AN INTEGRATED FRAMEWORK FOR QOS-AWARE DATA REPORTING IN WIRELESS SENSOR NETWORKS

by

HYUN JUNG CHOE

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2009

To my parent, my sister Hyun Young and brother Dong Hyun

who set the example and who made me who I am.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervising professor, Dr. Sajal K. Das, for constantly motivating and encouraging me, and also for his guidance and patience during the course of my doctoral studies. Special thanks are due to Prof. Kalyan Basu for his invaluable advice for discussing research problems as well as school life. Moreover, I deeply appreciate my academic advisors Dr. Mohan Kumar, Dr. Yonghe Liu, and Dr. Bob Weems for their insightful comments that improve my research and for taking valuable time to be in my dissertation committee.

I would like to extend my appreciation to all my colleagues in the Center for Research in Wireless Mobility and Networking (CReWMaN), Wook, Afrand, Samik, Preetam, Sumantra, Sourav, Pradip, Nirmalya, Habib, Wei, Indradip, Jun-won, Gautham, Na, Giacomo, Mayank, Avinash, Sajib, Mario and a visiting scholar Vanessa for not only their support and all the valuable discussions but also invaluable memories during my Ph.D. study.

I would also like to thank the Computer Science and Engineering (CSE) department of the University of Texas at Arlington for providing me Teaching Assistantship, Hermann Fellowship, and STEM Doctoral Fellowship. I take this opportunity to thank all the professors who taught me during the year I spent in school. Especially, I am grateful to Dr. Moon Hwa Park in South Korea for encouraging and inspiring me to pursue Ph.D. study in the Unites States.

Finally, I would like to express my deep gratitude to my parents, sister Hyun Young, and brother Dong Hyun who have always encouraged and supported me and

have been patient and sacrificed. I am extremely fortunate to be so blessed.

December 3, 2009

ABSTRACT

AN INTEGRATED FRAMEWORK FOR QOS-AWARE DATA REPORTING IN WIRELESS SENSOR NETWORKS

HYUN JUNG CHOE, Ph.D.

The University of Texas at Arlington, 2009

Supervising Professor: Sajal K. Das

Wireless sensor networks are being deployed in a wide variety of applications such as environment monitoring, smart buildings, security, machine surveillance system, and so on. The deployment of sensor networks for a specific sensing application enhances the ability to control and examine the physical environments while collecting meaningful information from the monitoring area. In densely deployed networks, the sensor nodes located in an adjacent area detect the targeted phenomena in its sensing range and report the gathered (raw or processed) data to designated sinks via single-hop or multi-hop communication paths. Although the correlation of data from proximity sensors cause overheads in terms of energy consumption for data delivery and processing, yet they improve data accuracy. Therefore, the definition of quality of service (QoS) and the metrics to evaluate the performance of a wireless sensor network are different from traditional networks in that the QoS attributes highly depend on the specific sensing tasks and applications. While energy efficiency is an important consideration for designing algorithms and protocols for wireless sensor networks, other QoS parameters such as the coverage rate, the end-to-end delay,

fairness, throughput, and error rates for delivery or sensing may be equally important depending on the application objectives. Thus, an important issue in a sensor network is to design task-specific QoS-aware data reporting algorithms and protocols that optimize resource consumption and extend the network lifetime. In this dissertation, we propose an integrated framework for QoS-aware data reporting in wireless sensor networks. More specifically, the proposed framework is designed for single-hop cluster-based wireless sensor networks and includes two strategies: an intra-cluster data reporting control strategy (IntraDRC) and an inter-cluster data reporting control strategy (InterDRC).

