this algorithm in different latency constraints and aggregation functions.

We randomly generate WSNs and control the minimum latency of them (i.e. the depth

of SPT in WSNs) as *4*. A collection of primitive events with an aggregation function of $N0$ ($\|A_2\| = 0$) are distributed in the WSN. Afterwards, the sink node issues a set of composite events with the ID from *0* to *20*, and with the latency constraints from 0 to 25, respectively. We call DBEA to build the event aggregation trees for these composite events. An example of DBEA's result is shown in Fig. 5.6. The latency of the event aggregation trees are checked and reported. If the latency is large, there is more aggregation opportunities in the event aggregation tree. However, the latency also must be less than the requested latency to meet the latency constraint. The simulations are repeated with different aggregation function $N1(\|A_2\| = 1)$, $N2(\|A_2\| = 2)$ and $N3(\|A_2\| = 3)$. The result is shown in Fig. 5.7.

All the latencies of the event aggregation trees obtained by DBEA are between the minimal latency and the requested latency, which denotes that all latency constraints are strictly met. Meanwhile this approach always has a better performance compared with simply selecting the minimal latency. Growing benefits goes with increasing aggregation functions. The event aggregation tree obtained according to $N3$ is longer than the aggregation tree obtained according to other aggregation functions. For event *12*, the latency of the event aggregation tree with $N3$ is *13*, while that with $N0$ is only *8*. This is a desirable result since the aggregation function $N3$ has a larger norm of the aggregation function compared with $N0, N1$ and $N2$. The increased latency in the event aggregation tree enable more aggregation opportunities. DBEA adapts to different aggregation function quite well.

5.6.3 The Performance of Event Aggregation Considering Multiple Composite Events

Several simulations are undertaken to investigate the performance of our approach in the event aggregation of multiple composite events. For comparison, we choose several baseline algorithms for comparison including *full aggregation*, *average aggregation* and *random selection*. In full aggregation, the minimum latency constraint and fully aggregation

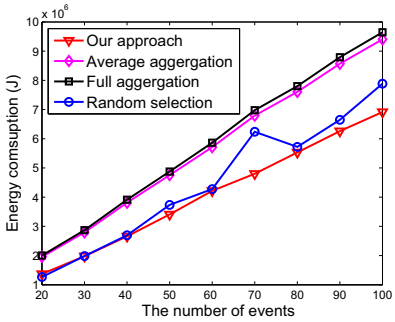


Fig. 5.8: The performance of our approach with different number of base events

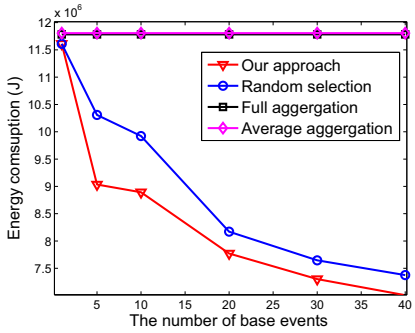


Fig. 5.9: The performance of our approach with different number of base events

function are used for all composite events to build the event aggregation tree. The average aggregation analyzes all the aggregation functions and calculates an average one to build the event aggregation tree. Random selection uses multiple base events like our approach, but only selects them randomly. We take the simulations with different number of events and different number of base events. The results are shown in Fig. 5.8 and 5.9.

In Fig. 5.8, we fix the number of base events as 20 and change the number of events from 20 to 100. As the result, our approach always has the least energy consumption among these four approaches. One interesting thing in the result is that the energy consumption of average aggregation is almost the same with that of full aggregation, which means the calculation of average value has limited help. This is because with the diversity of the events, only one average value cannot effectively help to decrease the latency gap and aggregation gap. Random selection and our approach use more base events hence have much better performance. When the number of events is 20, it saves 50% energy compared with fully aggregation. As the number of events increases to 50, the energy saved decreases to 30 percent but the quantity is increased to $1.5 \times 10^6(J)$. Between our approach and random selection, our approach always has better performance.

In Fig. 5.9, we fix the total number of events as 1000 and the number of latency constraints as 10, then change the number of base event from 0 to 40. The result in this

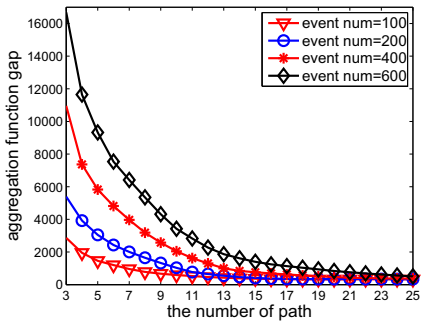


Fig. 5.10: the performance of LFBES

simulation can serves as a guideline to decide proper number of base events in the system. As shown in the figure, the energy consumption decreases sharply from 0 base event to 20 base events. In our approach, about 35% energy is saved. When the base events are more than 20, the curve decreases much more slowly. From 20 base events to 40 base events, less than 10% energy is saved. So in this system, 20 base events is a proper choice. According to this figure, our approach outperform other approaches in terms of quantity and decreasing speed.

5.6.4 The Performance of Latency Constraint Fixed Base Event Selection

LFBES is a subroutine in our solution to determine the optimal base event selection in a group of events with the same latency constraint. We have analyzed that the property of LFBES result can be analyzed by a convex non-decreasing function. Now we use the simulation to confirm it. The result is shown in Fig. 5.10.

We show the relation between aggregation gap and the number of base events. The result is repeated with different number of events of 100, 200, 400 and 600. We can clearly seen that the curve is like a convex non-decreasing function in all these situations. At beginning, the aggregation gap decreases very quickly with the increase number of base events. After a threshold, the decrease in aggregation gap is quite little. This property is very useful. First, it justifies that why we choose the LFBES to reduce the state space.

Second, we can use appropriation such as FPTAS to further reduce the complexity of the algorithm.

5.6.5 The Performance of the FPTAS-based Approach

In this section, we demonstrate the performance of the FPTAS-based approach used in base event selection. We will show that the complexity is greatly reduced using FPTAS-based approach compared with standard dynamic programming (SDP). We also compare the performance of the FPTAS-based approach with different values of K. The results can be seen in Fig. 5.11-5.13.

The performance of dynamic programming depends on several factors: the number of stages, the number of states in each stage, and the number of actions resulting in one state. Using FPTAS-based approach, we can greatly reduce the number of states needed to be investigated. We show that this improvement is useful with the increasing number of events, increasing number of latency constraints, and increasing number of base events.

As shown in Fig. 5.11, we fix the number of latency constraints as 10 and the number of base events as 20, then change the total number of events from 100 to 1000. We can see the execution time of standard dynamic programming increases very quickly, reaching $1200ms$ when there are 1000 events. The execution time of the FPTAS-based approaches increases much slower than it. For the FPTAS-based approach with a K larger than 1.005 (which means 0.5% relative error), the execution time is quite stable and always less than $400ms$. We also can see that there exists an optimal value of K considering the reduction of execution time. This value is near 1.005, and a larger value only results in marginal reduction of execution time but suffer relatively large accuracy loss. Comparing the FPTAS-based approach having $K = 1.9$ with the one having $K = 1.005$, the relative error is increased by 179 times but the execution time is only reduced by 16% when there are 1000 events.

As shown in Fig. 5.12, we fix the total number of events as 1000 and the number of base

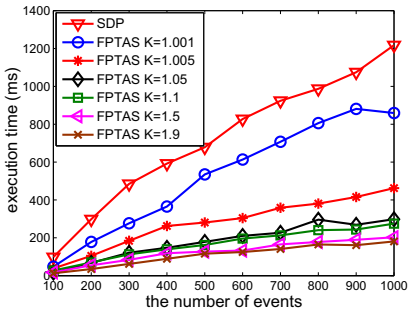


Fig. 5.11: FPTAS result with different number of events

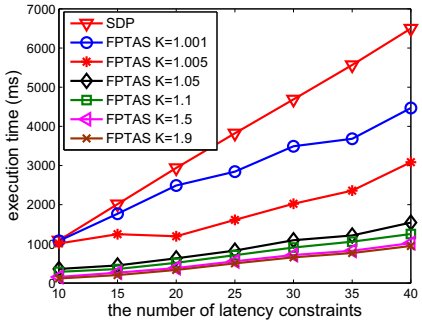


Fig. 5.12: FPTAS result with different number of latency constraints

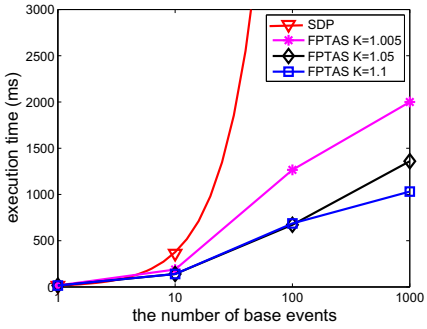


Fig. 5.13: FPTAS result with different number of base events

event as 20, then change the number of latency constraints from 10 to 40. The result is quite similar with Fig. 5.11. It shows the number of latency constraints has similar effect on the execution time. The FPTAS-based approach with a K of 1.005 saves 52% execution time compared with standard dynamic programming, when the number of latency constraints is 40. An optimal value of K also exists around 1.005.

As shown in Fig. 5.13, we fix the total number of events as 1000 and the number of latency constraints as 10, then change the number of base event among 1, 10, 100, 1000. We can see that the execution time of standard dynamic programming is exponential increased. This is true since it is a pseudo polynomial algorithm. Using the FPTAS-based approaches, we can make the complexity of the algorithm fully polynomial with a bounded accuracy loss.

In practice, the application may have fixed number of latency constraints or allow the user to specify the events with arbitrary latency constraint. According to above simulation results, we can see that in both cases the FPTAS-based approach can save much execution time. Moreover, our approach is not sensitive to the number of base events compared with standard dynamic programming, therefore it is suitable for large-scale applications.

5.7 Summary

In this chapter, we investigated the aggregation of multiple composite events with different latency constrains and aggregation functions. Latency constrain and aggregation function are two important factors impacting the structure of an optimal event aggregation tree. We proposed a three-phase solution for this problem, including base event selection phase, base aggregation tree building phase, and aggregation phase. In the base event selection phase, we select some of composite events as base events and then build their aggregation trees, while the other composites share these trees to avoid overhead including energy consumption, memory, etc. We use dynamical programming to choose these base event to ensure desirable aggregation performance for different composite events. FPTAS-based approach is further proposed for our problem to reduce the computation complexity. In the base aggregation tree building phase, we proposed the Delay Bounded Event Aggregation Algorithm (DBEA) to build the event aggregation tree for a specific composite event. This tree considers both latency constraint and aggregation function. In the aggregation phase, actual event aggregation is performed to detect composite events. Simulation results show that significant energy (up to 35% in our system) can be saved by using our algorithms.

Chapter 6

Fault-Tolerant RFID Reader Localization

In this chapter, we investigate the fault-tolerant RFID reader localization problem. We propose a new approach which can tolerate long-lasting reginal fault, an important kind of fault in RFID reader localization, and define quality index to measure the accuracy of a localization result. This chapter is organized as follows: Section 6.1 is the overview of this work. Section 6.2 describes the problem and the system models. Section 6.3 describes our solution. Section 6.4 extend our solution to network localization based on multidimensional scaling method. Simulation and experiment results are reported in Section 6.5 and Section 6.6, respectively. Finally, Section 6.7 concludes this chapter.

6.1 Overview

RFID reader localization is an important application of event inference, where the detected RFID tags are used to infer the location of the RFID reader. It is challenging since various RFID faults may occur.

We target the systems which provide localization service in a large region (e.g. a shopping mall, a warehouse, etc.), where the infrastructure provider cannot afford expensive RFID readers or on-site maintenance which may disturb normal operations. Instead, the

infrastructure provider can place a collection of cheap passive RFID tags in the region, and install readers at the user side (e.g. at shopping carts, forklifts, etc.). The readers are of a relative small number and easier to maintain in these applications. With the advance of RFID devices, we even can expect that a human being takes a phone-size RFID reader for localization in the future.

One fundamental problem of reader localization is how to handle the faults frequently occurred in the localization process. The faults can be caused by complex radio propagation, environment interference, or hardware failures. Many researchers have noticed this and proposed some countermeasures utilizing spatial/temporal redundancy [NLLP03, SWJ+05, LL06]. A tag's fault can be corrected by its neighboring tags or through multiple tag readings. However, these approaches cannot work in the situation that faults exist in a large region and last for a long time, which is called *long-lasting regional fault* in this work. Unfortunately, this kind of fault is not uncommon in RFID applications. For example, an object to be located happens to be near a metal equipment. Due to shielding effect, all the tags around the metal equipment cannot respond to the reader during the whole localization process. One more example is the localization near walls, corners or other objects. It is difficult to locate the target object in such circumstance since the identification region is deformed and quite different from that in an open region [WWT07]. Long-lasting regional fault causes serious localization error if not carefully considered.

Another problem of existing approaches is the lack of quality measurement of localization results. Since RFID faults occur frequently and the causes are complex in most of applications, a measurement representing the confidence of a localization result is important for the user. When the quality of a localization result is below a threshold, the user can take further actions. This kind of quality information is critical to designing hybrid localization

methods, in which the quality information can trigger the transition among different methods. The quality information is also useful for the network localization applications that involve multiple objects. For example, the multidimensional scaling method (MDS) [CC01] can utilize the objects' initial inaccurate locations and corresponding quality information, together with their inter-distances to perform joint optimization to increase the localization accuracy.

In this work, we first formally categorize the RFID faults in RFID reader localization. Then we propose an approach which is tolerant of long-lasting regional fault utilizing geometric knowledge of the identification region and linear second-order cone programming. We measure the quality of a localization result using *quality index* defined by us. Both 2D and 3D localization are discussed. After that, we extend our work to network localization applications where multiple objects need to be localized. We adopt MDS as the basic method, and provide the locations and quality index obtained by our approach to MDS as necessary input for further optimization. Extensive simulations and a real system are used to validate the effectiveness of our approach. In summary, this chapter makes the following specific contributions.

- We formally categorized the RFID faults in RFID reader localization.
- We proposed an effective localization approach which can tolerate long-lasting regional fault.
- We proposed quality index to measure the quality of a localization result obtained by our approach.
- We integrated our method with MDS to solve the network localization problem. The locations and especially quality index provided by our approach are validated helpful to further improve the localization accuracy.

6.2 System Model

Our work is based on the following system models about localization scenarios and RFID faults.

6.2.1 RFID Reader Localization of Individual Objects

We first describe the localization model of individual objects. For consistency, we adopt a system model similar to [WWT07, BP08]. As shown in Fig. 6.1, a hexahedron (i.e. a warehouse, a reading room, etc.) has the length, width and height of L, W, H, respectively. A set of reference tags are deployed in the floor and ceiling of the hexahedron, in a grid with the spacing of d (d is large enough to avoid the interference among tags). With an arbitrary coordinate system, the coordinates of the reference tags are known in advance. A target object carrying an RFID reader is in the hexahedron and needs to be located. The identification region of the reader is assumed to be a sphere. When projected on the ceiling or floor, the identification region is a circle with the radius of r_c and r_f, respectively. For 2D localization, the radius is simply denoted by r. We refer the projected identification region in an ideal environment as *ideal identification region* in this work. The localization depends on RFID identification process, in which the tags detected successfully by the reader are called *activated tags* and the others are called *un-activated tags*. The activated tags having both activated tags and un-activated tags as neighbors are called *border tags*. The environment is not assumed perfect and faults occur frequently.

There are two objectives in this problem: 1) to locate the target object with the minimum error; 2) to provide quality information of the localization result to the user.

Several issues need to be emphasized. First, this model is applicable to both 2D and 3D localization. In 2D localization, only the reference tags in the ceiling or in the floor are needed, but not both. Second, sphere identification region is not necessary. We first assume

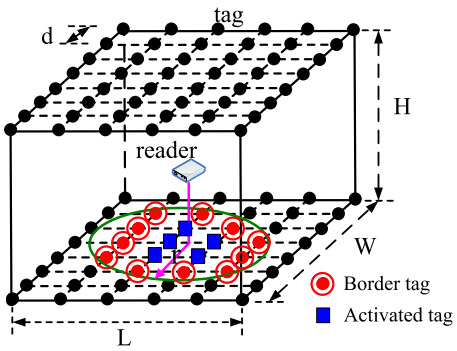


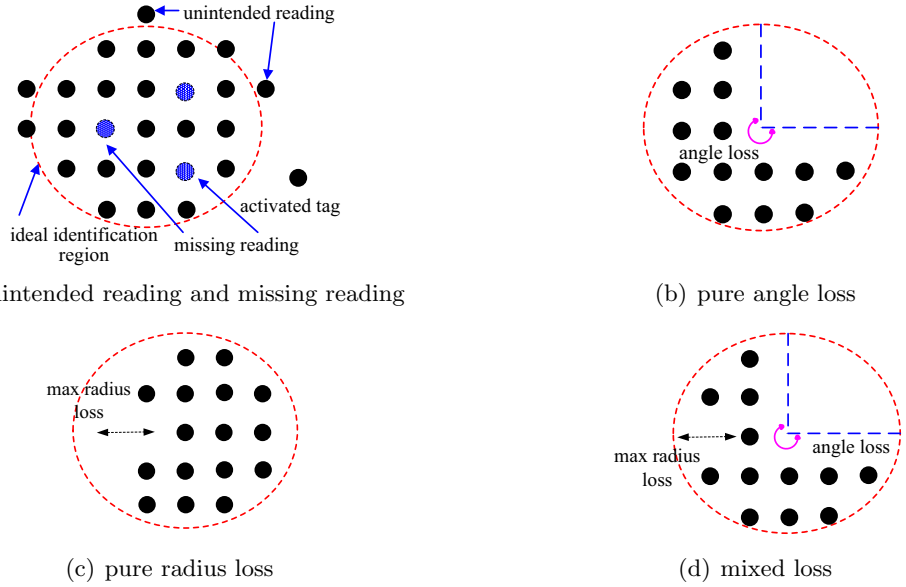
Fig. 6.1: RFID-based localization model

this for convenience in discussion and relax it in Section 6.3.6. Third, it is a range-free model. The reader only provides the information which tags are detected.

6.2.2 RFID Network Localization

In real applications, it is quite often that there are multiple objects forming a network needed to be localized. We assume that there are a number of n objects randomly distributed in a 2D area that has the length of L and the width of W. Each object utilizes an RFID reader localization method to obtain its own location and corresponding quality information, following the system model described in Section 6.2.1. Due to RFID faults, the localization results are inaccurate. Each object measures the distance from its neighbors through different techniques. For example, ranging techniques based on RSS, TOA, or TDOA leveraging various sensors equipped in the objects [WGD10], or based on adaptive transmission power of RFID reader assuming that each object is equipped with an RFID reader and also an RFID tag [AAHI10]. All the information is gathered to a central server and an optimization process is performed there. The objective is to determine all the locations of these objects with minimum average error.

Different with existing network localization methods [WZYB08, SW11], we do not simply classify objects into anchor nodes whose location are determined and non-anchor nodes



(a) unintended reading and missing reading

(b) pure angle loss

(c) pure radius loss

(d) mixed loss

Fig. 6.2: Different RFID faults in the localization process

whose location are not determined. Instead, each object in our model has an initial location and corresponding quality index, and thus is more generic.

6.2.3 RFID Faults

RFID faults are classified into two categories: *unintended reading* and *missing reading*. As shown in Fig. 6.2(a), unintended reading denotes that the tags outside the ideal identification region are read, while missing reading denotes that the tags in the ideal identification region are not read.

For unintended reading, the solution is based on the assumption that the further a tag is from the reader, the lower probability it has to be detected. By reading the tags for multiple times [BHE00, SWJ+05], or assigning different weights to the tags [NLLP03, LL06], the effect of unintended reading (especially the readings far from the ideal identification region) can be effectively eliminated.

Missing reading is further classified in temporal dimension and spatial dimension. In

temporal dimension, there are *intermittent fault* and *long-lasting fault*. In spatial dimension, there are *spot fault* and *regional fault*. The combination of long-lasting fault and regional fault is called *long-lasting regional fault*, which denotes that faulty tags are dominant in a region and fail to respond to the reader in a continuous time duration.

If a fault is not a long-lasting regional fault, it can be handled similarly using above-mentioned multiple times reading method based on spatial/temporal redundancy. However, long-lasting regional fault is difficult to deal with because the fault is continuous in both spatial dimension and temporal dimension. Unfortunately, this fault is not rare in the RFID applications. In this work, long-lasting regional fault is our major concern. Specifically, we assume that the faults last during the whole identification process. And in the rest of this section, we discuss more about the regional fault.

In the localization process, we use activated tags to infer the reader's identification region. We define *activated region* to denote the inferred region:

Definition 1 (activated region): Given an activated tag set A, the activated region inferred by A is defined as the convex hull of the activated tags, denoted by $conv(A)$.

Due to RFID faults, some tags may not be identified, so the activated region may be different from the ideal identification region. We build a polar coordinate system with the pole of the reader. Observing the difference between the ideal identification region and the activated region, we have three basic types of regional fault: *pure angle loss*, *pure radius loss* and *mixed loss*. More complex faults are the combinations of them. The examples of them can be seen in Fig. 6.2(b)-6.2(d), and the formal definitions are as follows:

Definition 2 (pure angle loss): Compared with ideal identification region $\rho = \phi(\theta)(0 \leq \theta \leq 2\pi)$, an activated region with pure angle loss is defined as $\rho = \phi(\theta)(\theta' \leq \theta \leq \theta'', 0 \leq \theta' \leq \theta'' \leq 2\pi)$. The angle loss is defined as $2\pi - (\theta'' - \theta')$.

Definition 3 (pure radius loss): Compared with ideal identification region $\rho = \phi(\theta)(0 \leq$

$\theta \leq 2\pi$), an activated region with pure radius loss is defined as $\rho = \phi'(\theta)(0 \leq \theta \leq 2\pi, 0 < \phi'(\theta) \leq \phi(\theta))$. The max radius loss is defined as $max_\theta(\phi(\theta) - \phi'(\theta))$, and the affected angle is defined as $\int_{\phi'(\theta) \neq \phi(\theta)} d\theta$.

Definition 4 (mixed loss): Compared with ideal identification region $\rho = \phi(\theta)(0 \leq \theta \leq 2\pi)$, an activated region with mixed loss is defined as $\rho = \phi'(\theta)(\theta' \leq \theta \leq \theta'', 0 \leq \theta' \leq \theta'' \leq 2\pi, 0 < \phi'(\theta) \leq \phi(\theta))$. The angle loss is defined as $2\pi - (\theta'' - \theta')$, the max radius loss is defined as $max_\theta(\phi(\theta) - \phi'(\theta))(\theta' \leq \theta \leq \theta'')$, and the affected angle is defined as $\int_{\phi'(\theta) \neq \phi(\theta)} d\theta$.

6.3 RFID Reader Localization

In this section, we propose an RFID reader localization approach which can tolerate long-lasting regional fault and provide corresponding quality information. This approach is for the localization of individual objects. We first invetigate it in 2D area in Section 6.3.1-6.3.6, and then extend it to 3D area in Section 6.3.7.

6.3.1 Basic Method

It is known that the ideal identification region of a reader in the ceiling/floor of the hexahedron is a circle centered with the reader. The purpose of localization is to find the circle center with the help of activated tags. The tags are processed firstly to filter out unintended readings. Then we discuss how to handle missing reading as follows.

In existing works such as the active scheme of Wang et al. [WWT07] (Wang's active scheme for short) and the centroid method [NLLP03, BP09], the activated region is just viewed as the reader's ideal identification region. The tags with intermittent fault or spot fault are corrected into activated tags utilizing spatial/temporal redundancy. However, this cannot work in the presence of long-lasting regional fault, making the localization result deviate from the real location (see Fig. 6.3). To overcome this drawback, we propose a method which makes all activated tags included in the ideal identification region. We

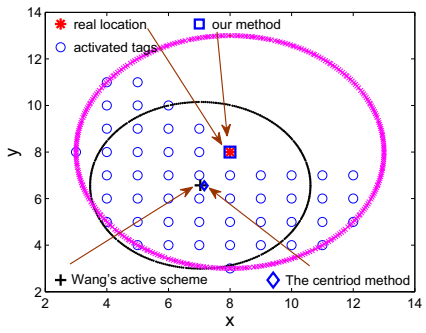


Fig. 6.3: An example to illustrate the difference between ATI and existing approaches. There is a regional fault with angle loss of $3\pi/4$.

call this method ATI (Activated Tag Included Method). ATI fully utilizes the geometric knowledge of the identification region hence has a better performance.