The IntraDRC strategy is based on the selection of data reporting nodes that applies the block design concept from combinatorial theory and a novel two-phase node scheduling (TNS) scheme that defines class-based data reporting rounds and node assignment for each time slot. The objective of IntraDRC is to provide optimized data reporting control in a distributed manner. In this strategy, a certain number of data reporting nodes are selected in each cluster in order to satisfy the throughput fidelity specified by the applications while reducing redundant data reporting by selecting a subset of cluster members. This intra-cluster reporting control eventually helps control the overall amount of traffic in the network. The TNS scheme schedules data reporting while considering the priority of data, yet guaranteeing that sensor nodes compete with each other in the same class only. The InterDRC strategy, on the other hand, is based on QoS-aware data reporting tree management scheme that balances the trade-off between the end-to-end delay and energy efficiency. The idea of this strategy is to manage variants of the data reporting tree based on two information, such as the hop counts to a data sink and the traffic amount generated from local area. For this purpose, each cluster head analyzes the traffic scenario of its cluster for load balancing and congestion control, thus improving the overall network performance. In InterDRC, the proposed spanning tree construction algorithm first builds the fewest hop-based reporting tree, used for delay constrained data delivery. This tree is updated with traffic load information in order to construct a traffic-adaptive reporting tree, used for energy efficient data delivery.

By separating the controls of data reporting within a cluster and that from one cluster to another, the proposed integrated framework can define different levels of various QoS parameters in each intra-cluster data reporting as well as inter-cluster reporting. To the best of our knowledge, we are the first to propose node arrangement using block designs in order to design task-specific data report scheduling in wireless sensor networks. This node arrangement strategy facilities an efficient local data collection in a cluster. Simulation results demonstrate that the proposed framework results in a significant conservation of energy by reducing the competition between data reporting nodes and establishing traffic-adaptive data reporting paths. The results also show that the throughput performance of our integrated framework is especially good due to stable data reporting independent of the network density.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

The rapid development of smart devices and advances in wireless communications technologies extend the areas of applications in the fields of science and engineering. Especially intelligent small devices which collaborate with each other with embedded systems, designed for a specific purpose, introduce a smart environment by monitoring and controlling a target element with information that the devices learn. A sensor device is the one that permits such intelligent and pervasive computing environments in our lives. Sensor networks form different types of network modes depending on the communication method, the network density, and so on.

A wireless sensor network also can include other types of wired and wireless devices such as cell phones and personal digital assistants (PDAs). One current research interests in wireless sensor networks is to connect sensors into interactive devices and networks fostering a wide class of interactive pervasive and ubiquitous computing applications. By integrating these devices to provide ubiquitous access to several types of networks, many new applications emerge. The trends to integrate wireless sensors into interactive devices such as cellular phones will include many potential applications.

1.1 Wireless Sensor Networks

Wireless sensor networks are being deployed in a wide variety of applications such as environmental monitoring, smart building, facility management, target tracking, security, and so on. An important role of a wireless sensor network is monitoring the target area and reporting the data acquired from that area to a sink. A wireless sensor network consists of several dozens, hundreds, or even thousands of wireless sensor nodes that have sensing, processing, and communication capabilities. A sensor node detects the targeted phenomena in its sensing range such as temperature, humidity, light, vibration, and sound depending on different types of the sensing applications or tasks [15, 26, 18, 53, 14] and transmits the sensing results, which can be raw sensing values or processed data, to its data sink in a single or multi-hop manner [34, 5, 10]. A sensor network may run for more than one applications or tasks.

A wireless sensor node is defined as a device with inexpensive prices, low-power energy source, and self-configuring network technologies that allow sensor nodes to be easily deployed in a wide monitoring area in an ad hoc manner. The deployment interfaces with the physical world and enhances the ability to examine and optimize the environments. A sensor device is an embedded system that means the processing capability is integrated with the control and the operations are not based on human interaction. A processor has omnidirectional sensors for measuring the environmental phenomena depending on the interests of the sensing applications such as temperature, light, vibration, sound, barometers, smoke detectors, and so on. Some types of sensor nodes have the ability to detect the location that the node is deployed using a global positioning system (GPS), but in usual sensors can distinguish between obstacles and nodes but cannot determine individual node location. Recent advances especially in hardware make a sensor node feasible to deploy various area but still many challenges remain. One example of wireless sensor devices is the Berkeley MICAx motes that are commonly used in wireless sensor network researches. The MICAx motes is constructed using off-the-shelf components and includes an I/O connector to provide a stack-able platform for effective integration with sensor sand alternative communication boards for experimentation. The MICAx is designed primarily to handle limited amounts of data from simple sensors such as temperature and light and is not suitable for general types of applications that require high bandwidth data such as multimedia data. The Intel Imote increases processing and memory capacity to provide multimedia data processing and performs robust in-network communications.