Let activated tags have the coordinates $\{(x_i, y_i) \mid i = 1...n\}$. ATI is formulated as follows:

$$\begin{aligned} &min\ r_f \\ &st.\ (x_i - x_f)^2 + (y_i - y_f)^2 \leq r_f^2 \ \ (i = 1...n) \end{aligned} \tag{6.1}$$

The above formulation can be transformed to be a linear second order cone programming. Let I be the identity matrix, and $w_i = (w_i^1, w_i^2, w_i^3)^T$ be a variable vector. The problem is to determine $x_f, y_f, r_f, w_i (1 \leq i \leq n)$ such that

$$\begin{aligned} min \quad & r_f \\ st. \quad & I \cdot w_i = \begin{pmatrix} x_i - x_f \\ y_i - y_f \\ r_f \end{pmatrix}, \quad 1 \leq i \leq n \\ & \sqrt{(w_i^1)^2 + (w_i^2)^2} \leq w_i^3, \quad 1 \leq i \leq n \end{aligned} \tag{6.2}$$

We obtain the result using SDPT3 4.0 [TTT99] which is a Matlab software package to solve the linear cone programming problem with interior point method.

An example of ATI and other methods is shown in Fig. 6.3. The reader's location is of a great deviation when determined by Wang's active scheme or the centroid method, but

quite accurate when determined by ATI.

6.3.2 Accuracy Analysis in Pure Angle Loss

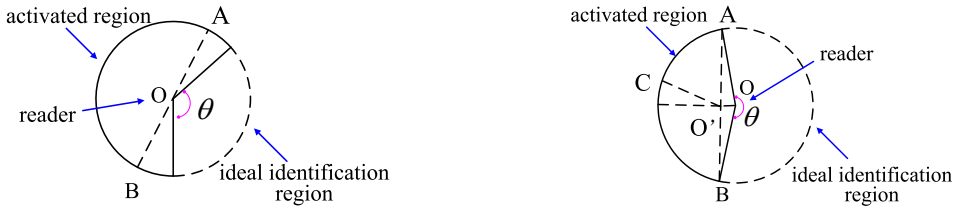
We analyze the effectiveness of ATI in the presence of pure angle loss. Three theorems are obtained, where the radius of the ideal identification region is denoted by r.

Theorem 3. *In pure angle loss, if the angle loss is less than π, the circle determined by ATI has the radius at least r.*

Proof. As shown in Fig. 6.4(a), the activated region is a circular sector, and the angle loss is θ. Since $\theta < \pi$, we always can find a diameter AB of the ideal identification region included in the activated region. $|AB| = 2r$. When ATI determines a circle with minimum radius say r' including all activated tags, point A and B must be included in that circle. According to the property that the length of a diameter is the maximum length between any two points in the circle, we have $2r' \geq |AB| = 2r$, so $r' \geq r$. □

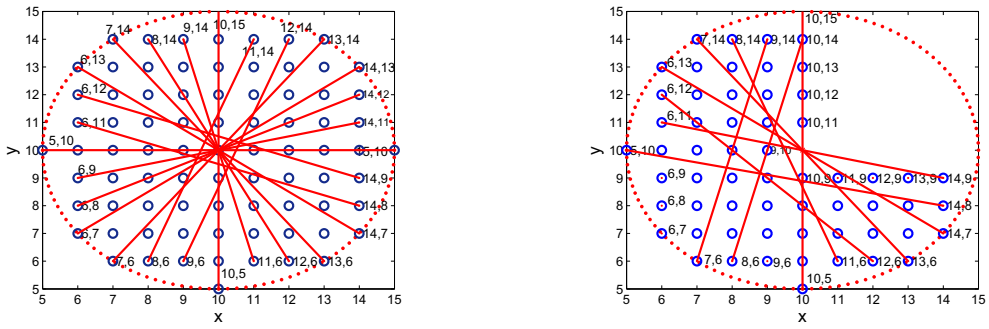
Theorem 4. *In pure angle loss, if the angle loss is not less than π, a circle with the center rather than the reader's exact location and the radius less than r can be found as a feasible solution of ATI.*

Proof. As shown in Fig. 6.4(b), the circular sector $\widehat{AOB}$ is the activated region and the angle loss is θ. We connect point A and B using a line and get the midpoint O' as the new circle center. Since $\theta \geq \pi$, we have $|AO'| = |BO'| < |AO| = r$. For an arbitrary point C in arc AB, we also have $|CO'| < r$. So at least we can find a circle with the center of O' and the radius less than r to include all activated tags. □



(a) angle loss $\theta < \pi$ (b) angle loss $\theta \geq \pi$

Fig. 6.4: Different situations in pure angle loss



(a) ideal identification region

(b) angle loss of $\pi/2$

Fig. 6.5: Key pairs and quality index in a) ideal identification region b) an activated region with pure angle loss.

Theorem 5. *In pure angle loss, ATI can determine the reader's exact location if and only if the angle loss is less than* π*.*

Proof. if the angle loss is less than π, according to Theorem 3, the circle determined by ATI has a radius of at least r. Considering that the reader's exact location is also a candidate of the circle center, ATI finally returns a circle centered at the reader's exact location and with the radius of r. On the other side, if ATI can determine the reader's exact location, we can prove the statement that the angle loss is less than π by contradiction. Assuming that the angle loss is not less than π, according to Theorem 4, we can find a circle whose center is not the reader's exact location as the solution. This contradicts the precondition that the reader's exact location can be determined. □

6.3.3 Quality Index in Pure Angle Loss

From previous analysis, the accuracy of a localization result depends on the angle loss. In fact, the angle loss can be viewed equivalently in another way. We observe how many diameters of the ideal identification region still exist in the activated region. When angle loss is less than π, we always can find a diameter of this kind in the activated region (Fig. 6.4(a)); while when angle loss is not less than π, we cannot do that (Fig. 6.4(b)). When the angle loss is less, we can find more such diameters, and vice verse. Motivated by this, we propose

Algorithm 11: Quality Index Algorithm

```
1  qualityIndex = 0
2  put all border tags in N
3  foreach A ∈ N do
4  |   find tag B such that |AB| is maximized (B ∈ N)
5  |   if |AB| == 2r then
6  |   |   remove A and B from N
7  |   |   qualityIndex++
8  |   end
9  endfch
10 return qualityIndex
```

to measure the quality of a localization result using the number of such diameters in the activated region. We then define the quality index of a localization result as follows:

Definition 5 (key pair): In a localization process, two activated tags are defined as a key pair if they are the end points of a diameter of the ideal identification region.

Definition 6 (quality index): The number of key pairs is defined as the quality index of a localization result.

An example of key pairs and quality index is shown in Fig. 6.5. In that figure, two tags connected by a line form a key pair, and quality index is defined as the number of key pairs. We use quality index to represent the confidence of a localization result with pure angle loss. Intuitively this is reasonable since quality index denotes angle loss.

We propose Algorithm 11 to calculate the quality index. The algorithm first initiates *qualityIndex* to 0 and obtains all border tags in N (line 1-2). For each tag in N, say A, a process is run as follows (line 3-9). The algorithm finds a tag B which has the maximum distance from A (line 4). Then the distance is compared with $2r$. If the equality holds, these two tags form a key pair. Remove these two tags and increase *qualityIndex* (line 5-8). Since the diameter is the maximum distance between two points in a circle, the correctness and convergence of this approach are easy to prove. The worst time complexity of this algorithm is $O(n^2)$.

6.3.4 Considering Grid Placement

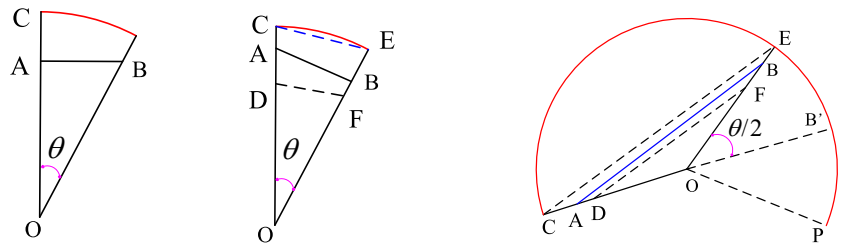
In previous two sub-sections, we discussed the solution in theory and assumed that the border of ideal identification region can be perfectly inferred by the border tags. This assumption only holds in quite dense placement of tags. However, this placement is not practical because not only the tag cost is high, but the near-filed effect among tags can no longer be ignored and affect the radio transmission. In our model, we place the tags in a grid with constant spacing. The spacing causes errors in the localization result and the quality index. Here we discuss these errors.

We first compare the border tags with the real border of ideal identification region. The difference is the error in a localization result caused by grid placement. We have Theorem 6, which will be validated by simulations.

Theorem 6. *The error of a localization result of ATI in the grid placement, is always less than d. And the relative error is always less than d/r.*

We focus on the error of quality index caused by grid placement. Since the border tags may deviate from the border of ideal identification region, the length of a diameter, also assumed by Algorithm 11(line 5), may be shorter than $2r$. We call this kind of diameter *degraded diameter*. In the following, we calculate the lower bound of the degraded diameter's length and then revise Algorithm 11 to take this into account.

At first glance, the length of the shortest degraded diameter seems to be $2(r - d)$.



(a) (b) (c)

Fig. 6.6: Accuracy analysis considering the grid placement.

However, this is not true. As shown in Fig. 6.6(c), when a border tag A selects its partner say B to form a key pair, B may deviate from the ideal location B' due to the spacing in grid placement. So the length of degraded diameter AB depends on the angle $\angle BOB'$. Assuming that B and P are the two border tags near B', A choose B to form the key pair only if $|BB'| \leq |PB'|$. Let the central angle $\angle BOP = \theta$, we have $\angle BOB' \leq \theta/2$. So in the following, we calculate the maximum central angle θ determined by two consecutive border tags before we discuss the length of degraded diameters.

As shown in Fig. 6.6(a) and 6.6(b), A and B are the consecutive border tags. O is the location of the reader. The central angle is $\theta = \angle AOB$. The arc represents the border of ideal identification region. There are two cases of θ due to different relations between two consecutive border tags.

1) Case 1: as shown in Fig. 6.6(a), the consecutive border tags are in the same horizonal or vertical line, then $|AB| = d$

$$\begin{aligned} \theta &= \arctan(|AB|/|AO|) \\ &< \arctan(d/(r-d)) \end{aligned} \tag{6.3}$$

2) Case 2: as shown in Fig. 6.6(b), the consecutive border tags are in the diagonal line, then $|AB| = \sqrt{2}d$. With law of cosines, we have

$$\begin{aligned} \cos\theta &= \frac{|OA|^2 + |OB|^2 - |AB|^2}{2|OA||OB|} \\ &= \frac{|OA|^2 + |OB|^2 - 2d^2}{2|OA||OB|} \\ (r-d < |OA| \leq r, r-d &< |OB| \leq r) \end{aligned} \tag{6.4}$$

Due to the symmetry structure of this equation, the minimum $\cos\theta$ is got when $|OA| = |OB|$:

$$
\begin{aligned}
min\cos\theta = \frac{2|OA|^2 - 2d^2}{2|OA||OA|} &= 1 - \frac{d^2}{|OA|^2} \\
&> 1 - \frac{d^2}{(r-d)^2} \\
\theta &< \arccos(1 - \frac{d^2}{(r-d)^2})
\end{aligned} \tag{6.5}
$$

The result in Eq. 6.5 subsumes the one in Eq. 6.3.

As shown in Fig. 6.6(c), the locations of A and B are bounded by $|CD| = d$ and $|EF| = d$ respectively, $CE//DF$, the length of degraded diameter $|AB|$ can be calculated as follows:

$$
\begin{aligned}
|AB| \geq |DF| &= \frac{r-d}{r}|CE| \\
&= 2(r-d)\cos(\frac{\theta}{4})(\theta < \arccos(1 - \frac{d^2}{(r-d)^2}))
\end{aligned} \tag{6.6}
$$

Then we have the following theorem:

Theorem 7. *The length of a degraded diameter is at least* $minDD = 2(r-d)\cos(\frac{\theta}{4})(\theta < \arccos(1 - \frac{d^2}{(r-d)^2}))$.

Theorem 7 gives the lower bound of a degraded diameter's length. Based on it, we modify Algorithm 11 to calculate quality index. We relax the condition that two activated tags form a key pair only if the distance between them is not less than $2r$. The threshold $2r$ is changed to $minDD$. However, this relaxation makes the computation of quality index not polynomial solvable. The following is an equivalent formulation and with the complexity of NP-hard. x_{ij} denotes whether tag i and tag j form a key pair ($x_{ij} = 1$) or not ($x_{ij} = 0$). N is the set of border tags. The objective is to find the maximum number of key pairs subject to that each tag can participate in at most one key pair.

$$
\begin{aligned}
&max \sum_{(i,j)\in N} x_{ij} \\
&st. \sum_{(i,j)\in N} x_{ij} + \sum_{(j,i)\in N} x_{ij} \leq 1 \\
&x_{ij} \in \{0,1\}
\end{aligned} \tag{6.7}
$$

Algorithm 12: Revised Quality Index Algorithm

```
1  qualityIndex = 0
2  put all border tags in N
3  threshold = 2r
4  while N ≠ ∅ && threshold > minDD do
5  |  foreach A ∈ N do
6  |  |  find tag B such that |AB| is maximized (B ∈ N)
7  |  |  if |AB| == threshold then
8  |  |  |  remove A and B from N
9  |  |  |  qualityIndex++
10 |  |  end
11 |  |  record all calculated distance |AB| in S
12 |  endfch
13 |  threshold = max S
14 end
15 return qualityIndex
```

Since the possible length of a degraded diameter is between $2r$ and $minDD$, we propose an iterative heuristic algorithm to compute quality index as shown in Algorithm 12. We set the threshold of a diameter's length first as $2r$ (line 3), and calculates the key pairs using similar processing in Algorithm 11(line 5-10). All the lengths of key pair candidates (may less than $2r$) are recorded in a set S (line 11). After that, the threshold is changed to the maximum value in N (line 13). The process repeats until the threshold is less than $minDD$ or all tags are considered (line 4). In the simulation section, we will see that the performance of this algorithm is quite desirable. The worst time complexity of this algorithm is $O(n^3)$.

Finally, we have further discussion about the spacing of the grid. In our work, we focus on that given the spacing of a grid how to calculate the localization error and quality index in various faults. This work can be extended to determine the spacing of the grid in specified environment that the required localization accuracy and the faults expected to be happened are known. The calculation of localization errors and quality index in this section can also be applied for this purpose.

6.3.5 Extension for Pure Radius Loss and Mixed Loss

We further extend the ATI method to tolerate pure radius loss and mixed loss.

In pure radius loss, if a key pair exists in the activated region, ATI also can find the exact location of the reader. The difference between it and corresponding pure angle loss (its angle loss equals to the affected angle of pure radius loss) is that the activated region of the former is larger than the latter. The previous defined quality index is still reasonable since the quality index in this situation is larger than that in corresponding pure angle loss. If a key pair cannot be found in the activated region, the situation is quite complex. However, one special case is easy to solve: the radius loss in all angles is a constant value, say a. ATI again can find the exact location of the reader. The quality index calculation needs to be revised since, if a is sufficient large, no key pair can be found according to Definition 5. We revise the threshold of a key pair's length from $2r$ to $2(r-a)$. Further revision of threshold like in Algorithm 2 can also be made to consider the grid placement.

Inspired by the two cases discussed above, we propose a simple method to define the quality index for the general case of pure radius loss.

$$a \cdot \frac{max\phi'(\theta)^2}{r} - b \cdot std(\phi'(\theta)) + c \cdot count(max\phi'(\theta)) \tag{6.8}$$

The quality index depends on three parts. The first part describes the maximum radius loss in all angles. The second part considers the distribution of radius loss. The third part is complementary to the second part and highlights the number of diameters with the length of $max(\phi'(\theta))$. a, b, c are system parameters used to adjust the weights of these three parts and their values are learned through experiments. Other forms of quality index definition taking these three factors into account are also possible and left as future work.

Mixed loss is a combination of pure angle loss and pure radius loss. The quality index defined in Eq. 6.8 is still applicable.

6.3.6 Analysis of Real Identification Region

We now consider the identification region of real RFID readers, which may not be a perfect circle due to different signal propagation attenuation amounts and different antenna gains in different directions. Our analysis is based on [WWT07], where the identification region is characterized by an upper bound (R_u) and a lower bound (R_l) of the reader's signal transmission range, and a Degree of Irregularity (DOI) denoting maximum variation of the reader's transmission range per unit degree.

We have two methods to make our solution applicable to a real identification region. The first method is proposed in [WWT07], to use a low-cost antenna array with multiple radiation elements in different directions to minimize DOI. If only common antenna at hand, the authors suggest to rotate the antenna to achieve similar effect of the antenna array. During this design, the identification region is quite approximate to a circle. The other method is to compare the ideal identification region with the circle determined by R_u. The process is like our analysis in the regional fault. If the difference is not so significant (e.g. angle loss $< \pi$), we just use this circle to perform localization. Otherwise, we conservatively base on the circle determined by R_l to perform localization.

6.3.7 3D RFID Reader Localization

In previous sub-sections, we discussed our solution for 2D localization. Now we demonstrate how to use it for 3D localization.

According to the system model in Section 6.2.1, both the ceil and floor are placed with a grid of RFID tags for 3D localization. Based on the ATI method, we can determine a target object's projected location in the ceiling and floor. After that, we calculate the target object's 3D location according to the geometric relations between the projected locations and the 3D location. The detailed approach follows [WWT07]. Assuming that the circle

determined by ATI in the ceiling is centered at (x_c, y_c, H) and with the radius of r_c, and in the floor is centered at $(x_f, y_f, 0)$ and with the radius of r_f. Then we can calculate the target object's 3D coordinate (x, y, z) as follows:

$$x = \frac{x_c + x_f}{2} \tag{6.9}$$

$$y = \frac{y_c + y_f}{2} \tag{6.10}$$

$$z = \frac{max(r_c, r_f)^2 - min(r_c, r_f)^2 + H^2}{2H} \tag{6.11}$$

The analysis of quality index is similar with the error analysis of the localization method in [WWT07]. Assuming that the quality index computed by ATI in the ceiling is q_c and in the floor is q_f, we get the joint average quality index q_{xy} for the coordinates (x, y) based on error propagation theory:

$$q_{xy} = \sqrt{q_x^2 + q_y^2} \tag{6.12}$$

Similarly, we get the quality index for the coordinate z:

$$q_z = \frac{1}{H}\sqrt{r_c^2 q_x^2 + r_f^2 q_y^2} \tag{6.13}$$

Finally we obtain the quality index q of a localization result in 3D area as

$$q = \sqrt{q_{xy}^2 + q_z^2} \tag{6.14}$$

6.4 RFID Network Localization

After investigating the localization of individual objects, we turn to RFID network localization which involves multiple objects. Although the localization accuracy of a single object is frequently affected by environmental factors and human activities, it is probability that some other objects in the neighborhood are localized quite accurately. If the location information of all neighboring objects and the distance relations among these objects are taken into account, a more accurate result is possible. We will formulate this problem and

solve it in this section. The localization results and quality indices obtained by the ATI method for individual objects are key inputs for this problem.

6.4.1 Problem Formulation

Based on the system model described in Section 6.2.2, we formulate the problem as follows. There is a number of n objects $x_i (1 \leq i \leq n)$ which need to be located. $(x_i^1, x_i^2)^T$ denotes the coordinates of object x_i in a 2D area. The distance $d_{i,j}$ between two objects x_i and x_j is known if $(i,j) \in N_x$. N_x is a node-pair set in which the distance between the nodes of each node pair can be obtained in the application. For example, if the objects use radio RSS (received signal strength) to measure the distance from each other, and the radio communication range is r_d, then $N_x = \{(i,j) : \|x_i - x_j\| \leq r_d\}$. Usually, $d_{i,j}$ is not accurate and includes measurement noises. Each object x_i performs the ATI method to localize itself, with the result $\hat{x_i}$, and quality index q_i. If ATI is not used by some objects, just set $\hat{x_i}$ as arbitrary value and q_i as 0. Each object is also known roughly within an area with the center of $\bar{x}_i$ and the radius of r_i according to the application context. If such information is lacked, just set $\bar{x_i}$ as arbitrary value and r_i as a sufficient large value. The objective is to determine $(x_i^1, x_i^2)^T, i = 1, 2....n$ such that the aforementioned constraints can be satisfied at well as possible.

Generally, it is a variant of MDS problem. We define the objective function and have the following formulation:

$$\begin{aligned} \min \quad & \sum_{1 \leq i \leq n} q_i \|x_i - \hat{x_i}\|^2 + \sum_{(i,j) \in N_x} (u_{i,j} + v_{i,j}) \\ s.t. \quad & \|x_i - x_j\|^2 + u_{i,j} - v_{i,j} = d_{i,j}^2, \forall (i,j) \in N_x \\ & \|x_i - \bar{x}_i\|^2 \leq r_i^2, 1 \leq i \leq n \\ & u_{i,j}, v_{i,j} \geq 0, \forall (i,j) \in N_x \end{aligned} \tag{6.15}$$

This formulation does not classify the nodes into anchor nodes whose actual locations are already known and non-anchor nodes whose locations need to be determined as in [SY07]. Instead, we use a more generic description of nodes based on the quality index. This can handle the case where no anchor nodes exist but several nodes are localized with relatively high accuracy, and also the case where nodes are localized with different accuracies. Similar idea is also adopted in [CPI04], but it does not discuss how to obtain such quality information. ATI bridges this gap. We refer to the method combining ATI and MDS as ATI-MDS method.

6.4.2 Solution

We follow the idea of [SY07] to transfer this problem into a SDP (semidefinite programming) problem and then solve it. Following symbols are used in this section: $Tr(A)$ denotes the trace of matric A, $< A, B >$ denotes the inner product of matrices A and B, which is defined as $Tr(A^T B)$. $rank(A)$ denotes the rank of matric A. I and 0 denotes the identity matrix and the zero matrix with proper size. e_i denotes a vector whose elements are all zero except the i^{th} element is 1. Z_{12} denotes the upper-left 2×2 principal sub-matrix of Z that is composed of the cross elements at the 1^{st} and 2^{nd} row and the 1^{st} and 2^{nd} column of Z.

Let $X = (x_1, x_2..., x_n)$, $Z = \begin{pmatrix} I & X \\ X^T & X^T X \end{pmatrix}$, we have the follows:

$$\begin{aligned} ||x_i - x_j||^2 &= (e_i - e_j)^T X^T X (e_i - e_j) \\ &= Tr((e_i - e_j)(e_i - e_j)^T X^T X) \\ &=< \begin{pmatrix} 0 \\ e_i - e_j \end{pmatrix} \begin{pmatrix} 0 \\ e_i - e_j \end{pmatrix}^T , Z > \end{aligned}$$

Similarly, we have

$$||x_i - \hat{x}||^2 =< \begin{pmatrix} \hat{x}_i \\ -e_i \end{pmatrix} \begin{pmatrix} \hat{x}_i \\ -e_i \end{pmatrix}^T, Z >$$

$$||x_i - \bar{x}||^2 =< \begin{pmatrix} \bar{x}_i \\ -e_i \end{pmatrix} \begin{pmatrix} \bar{x}_i \\ -e_i \end{pmatrix}^T, Z >$$

Replacing corresponding items in Eq. 6.15, we obtain an equivalent equation with the variable Z. Then the problem is to find a matric Z which is the solution of the equivalent equation and satisfy the constraints:

$$rank(Z) \leq 2$$

$$Z \succeq 0$$

$$Z_{12} = I$$

It is a non-convex problem due to the constraint $rank(Z) \leq 2$. We take a SDP relaxation by removing this constraint and have the following linear SDP problem. It can be solved by using matlab software SDPT3 4.0 [TTT99].

$$\begin{aligned} \min \quad & \sum_{i \in N_x} q_i < \begin{pmatrix} \hat{x}_i \\ -e_i \end{pmatrix} \begin{pmatrix} \hat{x}_i \\ -e_i \end{pmatrix}^T, Z > + \\ & \sum_{i,j \in N_x} (u_{i,j} + v_{i,j}) \\ s.t. \quad & < \begin{pmatrix} 0 \\ e_i - e_j \end{pmatrix} \begin{pmatrix} 0 \\ e_i - e_j \end{pmatrix}^T, Z > + \\ & u_{i,j} - v_{i,j} = d_{i,j}^2, \forall (i,j) \in N_x \\ & < \begin{pmatrix} \bar{x}_i \\ -e_i \end{pmatrix} \begin{pmatrix} \bar{x}_i \\ -e_i \end{pmatrix}^T, Z >\leq r_i^2, i \in N_x \\ & u_{i,j}, v_{i,j} \geq 0, \forall (i,j) \in N_x \\ & Z \succeq 0 \\ & Z_{12} = I \end{aligned} \tag{6.16}$$

6.4.3 Discussion

This method utilizes the localization results of individual objects and their pairwise distances to increase the overall localization accuracy. Since the pairwise distance information is sparse (depends on $|N_x|$), the localization results especially the quality index are quite useful to guide the results to the real locations. Further improvements also can be made for this method. For speeding up the computing process, we can follow the work [WZYB08] to further relax the SDP problem to reduce the search space. We also can follow the work [CPI04] to make our method distributed, by reformulated the objective function as a sum of several localized objective functions. More application-specific constrains are also can be considered to improve the accuracy.