In usual sensor nodes are deployed for specific applications and the operations of the nodes highly depend on the application-specific requirements. The area that sensors are deployed also affects the capabilities and types of sensors; for example, a certain sets of sensors may be able to be wired to a nearby closed-loop monitoring systems. In densely deployed wireless sensor networks, several sensor nodes collaborate with each other in order to make the decision about a particular event occurred in the monitoring area.

Compared to traditional networks such as a wireless local area networks (WLAN), a wireless sensor networks has its own characteristics. These characteristics also become main challenges in designing and developing algorithms and protocols for wireless sensor networks.

- A wireless sensor network is expected to monitor an event and/or collect meaningful information rather than just collect data to have high performance of throughput. The desired information depends on the objectives of the sensing applications. Therefore, the applications/tasks-specific demands decide the design and operation issues of sensor networks. For example, in some cases sensor nodes are required to be identified with their own node id, as known as an address-centric system, but in some other cases, specific geographic location information is important rather than the identification of a node. In these cases, a location detection device such as GPS can be attached onto a sensor device. Yet in the other cases, he sensing values in a given location area is more important then where or how many sensors the data came from. This is called a data-centric system such that data can be used to set triggers a particular action to a network or query information from a network.

- A sensor node has scarce resources such as bandwidth, memory, processing capability, and energy. Especially limited supply of energy is one main consideration to develop algorithms and protocols for the wireless sensor networks. In practice, most of wireless sensor device products operate using batteries. Replacing or recharging the batteries after deploying the devices is usually not practicable, but a wireless sensor network is usually expected to be operated for a given mission time or as long as possible. Therefore, an energy-efficient operations of sensor nodes are essential.
- Scalability is another issue since a wireless sensor network usually consists of a large number of sensor nodes. The embedded architectures and algorithms/protocols have to provide the way how to configure and support these nodes. On the other hand, the number of nodes per unit area, defined as the density of a network, can vary. Also, sensor nodes can easily fail the operations because of a energy problem or environmental causes. Therefore, the algorithms and protocols have to adopt the variable scalability and density problems.
- A wireless sensor network is required to self-configured in most of its applications and protocols. For example, sensor nodes are able to determine their geographical positions using the information form the other nodes.
- In many cases several sensors collaborate with each other to satisfy the objectives and goals of the sensing applications. In order to provide enough information to detect a certain event, the joint data of several sensors are processed in the network in various forms.

1.2 Motivation for This Dissertation Work

The main objective of a wireless sensor network is monitoring physical phenomena specified by the sensing application and delivering the sensing results to the sink via wireless communications so that the end user can extract information of the monitoring area based on the collected data.

Traditional quality of service (QoS) parameters are defined related to the quality of multimedia data in order to provide high throughput, low end-to-end delay, low jitter, and low packet loss rate. In wireless sensor networks, the QoS parameters highly depends on the types of applications and sensing tasks. For example, some applications do not require to collect data from the all nodes deployed in the networks. Energy efficient operations may be also one explicit QoS parameter in a wireless sensor network.

Some general possible parameters specified in wireless sensor networks follow. Firstly, quality of information (QoI) is important rather than high throughput and low data loss rate. In other words, reliable event detection and the adapted accuracy level of approximation quality may be more important concern depending on the sensing applications.

Compared to traditional QoS demands that require maximized quality, QoS in wireless sensor network is expected to provide minimum required level of quality so that a network uses small amount of limited resources and extends the lifetime. Some applications may have their own throughput fidelity such that a certain amount of data is enough to achieve the objectives of tasks. In this case, a network is required to efficiently manage traffic generated from sensor nodes so that unnecessary data traffic should be controlled not to be delivered wasting network resources.

In order to allow a system to control the networks, cross-layer design to handle parameters in different layers is necessary. Especially in wireless sensor networks, tunable parameters by applications or users easily affect the others in different layer. As an example, changing the sampling rate will affect the performance of MAC protocols and the decision of routing paths. In [50], a combination of link schedule and power control algorithm has been proposed while minimizing total power consumption by controlling the data rate for each link. It is also able to give a solution for determining routing paths. Another aspect is that the low layer protocols may be able to effectively handle the requirements by the applications. As the design complexity is not ignorable, balancing the trade-off between the design complexity and the saved resources from the design is important.