6.5 Simulations

This section presents our simulation results with two purposes. The first one is to validate the theorems stated in previous sections. The second one is to compare our approach with existing approaches in different situations. We choose Wang's active scheme [WWT07], and the centroid method [BHE00] for comparison. For Wang's active scheme, we adopt its full border method which has the best reported performance. We first check the performance in the single object scenario (both 2D and 3D), and then further check the performance in the multiple objects scenario.

For the 2D localization, without loss of generality, we place the tags in a grid with the spacing of $1m$ and varies the radius of the ideal identification region. Considering the effects of grid placement, we carry out simulations by putting the reader at different places in a grid cell, and calculate the average value. Specifically, we enumerate all the places in a grid cell in a horizontal and vertical step of $0.1m$. In some simulations, we also show the results only at the grid intersections for comparison. The identification range of a reader is set to

$10m$. For the 3D localization and network localization, we will describe their simulation parameters in corresponding sub-sections.

6.5.1 Errors Caused by Grid Placement

We first check the errors of location and quality index caused by grid placement. The results are to validate Theorem 6 and Theorem 7.

The relative error of location caused by grid placement is shown in Fig. 6.7 and 6.8. In the former figure, the reader is placed at the grid intersection $(20, 20)$, and in the latter figure, the reader's location is evenly distributed in the grid cell encompassed by $(20, 20)$, $(20, 21)$, $(21, 20)$ and $(21, 21)$. Due to the symmetry structure of grid, other locations in the grid share the same results. In both cases, we enumerate all the border tags to check their differences from the border of the ideal identification region, which is the error caused by grid placement. The relative error can be calculated correspondingly. We show the maximum, mean and standard deviation of relative error in the figures. It is shown that the upper bound of relative error stated in Theorem 6 (d/r) is strictly obeyed, especially when the reader is in a grid cell (Fig. 6.8). When the reader locates at grid intersections, the curve is zigzag and some kind of periodic (not in strict sense) (Fig. 6.7). This can be explained by examining the error computation. Since the tags are restricted to grid intersections, the error equation has a floor function in it. When expanding the floor function with Fourier series, there is a sine function, so the result is some kind of periodic.

The error of quality index caused by grid placement depends on the maximum central angle formed by two consecutive border tags (θ in Theorem 7). We compare the maximum central angle calculated by our approach with the uniform estimation used in [WWT07] which considers all border tags uniformly distributed in the circle border. The result is shown in Fig. 6.9 with different radius r of ideal identification region. When $r \leq 6$, our result can serve as the upper bound for all locations while the uniform estimation cannot.

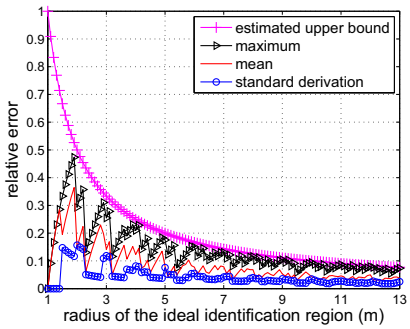


Fig. 6.7: Localization relative error caused by grid placement (the reader is at grid intersections)

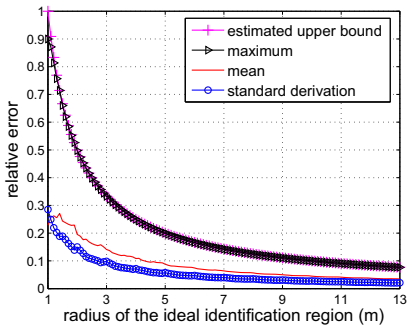


Fig. 6.8: Localization relative error caused by grid placement (the reader is in a grid cell)

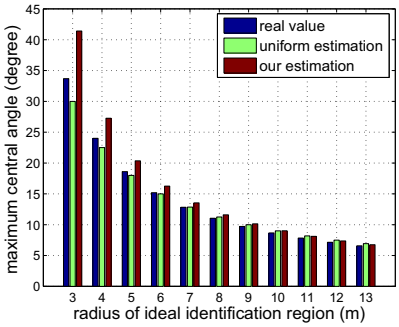


Fig. 6.9: The maximum central angle determined by two consecutive border tags

When $6 < r \leq 10$, uniform estimation also can serve as the upper bound and is stricter than our result. However, the result reverses when $r > 10$. As a general estimation of maximum central angle in different cases, the result of our method is more accurate than that of the uniform estimation method.

6.5.2 Performance in Pure Angle Loss

The localization accuracy of ATI is compared with that of Wang's active scheme and the centroid method in pure angle loss. The results are shown in Fig. 6.10 and 6.11.

When the reader locates at grid intersections, ATI can find the exact location of the reader (error$<0.01m$) if the angle loss is not more than 140 degrees, while Wang's active scheme and the centroid method cannot do that (Fig. 6.10). The result is less than 180

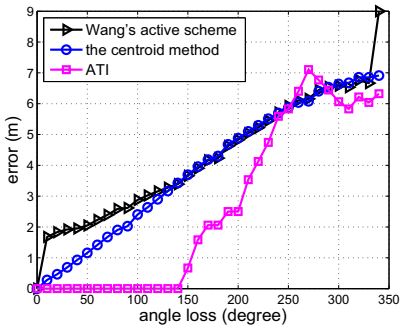


Fig. 6.10: Localization result in pure angle loss (the reader is at grid intersections)

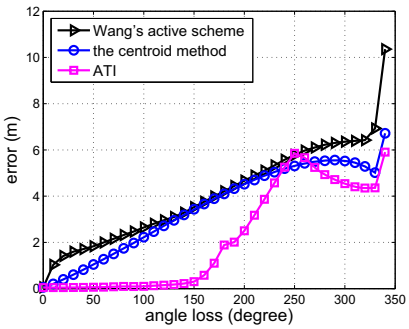


Fig. 6.11: Localization result in pure angle loss (the reader is in a grid cell)

degrees proved by Theorem 3, due to the error caused by grid placement. When the reader is distributed in a grid cell, the result is shown in Fig. 6.11. At the first stage, the result is quite accurate (error$<0.5m$), and when the angle loss is more than 140 degrees, the error sharply increases. It finally increases to be slightly more than ($<0.55m$) that of the centroid method when the angle loss is around 250 degrees. After that, the error of ATI drops below that of Wang's active scheme and the centroid method again. For a common angle loss of 90 degrees, the error of ATI is 3.8% of Wang's active scheme and 4.7% of the centroid method. We also take similar simulations using other radius values. When the radius is 5 and angle loss is 90 degrees, the error of ATI is 10.3% of Wang's active scheme and 12.1% of the centroid method.

Fig. 6.12 shows the quality index in pure angle loss. The ideal value is computed assuming border tags evenly locate at the border of ideal identification region. The theoretical value of quality index in the grid placement is computed when the location of the reader is already known. This value is always smaller than the ideal value due to the error caused by grid placement. From the figure, we can see that our result is quite close to the theoretical value in most of cases. Slight deviation occurs around 180 degrees but is still acceptable.

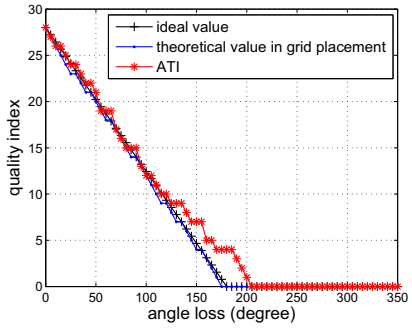


Fig. 6.12: Quality index in pure angle loss calculated by Revised Quality Index Algorithm

6.5.3 Performance in Pure Radius Loss/Mixed Loss

We continue to check the performance of different methods in pure radius loss and mixed loss. If a key pair still exists, the result is similar to that in pure angle loss. We show a result of pure radius loss with the affected angle of 150 degrees in Fig. 6.13 to validate this. When key pair does not exist, we show a result of mixed loss with the affected angle of 270 degrees in Fig. 6.14. In the former case, with respect to the affected angle, the maximum radius r of the identification region varies from $4m$ to $10m$, and the radius are evenly distributed between r and $r-3$. In the latter case, the maximum radius r varies in the same range, but the radius are evenly distributed between 0 to r. In both cases, the performances of Wang's active scheme and the centroid method are quite close. Compared with these two methods, the accuracy of ATI is improved by 55% to 80% when a key pair exists (Fig. 6.13), and 8% to 29% when key pair does not exist (Fig. 6.14). The difference between ATI and these two methods is not so significant when angle loss is large (e.g. 270 degrees as in Fig. 6.14). Considering moderate faults in practical situations, ATI can achieve desirable performance.

Eq. 6.8 is proposed to compute the quality index in pure radius loss and mixed loss. Here we check whether the computed quality index matches the ground truth. As shown in Fig. 6.15, we compare the quality index in pure radius loss with that in corresponding

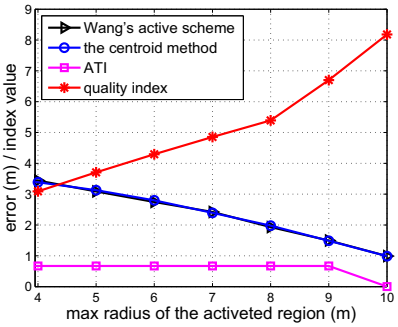


Fig. 6.13: Localization result and quality index in pure radius loss (affected angle=150 degrees)

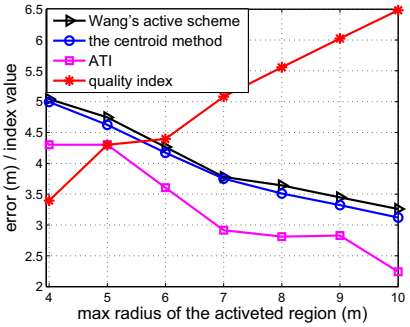


Fig. 6.14: Localization result and quality index in mixed loss (affected angle=270 degrees)

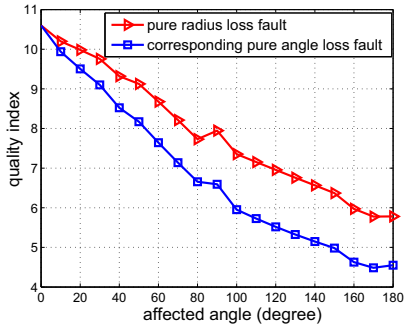


Fig. 6.15: The comparison of quality index in pure radius loss and corresponding pure angle loss

pure angle loss when the affected angle varies from 0 to 180 degrees. The former is always higher than the latter, which is consistent with that the activated region in pure radius loss is larger than the one in corresponding pure angle loss. We also check the case of constant radius loss in all directions, the result shows that the quality index increase with the increased area of activated region (figure omitted). In a general case of pure radius loss/mix loss, quality index also fairly matches the ground truth, as shown in Fig. 6.13 and 6.14. All these results show that our definition of quality index is consistent and distinguishes different situations quite well.

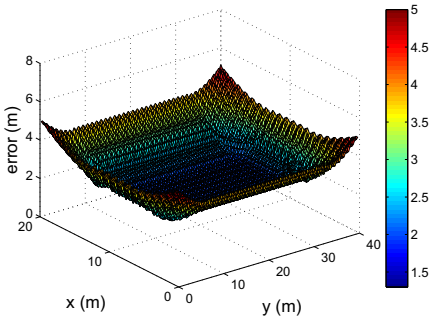


Fig. 6.16: 3D localization result of the centroid method

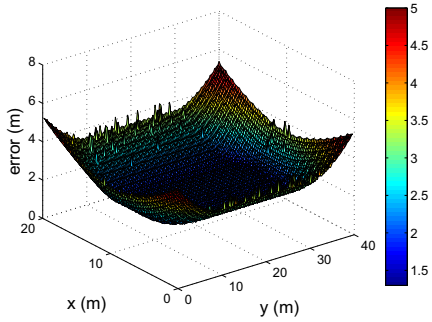


Fig. 6.17: 3D localization result of Wang's active scheme

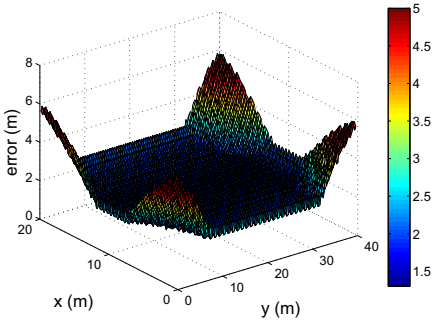


Fig. 6.18: 3D localization result of ATI

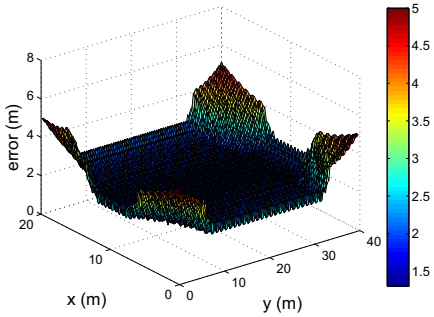


Fig. 6.19: 3D localization result of ATI hybrid method

6.5.4 Performance in 3D RFID Reader Localization

We further check the localization performance of ATI in a 3D area of $40m \times 20m \times 12m$. The object is placed at a height of $5m$ and moved around the area to get the results. The results are shown in Fig. 6.16-6.20. Notice that although the object's actual height is fixed, the calculated height may deviate from it according to the method discussed in Section 6.3.7.

As shown in Fig. 6.16, the centroid method performs well only in a small central area, but suffer large error around the sides and at the corners. Wang's active scheme improves the localization accuracy in the central area, but introduces even larger error at the corners (Fig. 6.17). Neither the centroid method nor Wang's active scheme provide quality index for the localization results, and thus it is difficult to know whether such large error happens

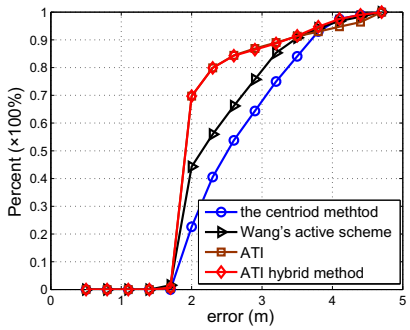


Fig. 6.20: Cumulative distribution function of errors in 3D localization

or not. ATI gains a obvious accuracy increase around the sides where the angle loss is less than 180 degrees (Fig. 6.18). It also causes slight larger error at the corners, compared with that of the centroid method and Wang's active scheme. However, ATI can provide quality index of its localization results, which enables further processing. For instance, we can combine ATI with the centroid method (named ATI hybrid method), in which the ATI method is used to locate the object in normal situation, but changed to the centroid method when the quality index of ATI's result drops to a threshold that denotes no key pairs exist. The localization result of ATI hybrid method is shown in Fig. 6.19. Fig. 6.20 shows the cumulative distribution function of errors in these methods. It can be seen that 69.7% localization results of ATI and ATI hybrid method have the errors less than $2m$, while 22.4% and 44.3% localization results of the centroid method and Wang's active scheme have the same accuracy, respectively. Compared with ATI, ATI hybrid method successfully reduces the localization results whose error is more than $4.4m$ from 3.7% to less than 1%.

6.5.5 Performance in RFID Network Localization

After considering the localization of individual objects, we make one step further to consider the network localization. We compare ATI-MDS with original ATI, Wang's active scheme, and the MDS method that takes the result of Wang's active scheme as the input

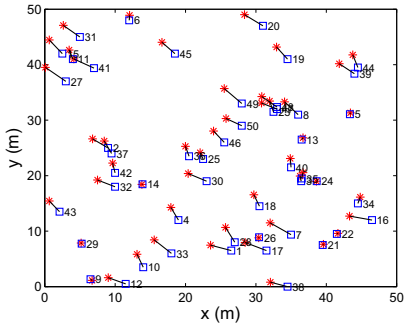


Fig. 6.21: An example of the localization result of ATI in the network localization

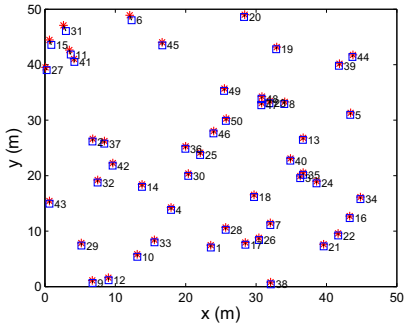


Fig. 6.22: An example of the localization result of ATI-MDS in the network localization

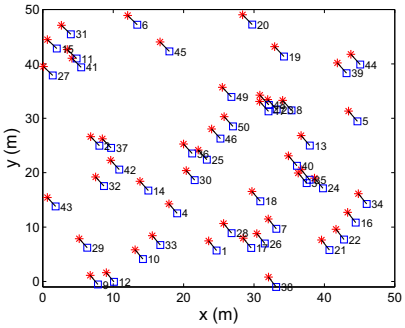


Fig. 6.23: An example of the localization result of Wang-MDS in the network localization

(Wang-MDS for short). 50 objects are randomly distributed in an area of $50m \times 50m$. The results are shown in Fig. 6.21-6.25.

Intuitive examples of the localization results of these methods can be seen in Fig. 6.21-6.23. According to Fig. 6.21 and 6.22, the accuracy of ATI-MDS is much better than that of ATI, due to joint optimization of the locations of all objects. The effectiveness of quality index is clear when comparing ATI-MDS with Wang-MDS. Since Wang-MDS's input lacks quality information, it is more difficult to converge to the real locations (Fig. 6.23).

We also compare the performance of these approaches in different situations quantitatively. The angle losses of the objects are first set uniformly distributed between 90 and 330 degrees, denoting a quite harsh environment. We change the communication range from

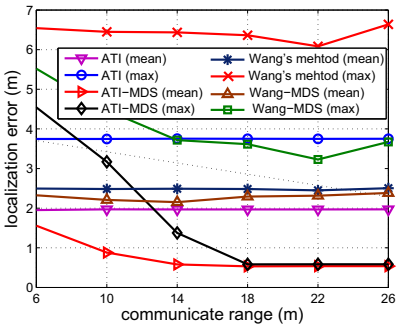


Fig. 6.24: Localization result varying communication range in the network localization

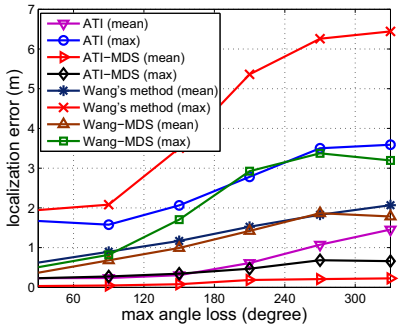


Fig. 6.25: Localization result varying maximum angle loss in the network localization

$6m$ to $26m$ to check the performance, as shown in Fig. 6.24. For each method, we record its average error and maximum error. It can be seen that both ATI-MDS and Wang-MDS significantly reduce the maximum error, compared with ATI and Wang's method, respectively. However, Wang-MDS has limited effect on reducing the average error ($< 14\%$), while ATI-MDS reduces the average error by $20\% - 74\%$. Both the maximum error and average error of ATI-MDS degrease when communication range increases, and reach respective minimum when communication range is about $18m$. After that, the increase of communication range is not helpful to reduce the errors any more. This is because increasing communication range provides more distance information of the objects for localization, and thus makes the final result more accurate, until the information is saturated for MDS method.

Then we fix the communication range of each object as $15m$ and change the maximum angle loss to check the localization performance. The result is shown in Fig. 6.25. We can see that all methods suffer increased error when the maximum angle loss increases. But the error of ATI-MDS increases much slower than other methods. Its average error is about $15\% - 30\%$ that of ATI and $7\% - 13\%$ that of Wang-MDS, and its maximum error is about $14\% - 19\%$ that of ATI and $16\% - 45\%$ that of Wang-MDS in different angle losses.

Fig. 6.26: Experiment configurations of RFID reader localization in an office

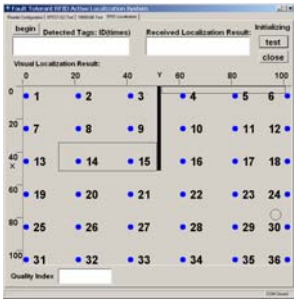
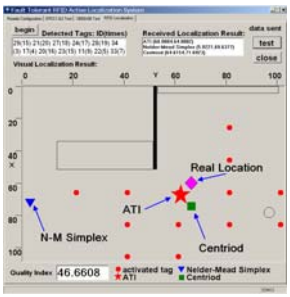
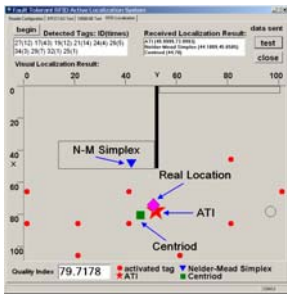


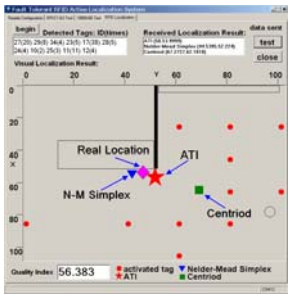
Fig. 6.27: RFID tag deployment shown in the software GUI



(a)



(b)



(c)

Fig. 6.28: Experiment results of different methods.

6.6 Experiments

We developed an RFID reader localization system called RFID-RLS using C# language and conducted experiments in our office, which can be extended to other environments including the warehouse. Due to limited resources, we only conducted 2D localization experiments for single object. Although the experiment itself is simple, we find useful information from it and validate our approach.

The devices include a 915 MHz UHF RFID reader and some EPC Class 1 Generation 2 UHF RFID tags. The reader has an isotropic antenna. The power of the reader is set to 18 dBm which means an identification range of 2.5m in the floor. The on-site configurations of the system are shown in Fig. 6.26. The tags are placed in a grid with the spacing of $40cm$. Detailed tag placement can be seen in the software GUI (Graphical User Interface)

(Fig. 6.27) where walls and desks in the office are also shown as the background. Quality index is shown in the lowest textbox. For ease of comparison, we show the localization results of ATI, Wang's active scheme and the centroid method together. The result of Wang's active scheme is shown as "N-M Simplex" since it uses Nelder-Mead Simplex method to get the localization result.

We set a time duration of 6 seconds in each localization process for obtaining more reliable raw data. Then the data are fed into the localization module to calculate the result. We observed that the detected tags varied from time to time even the reader was static. This may be caused by environment interference and human activities in the office. The walls and desks also make the localization difficult because they block the paths of radio propagation and cause multipath propagation. For example, the upper-left nine tags (tag 1-3, 7-9 and 13-15 in Fig. 6.27) failed to be detected during the whole experiment process due to blocking of the wall.

We move the reader around to check the localization results. Some results are shown in Fig. 6.28(a) to 6.28(c). The real location of the reader is at a)(60,65) b)(50,75) c)(45,55). For Wang's active scheme, we need to roughly estimate the object's position before starting the optimization process, which is not easy in practice. Here we use the result of the centroid method as the initial position. We find Wang's active scheme is quite unstable. As shown in Fig. 6.28(a) and 6.28(b), the result is far from the real location. Especially in Fig. 6.28(b), the result is under the desk, which may make the user quite confused. The centroid method is relatively stable in the experiments. However, the result of it sometimes is not accurate. As shown in Fig. 6.28(c), the localization result of the centroid method deviates from the real location. On the contrary, the result of ATI is quite stable and accurate during all the experiments.

The quality index works well in the experiments. When the detected tags scatter more

evenly in all directions, the quality index is higher (Fig. 6.28(b)). When there are more angle loss or radius loss in the activated region, the quality index is lower (Fig. 6.28(a) and 6.28(c)).

6.7 Summary

In this chapter, we investigated the RFID reader localization when RFID faults frequently happen. We categorized the RFID faults and pointed out that long-lasting regional fault is the most difficult one to be dealt with. We proposed a method named ATI which can tolerate long-lasting regional fault, and defined quality index to measure the accuracy of localization results in both 2D and 3D environment. Our approach is also useful to provide more accurate information to MDS method for network localization which involving multiple target objects to be located. Simulation and experiment results show that our approach outperforms existing approaches in terms of localization accuracy, and our defined quality index matches the ground truth quite well.