By designing additional architecture such as an integrated cross-layer design or middleware [59], the user-defined QoS goals can be achieved by managing data flows in a network as well as the sensor nodes and network resources. For example, MiLAN is linking applications and networks to to provide middleware using cross-layer management. The MiLAN applications provide the required QoS description so that middleware adjusts the parameters in different layers to optimize the usage of network resources while monitoring the current network conditions in order to maximize the network lifetime. The idea of MiLAN is to adapt to the network-specific features regardless which protocols are being used for communications. The goal is to efficiently manage the network resources and satisfy the application-specific QoS requirements.

The QoS parameters for a particular sensing task can be satisfied using data from one or more sensors. The applications can specify this kind of information such as how many sensors or which sets of sensors can satisfy the QoS requirements, and the sensor networks and systems can learn the applications/user-specified information during the QoS provisioning time.

1.3 Contributions of This Dissertation

In this dissertation, we propose an integrated framework for QoS-aware data reporting in wireless sensor networks. More specifically, the proposed framework is designed for single-hop cluster-based wireless sensor networks and includes two strategies: an intra-cluster data reporting control strategy (IntraDRC) and an inter-cluster data reporting control strategy (InterDRC).

The IntraDRC strategy is based on the selection of data reporting nodes that applies the block design concept from combinatorial theory and a novel two-phase node scheduling (TNS) scheme that defines class-based data reporting rounds and node assignment for each time slot. The objective of IntraDRC is to provide optimized data reporting control in a distributed manner. In this strategy, a certain number of data reporting nodes are selected in each cluster in order to satisfy the throughput fidelity specified by the applications while reducing redundant data reporting by selecting a subset of cluster members. This intra-cluster reporting control eventually helps control the overall amount of traffic in the network. The TNS scheme schedules data reporting while considering the priority of data, yet guaranteeing that sensor nodes compete with each other in the same class only. The InterDRC strategy, on the other hand, is based on QoS-aware data reporting tree management scheme that balances the trade-off between the end-to-end delay and energy efficiency. The idea of this strategy is to manage variants of the data reporting tree based on two information, such as the hop counts to a data sink and the traffic amount generated from local area. For this purpose, each cluster head analyzes the traffic scenario of its cluster for load balancing and congestion control, thus improving the overall network performance. In InterDRC, the proposed spanning tree construction algorithm first builds the fewest hop-based reporting tree, used for delay constrained data delivery. This tree is updated with traffic load information in order to construct a traffic-adaptive reporting tree, used for energy efficient data delivery.

By separating the controls of data reporting within a cluster and that from one cluster to another, the proposed integrated framework can define different levels of various QoS parameters in each intra-cluster data reporting as well as inter-cluster reporting. To the best of our knowledge, we are the first to propose node arrangement using block designs in order to design task-specific data report scheduling in wireless sensor networks. This node arrangement strategy facilities an efficient local data collection in a cluster.

In particular, our contributions include:

- We separate data reporting control in a network into intra-cluster and inter-cluster data reporting schemes based on our network model that is a single-hop cluster-based topology. By dividing control mechanisms into intra-cluster and inter-cluster operations, the available resources of a network can be easily utilized and simplified.
- In intra-cluster data reporting control, we consider the application-specific throughput fidelity in order to choose a certain subset of data reporting nodes. As the reliability of reporting paths is not 100%, we also consider the delivery error rate to satisfy the requirement level at the end system.
- In inter-cluster data reporting control, we adopt two QoS parameters: the end-to-end delay and energy efficiency. For the end-to-end delay constraint, the proposed scheme constructs a spanning tree based on hop counts of each cluster head to a data sink; on the other head, load-balanced spanning tree is considered to distribute traffic load. In order to compromise the trade-off between two parameters, we use the weighting value to give different important level based on the requirement.

- We propose the local addressing scheme that includes application and priority information within a short local address. The objective of this scheme is to represent the required QoS information of a packet inside the packet header while reducing the size of the header to reduce the energy consumption caused by frequent communications in a cluster.