Chapter 7

Conclusions and Suggestions for Future Research

In this chapter, we conclude this thesis in Section 7.1 and outline some possible future works in Section 7.2.

7.1 Conclusions

RFID and WSNs are two important data collection techniques in many fields including mobile computing, pervasive computing, and internet of things. Numerous data are obtained through them, which on one hand promote the development of various applications based on them, on the other hand pose new challenges to the processing approaches. The processing approaches also need to meet the special requirements of RFID and WSNs including increasing the reliability and optimizing the utilization of limited resources. Event detection is a data processing technique widely used in many applications. An event encapsulates raw data into a meaningful form that denotes a user-specified activity, and thus relieves the users from tedious underlying data processing. It is desirable to revise and apply event detection technique to the applications of RFID and WSNs.

In this thesis, we investigated complex event detection in RFID and WSNs. We focused on the reliability and energy efficiency in different aspects of event detection: data collection,

event aggregation, and event inference. In each aspect, we identified the problems which lack sufficient studies and proposed corresponding solutions. We conclude these works as follows:

For data collection, we focused on reliable data collection in mobile RFID systems. Specifically, we studied the mobile RFID problem to achieve the maximal tag moving speed while maintaining a high identification rate. We first classified tag arrival models into the combinations of constant/variable arrival and dynamic/isolated arrival. We adopted the dynamic constant arrival model, which is more suitable for an industry environment with dense tag placement and high-speed moving tags. We proposed two principles for mobile RFID anti-collision protocols: workload optimal and the earliest deadline first. The former is used to maintain an optimal number of tags competing for the channel so as to guarantee the identification rate. The latter is to assign a high identification priority to the tags that have tight identification deadlines. Following these principles, we proposed the Schedule-based Anti-collision Algorithm (SAC) to support a high tag moving speed given a high identification rate requirement. Simulation results show that SAC can increase the moving speed of tags by 120% compared with existing approaches, given an identification rate of 99.999%.

For event aggregation, we focused on two energy-efficient event aggregation problems in WSNs. One problem was optimizing event aggregation utilizing complex relations included in a composite event. The other problem was optimizing the aggregation involving multiple composite events with different latency constraints and aggregation functions.

With respect to the first problem, we first built a composite event detection model based on event definition tree and event aggregation tree. Event definition tree supports generic event definitions that may include arbitrarily complex relations. Event aggregation tree is a routing tree among sensor nodes for event aggregation. We proposed two principles

for building event aggregation tree: supporting complex composite event definition and supporting decreasing aggregation function. Following these two principles, both centralized and distributed algorithms were proposed to build energy-efficient event aggregation tree. Simulation results show that our approach saves up to 20% energy than existing approaches.

With respect to the second problem, we first proposed the Delay Bounded Event Aggregation Algorithm (DBEA) to build the optimal energy-efficient event aggregation tree which meets the latency constraint and also considers the event relations for a particular composite event. We further optimized the routing structure for the aggregation of multiple composite events to save energy, by making some composite events share the event aggregation trees of others instead of building their own. Dynamical programming was used to select the base events to build the event aggregation trees. Simulation results show that significant energy (up to 35% in our system) can be saved by using our algorithm.

For event inference, we focused on RFID reader localization, where detected RFID tags were used to infer the location of an RFID reader. It is a challenging task to achieve such an objective in the presence of various faults. Among all the faults, long-lasting regional fault is the most difficult one to be dealt with. We proposed the Activated Tag Included Method (ATI) which can tolerate long-lasting regional fault, and defined quality index to measure the accuracy of localization results in both 2D and 3D environment. Our approach is also useful to provide more accurate information to MDS method for network localization. We have taken extensive simulations and implemented a simple system to validate our approach. Simulation and experiment results shown that our approach outperforms existing approaches in terms of localization accuracy, and our defined quality index matches the ground truth quite well.

In summary, event detection is a powerful technique of data processing in RFID and WSNs if carefully considering the characteristics of RFID and WSNs. We identified several

important problems in different aspects of event detection, and proposed corresponding solutions. The evaluation results show that our approaches can increase the reliability and energy efficiency of event detection in RFID and WSNs.

7.2 Suggestions for Future Research

We close this thesis by providing some suggestions for future research. Specifically, we believe that the following aspects are worth further investigations.

First, in Chapter 3, we consider RFID anti-collision protocols in the dynamic constant arrival model. The problem is still open in the dynamic variable arrival model and isolated variable arrival model. Since in these models, the number of unidentified tags in the interrogation area can change and this change may be quite complex, exerting control over tags is more difficult and needs further investigations. In SAC, we propose a reader-control tag grouping approach to organize tags and then schedule tag identification. This is suitable for passive tags as discussed in this thesis. It is still worth developing tag-control tag grouping technology in RFID anti-collision protocols for active tags. Finally, it is important to investigate the energy consumption of data collection for both active RFID tags and WSNs. Currently, the investigation of energy consumption focuses more on data transmission. These two kinds of energy consumption may be considered together to have a more energy-efficient approach.

Second, in Chapter 4, the complex relations included in a composite event are ultilized to optimize the event aggregation tree for individual events. This work can be extended to the multiple events scenario to consider the relations among composite events. Moreover, the fusion cost in event aggregation is also needed to be considered to make the approaches more practical especially for the data intensive applications such as multimedia applications. In Chapter 5, we optimize the aggregation of multiple composite events considering weak

performance constraint, which means the performance degradations of some events are bounded. This work can be further extended to consider strong performance constraint and multi-level performance constraint. The former denotes the performance degradation of every event is bounded, and the latter denotes the performance degradations of different events have different bounds. The objective also can change to maximize the number of events that satisfy our constraints. By solving these variants of the problem, we can meet diverse requirements of different applications.

Finally, in Chapter 6, we investigate the fault-tolerant RFID reader localization based on passive RFID tags. This topic is studied in the stationary scenario. We can extend it to the object tracking applications where the objects move continuously. The mobility information can be taken into account and combined with our approach. Several details in our approach also need further investigations, such as the shape of the identification region and the grid placement. We need to follow the discussions in our work to consider the irregular identification region and revise the localization approach. Other tag placements rather than the grid placement are also worth more research. For the network localization, we plan to develop a distributed approach which can integrate with our approach and MDS approach. More other event inference problems are also needed to be investigated, to increase the inference accuracy and measure the quality of each particular inference result.

References

[AAHI10] A. Almaaitah, K. Ali, H. S. Hassanein, and M. Ibnkahla. 3D passive tag localization schemes for indoor RFID applications. In *Proc. of ICC*, pages 1–5, 2010.

[ASMM12] K.A. Al-Saud, M. Mahmuddin, and A. Mohamed. Wireless body area sensor networks signal processing and communication framework: Survey on sensing, communication technologies, delivery and feedback. *Journal of Computer Science*, 8(1):121–132, 2012.

[Bac06] B. Bacheldor. China Post deploys EPC RFID system to track mailbags. RFID Journal, July 2006. http://www.rfidjournal.com.

[Ber05] Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control Vol. I (3rd edition)*. Athena Scientific, 2005.

[BHE00] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *Personal Communications*, 7(5):28–34, 2000.

[BKMs+06] Luca Becchetti, Peter Korteweg, Alberto Marchetti-spaccamela, Martin Skutella, Leen Stougie, and Andrea Vitaletti. Latency constrained aggregation in sensor networks. In *Proc. of the European Symposium on Algorithms (ESA)*, pages 88–99, 2006.

[BP08] M. Bouet and G. Pujolle. A range-free 3-D localization method for RFID tags based on virtual landmarks. In *Proc. of PIMRC*, pages 1–5, 2008.

[BP09] M. Bouet and G. Pujolle. L-VIRT: Range-free 3-D localization of RFID tags based on topological constraints. *Computer Communications*, 32(13-14):1485–1494, 2009.

[BSI06] N. Bhandari, A. Sahoo, and S. Iyer. Intelligent query tree (IQT) protocol to improve RFID tag read efficiency. In *Proc. of the 9th International Conference on Information Technology*, pages 46–51, 2006.

[CBLV04] R. Cristescu, B. Beferull-Lozano, and M. Vetterli. On network correlated data gathering. In *Proc. of INFOCOM*, pages 2571–2582, 2004.

[CC01] T. F. Cox and M. A. A. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 2001.

[CCCT06] W. Chen, L. Chen, Z. Chen, and S. Tu. WITS: A wireless sensor network for intelligent transportation system. In *Proc. of IMSCCS*, pages 635–641, 2006.

[CK05] J. R. Cha and J. H. Kim. Novel anti-collision algorithms for fast object identification in RFID system. In *Proc. of ICPADS*, pages 63–67, 2005.

[CM94] S. Chakravarthy and D. Mishra. Snoop: An expressive event specification language for active databases. *Data and Knowledge Engineering*, 14(1):1–26, 1994.

[CPI04] Jose A. Costa, Neal Patwari, and Alfred O. Hero Iii. Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks. *ACM J. Sensor Networking*, 2:39–64, 2004.

[Cro06] Crossbow. MPR-MIB users manual revision b, 2006. http://www.xbow.com.

[CSR04] T. Clouqueur, K.K. Saluja, and P. Ramanathan. Fault tolerance in collaborative sensor networks for target detection. *IEEE Trans. on Computers*, 53(3):320–333, 2004.

[CTTB06] R. H. Clarke, D. Twede, J. R. Tazelaar, and K. K. Boyer. Radio frequency identification (RFID) performance: the effect of tag orientation and package contents. *Packaging Technology and Science*, 19(1), 2006.

[DCX03] M. Ding, X. Cheng, and G. Xue. Aggregation tree construction in sensor networks. In *Proc. of VTC-Fall*, pages 2168–2172, 2003.

[DFDA11] Mario Di Francesco, Sajal K. Das, and Giuseppe Anastasi. Data collection in wireless sensor networks with mobile elements: A survey. *ACM Trans. on Sensor Networks*, 8(1):7:1–7:31, 2011.

[EPC07] EPC Global, Inc. EPC^{TM} Radio Frequency Identity Protocols Class 1 Generation 2 UHF RFID Protocol for Communications at 860MHz-960MHz Version 1.1.0. *Auto-ID Labs White Paper WP-HARDWARE-045*, Oct 2007.

[Fin03] K. Finkenzeller. *RFID handbook: Fundaments and Application in Contactless Smart card and Identification.* John Wiley & Sons, 2003.

[Flo07] C. Floerkemeier. Bayesian transmission strategy for framed ALOHA based RFID protocols. In *Proc. of RFID*, pages 228–235, 2007.

[FLS06] K. Fan, S. Liu, and P. Sinha. Scalable data aggregation for dynamic events in sensor networks. In *Proc. of SenSys*, pages 181–194, 2006.

[FRL07] C. Floerkemeier, C. Roduner, and M. Lampe. RFID application development with the accada middleware platform. *IEEE Systems Journal*, 1(2):82–94, 2007.

[FW06] C. Floerkemeier and M. Wille. Comparison of transmission schemes for framed ALOHA based RFID protocols. In *Proc. of the International Symposium on Applications on Internet Workshops*, pages 92–97, 2006.

[GA02] A. Galton and J. G. Augusto. Two approaches to event definition. In *Proc. of DEXA*, pages 547–556, 2002.

[GD94] S. Gatziu and K. R. Dittrich. Detecting composite events in active database systems using petri nets. In *Proc. of the Fourth International Workshop on Research Issues in Data Engineering: Active Database Systems*, pages 2–9, 1994.

[GE05] Ashish Goel and Deborah Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. *Algorithmica*, 43(1-2):5–15, 2005.

[GJS92] N. H. Gehani, H. V. Jagadish, and O. Shmueli. Event specification in an active object-oriented database. In *Proc. of SIGMOD*, pages 81–90, 1992.

[GJV+05] Lin Gu, Dong Jia, Pascal Vicaire, Ting Yan, Liqian Luo, Ajay Tirumala, Qing Cao, Tian He, John A. Stankovic, Tarek Abdelzaher, and Bruce H. Krogh. Lightweight detection and classification for wireless sensor networks in realistic environments. Proc. of SenSys, pages 205–217, 2005.

[Gow66] J. C. Gower. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, 53:325–338, 1966.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd edition)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[HBF+04] D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with RFID technology. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 1015–1020, 2004.

[HG05] Y. Huang and R. Guerin. Does over-provisioning become more or less efficient as networks grow larger? In *Proc. of ICNP*, pages 225–235, 2005.

[HHB+03] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. Range-free localization schemes for large scale sensor networks. In *Proc. of MobiCom*, pages 81–95, 2003.

[HKI08] HKIA Boosts Baggage Handling Efficiency with RFID Technolog, 2008. http://www.hongkongairport.com/eng/media/press-releases/pr_914.html.

[HKL+08] Nir Halman, Diego Klabjan, Chung-Lun Li, James Orlin, and David Simchi-Levi. Fully polynomial time approximation schemes for stochastic dynamic programs. In *In Proc. of ACM-SIAM SODA*, pages 700–709, 2008.

[HKM+09] Nir Halman, Diego Klabjan, Mohamed Mostagir, Jim Orlin, and David Simchi-Levi. A fully polynomial-time approximation scheme for single-item stochastic inventory control with discrete demand. *Mathematics of Operations Research*, 34(3):674–685, 2009.

[IGE00] C. Intanagonwiwat, R. Goviindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. of MobiCOM*, pages 56–67, 2000.

[Imp09] Impinj Inc. RFID case study: Purdue Pharma, 2009. http://www.impinj.com/Documents/Applications/Case_Studies/Purdue_Pharma_Case_Study.

[KEW02] B. Krishnamachari, D. Estrin, and S. Wicker. Modeling data-centric routing in wireless sensor networks. *USC Computer Engineering Tech. Rep. CENG 02C14*, 2002.

[KI04] B. Krishnamachari and S. Iyengar. Distributed bayesian algorithms for fault-tolerant event region detection in wireless sensor networks. *IEEE Trans. on Computers*, 53(3):241–250, 2004.

[KK04] K. Kaemarungsi and P. Krishnamurthy. Modeling of indoor positioning systems based on location fingerprinting. In *Proc. of INFOCOM*, pages 1012–1022, 2004.

[KN06] M. Kodialam and T. Nandagopal. Fast and reliable estimation schemes in RFID systems. In *Proc. of Mobicom*, pages 322–333, 2006.

[KRJ05] A. V. U. Kumar, A. M. Reddy, and D. Janakiram. Distributed collaboration for event detection in wireless sensor networks. In *Proc. of MAPC*, pages 1–8, 2005.

[LAV+10] Y. Li, C. Ai, C. T. Vu, Y. Pan, and R. Beyah. Delay-bounded and energy-efficient composite event monitoring in heterogeneous wireless sensor networks. *IEEE Trans. on Parallel and Distributed Systems*, 21(9):1373–1385, 2010.

[LCF11] Y. Lai, J. Cao, and X. Fang. TED: Efficient type-based composite event detection for wireless sensor network. In *Proc. of DCOSS*, pages 1–8, 2011.

[LDH06] X. Luo, M. Dong, and Y. Huang. On distributed fault-tolerant detection in wireless sensor networks. *IEEE Trans. on Computers*, 55(1):58–70, 2006.

[Leh86] E. L. Lehmann. *Testing Statistical Hypothesis*. Wiley, 1986.

[LJL05] S. Lee, S. Joo, and C. Lee. An enhanced dynamic framed slotted ALOHA algorithm for RFID tag identification. In *Proc. of Mobiquitous*, pages 166–172, 2005.

[LL06] H. J. Lee and M. C. Lee. Localization of mobile robot based on radio frequency identification devices. In *Proc. of SICE-ICASE*, pages 5934–5939, 2006.

[LLC08] M. Li, Y. Liu, and L. Chen. Nonthreshold-based event detection for 3d environment monitoring in sensor networks. *IEEE Trans. on Knowl. and Data Eng.*, 20(12):1699–1711, December 2008.

[LLD06] H. Luo, Y. Liu, and S. K. Das. Routing correlated data with fusion cost in wireless sensor networks. *IEEE Tran. on Mobile Computing*, 5(11):1620–1632, 2006.

[LLS00] C. Law, K. Lee, and K. Siu. Efficient memoryless protocol for tag identification. In *Proc. of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 763–775, 2000.

[LSKG05] Zhu Li, G.M. Schuster, A.K. Katsaggelos, and B. Gandhi. Rate-distortion optimal video summary generation. *IEEE Trans. on Image Processing*, 14(10):1550–1560, 2005.

[Luc02] D. Luckham. *The Power of Events: An Introduction to Complex Event in Distributed Enterprise Systems.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2002.

[LW06] Weinan Marc Lee and Vincent W. S. Wong. E-Span and LPT for data aggregation in wireless sensor networks. *Computer Communications*, 29(13-14):2506–2520, 2006.

[MCM+06] U. Monaco, F. Cuomo, T. Melodia, F. Ricciato, and M. Borghini. Understanding optimal data gathering in the energy and latency domains of a wireless sensor network. *Computer Networks*, 50(18):3564–3584, 2006.

[MFHH02] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. In *Proc. of OSDI*, pages 131–146, 2002.

[MMP00] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: Two metric network design. In *Proc. of FOCS*, pages 624–630, 2000.

[NLLP03] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: Indoor location sensing using active RFID. In *Proc. of PerCom*, pages 407–415, 2003.

[NSL08] E. W. T. Ngai, F. F. C. Suk, and S. Y. Y. Lo. Development of an RFID-based sushi management system: the case of a conveyor-belt sushi restaurant. *International Journal of Production Economics*, 112(2):630–645, 2008.

[PLK+10] Y. Park, J. W. Lee, D. Kim, J. J. Jeong, and S. Kim. Mathematical formulation of RFID tag floor based localization and performance analysis for tag placement. In *Proc. of ICARCV*, pages 1–5, 2010.

[PW09] L. Pan and H. Wu. Smart trend-traversal: A low delay and energy tag arbitration protocol for large RFID systems. In *Proc. of INFOCOM (mini-conference)*, pages 2571–2575, 2009.

[QNL08] C. Qian, H. Ngan, and Y. Liu. Cardinality estimation for large-scale RFID systems. In *Proc. of PerCom*, pages 30–39, 2008.

[Rob04] M. Roberti. Wal-Mart begins RFID rollout. RFID Journal, April 2004. http://www.rfidjournal.com.

[RTWL09] Z. Ren, C. C. Tan, D. Wang, and Q. Li. Experimental study on mobile RFID performance. In *Proc. of WASA*, pages 12–21, 2009.

[RZHJ07] A. Rahmati, Lin Zhong, M. Hiltunen, and R. Jana. Reliability techniques for RFID-based object tracking applications. In *Proc. of DSN*, pages 113 –118, 2007.

[SDOS07] J. Singh, C. Deupser, E. Olsen, and S. P. Singh. An examination of the variables affecting RFID tag readability in a conveyor belt environment. *Journal of Applied Packaging Research*, 2(2):61–73, 2007.

[SDR08] V. Sarangan, M. R. Devarapalli, and S. Radhakrishnan. A framework for fast RFID tag reading in static and mobile environments. *Computer Networks*, 22(5):1058–1073, 2008.

[SK08] T. Sanpechuda and L. Kovavisaruch. A review of RFID localization: applications and techniques. In *Proc. of ECTI-CON*, pages 769–772, 2008.

[SMSC08] S. Singh, M. McCartney, J. Singh, and R. Clarke. RFID research and testing for packages of apparel, consumer goods and fresh produce in the retail distribution environment. *Packaging Technology and Science*, 21(2):91–102, 2008.

[SN11] S. S. Saad and Z. S. Nakad. A standalone RFID indoor positioning system using passive tags. *IEEE Tran. on Industrial Electronics*, 58(5):1961–1970, 2011.

[Spi01] M.A. Spirito. On the accuracy of cellular mobile station location estimation. *IEEE Tran. on Vehicular Technology*, 50(3):674 –685, may 2001.

[SS97] A. Shaikh and Kang Shin. Destination-driven routing for low-cost multicast. *IEEE Journal on Selected Areas in Communications*, 15(3):373–381, 1997.

[SW11] Wenbo Shi and Vincent W. S. Wong. MDS-based localization algorithm for RFID systems. In *Proc. of ICC*, pages 1–6, 2011.

[SWJ+05] X. Shen, Z. Wang, P. Jiang, R. Lin, and Y. Sun. Connectivity and RSSI based localization scheme for wireless sensor networks. In *Proc. of ICIC*, pages 578–587, 2005.

[SY07] Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. *Math. Program.*, 109(2):367–384, 2007.

[TM80] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Marh. Japonic*, 24(6):573–577, 1980.

[TTT99] K. C. Toh, M.J. Todd, and R. H. Ttnc. SDPT3 – a matlab software package for semidefinite programming. *Optimization methods and software*, 11:545–581, 1999.

[Vog02a] H. Vogt. Efficient object identification with passive RFID tags. In *Proc. of Pervasive*, pages 98–113, 2002.

[Vog02b] H. Vogt. Multiple object identification with passive RFID tags. In *Proc. of IEEE International Conference on Systems, Man and Cybernetics*, pages 6–12, 2002.

[Wax88] B. M. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, 1988.

[WB09] Markus Wälchli and Torsten Braun. Efficient signal processing and anomaly detection in wireless sensor networks. In *Proc. of the EvoWorkshops*, pages 81–86, 2009.

[WBBB07] Evan Welbourne, Magdalena Balazinska, Gaetano Borriello, and Waylon Brunette. Challenges for pervasive rfid-based infrastructures. In *Proc. of PerCom Workshops*, pages 388–394, 2007.

[WGD10] Jing Wang, R. K. Ghosh, and Sajal K. DAS. A survey on sensor localization. *J. Control Theory Appl.*, 8(1), 2010.

[WLZ+07] Z. Wang, D. Liu, X. Zhou, X. Tan, J. Wang, and H. Min. Anti-collision scheme analysis of RFID system. *Auto-ID Labs White Paper WP-HARDWARE-045*, 2007.

[WWT07] C. Wang, H. Wu, and N.-F. Tzeng. RFID-based 3-D positioning schemes. In *Proc. of INFOCOM*, pages 1235–1243, 2007.

[WZYB08] Z. Wang, S. Zheng, Y. Ye, and S. Boyd. Further relaxations of the semidefinite programming approach to sensor network localization. *SIAM Journal on Optimization*, 19(2):655–673, 2008.

[XGWX10] X. Xu, L. Gu, J. Wang, and G. Xing. Negotiate power and performance in the reality of RFID systems. In *Proc. of PerCom*, pages 88–97, 2010.

[XLCL06] W. Xue, Q. Luo, L. Chen, and Y. Liu. Contour map matching for event detection in sensor networks. In *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*, SIGMOD'06, pages 145–156, 2006.

[XST+10] L. Xie, B. Sheng, C. C. Tan, H. Han, Q. Li, and D. Chen. Efficient tag identification in mobile RFID systems. In *Proc. of INFOCOM*, pages 1–9, 2010.

[YKP04] Y. Yu, B. Krishnamachari, and V. K. Prasanna. Energy-latency tradeoffs for data gathering in wireless sensor networks. In *Proc. of INFOCOM*, pages 244–255, 2004.

[YL10] Zheng Yang and Yunhao Liu. Quality of trilateration: Confidence-based iterative localization. *IEEE Tran. on Parallel and Distributed Systems*, 21(5):631–640, 2010.

[YMG08] J. Yick, B. Mukherjee, and D. Ghosali. Wireless sensor networks survey. *Computer Networks*, 52(12):2292–2330, 2008.

[YTH+04] K. Yamano, K. Tanaka, M. Hirayama, E. Kondo, Y. Kimuro, and M. Matsumoto. Self-localization of mobile robots with RFID system by using support vector machine. In *Proc. of IROS*, pages 3756–3761, 2004.

[ZGC+09] S. Zhang, S. Gong, Z. Cui, Q. Liu, and J. Fan. An aggregation tree approach for event detection in wireless sensor networks. *Journal of Software*, 4(8):899–906, 2009.

[ZVPS08] Y. Zhu, R. Vedantham, S.-J. Park, and R. Sivakumar. A scalable correlation aware aggregation strategy for wireless sensor networks. *Information Fusion*, 3(9):354–369, 2008.

[ZYW06] X. Zhang, S. Yue, and W. Wang. The review of RFID applications in global postal and courier services. *The Journal of China Universities of Posts and Telecommunications*, 13(4):106–110, 2006.