We analyze that the TNS scheme supports stable data reporting by selecting reporting nodes at a given time and the QRT scheme provides energy or delay adaptive reporting environments by offering traffic-adaptive data reporting tree from simulation results.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. Chapter 2 deals with detailed background and related works of cross-layer design and quality of service issues in wireless sensor networks. We also present the network model and the overview of the proposed framework. Chapter 3 describes the network model where we can apply our integrated QoS-aware data reporting control framework followed by Chapter 4 that presents the overview of the framework and the problem descriptions. Chapter 5 describes a class-based node allocation scheme, which includes QoS-aware data reporting node selection and two-phase node scheduling (TNS) scheme. In this chapter, we describe the problem for data reporting inside a cluster and discuss the solutions. Chapter 6 describes the QoS-aware data reporting tree management, called QRT scheme, which offers traffic-adaptive data reporting paths. It begins with related works about congestion control problem in wireless sensor networks and explains QoS parameters to be considered for the reporting tree construction. Then, we present our data reporting tree management strategy. Chapter 7 presents the QoS-aware local addressing scheme that represent the required QoS information within a packet header to facilitate the treatment of packets while reducing the energy consumption caused by frequent local communications to support the QoS demands. Finally, Chapter 8 concludes this dissertation with future research directions.

CHAPTER 2

BACKGROUND AND RELATED WORKS

Wireless sensor networks (WSNs) are being deployed in a wide variety of applications such as environment monitoring, smart buildings, security, and so on. An important role of a WSN is monitoring the target area and reporting the data acquired from that area to the end system. In densely deployed networks, several sensor nodes located in an adjacent area detect the targeted phenomena in its sensing range such as temperature, humidity, light, vibration, or sound depending on one or more types of applications [15, 26, 18, 53, 14]. Then the sensors report the results, which can be raw sensing values or processed data, to data sinks in a single or multi-hop manner [34, 5, 10]. Although such correlation of data from proximity sensors cause overheads in terms of energy consumed for delivery and processing, yet they improve the data accuracy.

Designing protocols and algorithms for WSNs is more challenging due to limited resources, lack of centralized control, unreliable wireless channel conditions, and various application-specific demands. In addition, some parameters may affect others at different layers; for example, changing the duty cycle parameter determined by a scheduling function in the MAC layer affects the routing decision at the network layer.

In order to provide optimized service for task-specific requirements while efficiently using scarce resources, an integrated cross-layer design is necessary to alleviate the effects of some parameters on others at different layers.

The rest of this chapter is organized as follows: In Section 2.1, we articulate the important research challenges in wireless sensor networks. Section 2.2 presents existing communication protocols, which mainly focuses on medium access control protocols. Section 2.3 demonstrates data management in wireless sensor networks in details, and Section 2.4 describes QoS and integrated cross-layer design issues proposed in wireless sensor networks.

2.1 Challenges

- **Data reporting** In densely deployed wireless sensor networks, data redundancy is an important issue. Redundant data reporting results in unnecessary power consumption and hence significantly reduces the network life time; on the other hand, data redundancy provides data accuracy at the end system. Therefore, the optimize data reporting while considering the trade-off between data redundancy and data accuracy is important. In order to deal with this trade-off, our framework uses the concept of *data aggregation* and the selection of data reporting nodes.
- **Medium access control (MAC)** When several sensor nodes attempt to transmit data simultaneously, collisions and transmission failures cause unnecessary energy consumption. Therefore, an efficient medium access control protocol is essential especially in densely deployed wireless sensor networks. MAC protocols can be categorized in three classes: schedule-based access mode, contention-based mode, and hybrid access mode, which adopts both schedule-based and contention-based modes. While the schedule-based protocols can reduce energy consumption using sleep/wake-up modes by scheduling node sleep time, this mode my waste of time slot assigned to idle nodes. On the other hand, the contention-based protocols is simple to implement and operate with no synchro-

nization and the decentralized nature, this mode may cause inefficient energy consumption generated from transmission failures by interference and retransmission. Therefore, the optimized medium access control design depending on the types of tasks is important to reduce the unnecessary energy consumption.

- **Quality of service (QoS)** The definition of quality of service (QoS) and the metrics to evaluate the performance of a WSN are also different from traditional networks in that the QoS attributes highly depend on the specific sensing tasks and applications. While energy efficiency is an important consideration for designing algorithms and protocols for WSNs [16], other QoS parameters such as the coverage rate, the end-to-end delay, fairness, throughput, and error rates for delivery or sensing may be equally important depending on the application objectives. Fidelity and scalability are important design consideration factors. Our integrated framework is based on the knowledge that local decision-making is sufficient for scalability. The sensor network problem is to extract information concerning some physical phenomenon to within some fidelity, given nodes with some constraint on resources. As the density of nodes increases, the possibilities for spatial correlation of sensing results increases, and the information that must be delivered will saturate according to the fidelity threshold, if only nodes have a mechanism for determining which ones will be involved in some form of local fusion and which ones will report nothing. Our integrated framework attempts to provide a QoS adaptive data reporting control scheme that considers throughput, delay, and energy as QoS parameters.

2.2 Communication Protocols

MAC protocols in WSNs can be classified into schedule-based and contention-based protocols. Schedule-based mechanism provides collision-free medium access but includes possible drawbacks, such like time synchronization overhead and increased latency caused by idle slots. These protocols require time synchronization either globally or locally.

Medium access control(MAC) protocols, which specify how sensor nodes share the communication channel, have been considered as an important area to decide the performance of sensor nodes and further network lifetime of WSNs while devoting to development of energy efficient mechanism including sleep/wake-up mode and adaptive listening to reduce idle listening time, control channel approaches, etc. In our proposed framework, we focus on a channel access algorithm which can be used with other components of existing MAC protocols. Channel access schemes for WSNs can be classified into contention-free and contention-based protocols. In contention-free protocols, since there is no collision, which occurs when more than one sensor node in overlapped transmission range try to communicate via shared media, sensor nodes can reduce energy consumption caused by transmission failures and retransmissions and further provide increased accuracy of decision making from collected data. In addition, it is easy to provide fairness for each sensor to send its data with contention-free protocols. In contention-based protocols, the probability to waste idle slot dedicated to a particular node will be less than TDMA based mechanism and reduce the transmission delay and provide more flexible and efficient resource share in irregular event occurrence and frequent topology changes. However, energy consumption is greater than contention-free protocols since a sensor wastes energy for failed transmission caused by collisions and sometimes it is required to spend more energy for retransmission. Our design of the MAG scheme is based on hybrid mechanism managing contention-free and contention-based portion depending on different types of application. The main part of this work is to find optimized grading pattern to handle the tradeoff between energy efficiency and delay to support application-specific sensor operation to collect sensing data.

MAC protocols in WSNs can be classified into TDMA based and random access protocols. TDMA mechanism provides collision-free medium access but includes possible drawbacks, such like time synchronization overhead and increased latency caused by idle slots. TDMA protocols require time synchronization either globally or locally. [55] partitions sensor nodes into clusters and the cluster heads maintain a TDMA schedule to exchange data between cluster members and heads. In [55], there is no peer-to-peer communication. In order to communicate between clusters, CDMA code, which has been known as an expensive mechanism in WSNs, is used. [32] and [33] propose the Self-Organizing Medium Access Control for Sensor networks (SMACS) to combine neighborhood discovery and TDMA schedule assignment. SMACS assumes that sensor nodes are able to use many CDMA codes in many channels. A node uses fixed-length superframes which do not need to use the same phase as the neighbor's superframes. However, all nodes use the same superframe length and it requires time synchronization. In [49], sensor nodes access a single channel in a collision-free manner. The proposed protocol, Traffic-Adaptive Medium Access (TRAMA), assumes that all nodes are time synchronized and the schedules are managed in a distributed manner on an on-demand basis. TRAMA protocol uses two time periods, random access and schedule access periods. During a random access period, each node broadcasts its schedule and neighbor information and learns its two-hop neighbor information. Also it sends a list of receivers for the packets in a queue periodically.

The main purpose of low duty cycle is to avoid waste of energy during idle state. Although the Sparse Topology and Energy Management (STEM) protocol does not cover all MAC functions, it provides a solution for the idle listening problem in [9]. It defines Wakeup and Data channels and uses the wakeup channel as a control channel to identify if there is transmission activity. In STEM-B, a node to transmit a packet sends beacons on the wakeup channel periodically without prior carrier sensing. As soon as a receiver sends an acknowledgment frame for the beacon, the transmitter and receiver nodes perform the transmission on the data channel. In STEM-T, a node uses a simple busy tone to provide cheaper and less energy-consuming transmission. S-MAC(Sensor-MAC) is proposed in [67, 68] where sensors within a certain virtual cluster manage local synchronization and agree the same schedule performing fixed sleep and listen time. In S-MAC, the listen schedule is required to coordinated in the virtual cluster so a node exchanges its schedule with neighbors in a SYNCH field. The SYNCH field is subdivided into time slots and neighbors contend with backoff scheme. After setting up the schedule, a node uses RTS/CTS handshake mechanism to reduce collisions of data packets. In usual nodes on the border of a cluster are required to manage more than one schedule and spend more energy. The listen period of S-MAC can be used to both transmit and receive packets. Due to the delivery latency in [67], [68] proposes the adaptive-listening scheme to reduce per-hop latency and [52] proposes Timeout-MAC (T-MAC) for a node to go to a sleep mode when there is no activity during short time listening period. Dynamic Sensor MAC (DSMAC) proposed in [39] suggests to use dynamic duty cycle mechanism to decrease the latency by sharing one-hop latency values, which means the time difference between enqueued time and transmitted time. In [40], DMAC is proposed to construct unidirectional convergecast tree from sensor nodes to the sink for data gathering in WSNs. The purpose of DMAC is to reduce latency and also achieve energy-efficiency. Low latency is achieved by assigning subsequent slots to the nodes that are successive level in the tree. Even though [40] shows good results for decreased latency compared to other methods, it does not provide collision avoidance mechanism for the nodes using the same time schedule in the same level of the tree and attempting transmission to the same node in the successive level.

While other works presented above do not focus on collisions between nodes, [31] and [63] propose slot assignment mechanisms. They work for event-driven sensor network environments achieving only the subset of total packets. If no activity is sensed, sensor nodes increase their transmission probability exponentially for the next slot assuming that only small amount of traffic exists in the network. [63] further presents how to decide a non-uniform probability distribution.

The following works are related to our proposed work in terms of background concept. The predictive p-persistent CSMA protocol is designed to cope with overload situations. The probability p is derived based on traffic of acknowledgments and retransmissions in order to adjust to the expected traffic dynamically and the estimated backlog concept is applied for collision avoidance. However, the derivative traffic, such as acknowledgments and retransmissions, is not applicable in some sensor network applications since reliable delivery requires extra energy consumption.

In [49], Traffic-Adaptive Medium Access (TRAMA) is proposed for hybrid channel access mechanism based on the amount of traffic. The efficient management of duty cycle and sleep/wake-up scheduling discussed in [9, 67, 68] are also important issue to balance the trade-off between the performance of network operations and energy efficiency. [31] and [63] propose slot assignment mechanisms for event-driven sensor network environments by achieving only the subset of total packets. If no activity is sensed, sensor nodes increase their transmission probability exponentially for the next slot assuming that only small amount of traffic exists in the network.

In [55], sensor nodes are partitioned into clusters and the cluster heads maintain a TDMA schedule to exchange data between cluster members and heads with no peer-to-peer communication. In order to communicate between clusters, CDMA code, which has been known as an expensive mechanism in WSNs, is used. In [32] and [33], the Self-Organizing Medium Access Control for Sensor networks called SMACS is proposed to combine neighborhood discovery and TDMA schedule assignment. The scheme assumes that sensor nodes are able to use many CDMA codes in many channels. In SMACS, a node uses fixed-length superframes, which do not need to use the same phase as the neighbor's superframes. All nodes, however, use the same superframe length and it requires time synchronization. In [49], Traffic-Adaptive Medium Access (TRAMA) is proposed. In TRAMA, sensor nodes access a single channel in a collision-free manner assuming that all nodes are time synchronized and the schedules are managed in a distributed manner on an on-demand basis. The protocol uses two time periods, random access and schedule access periods. During a random access period, each node broadcasts its schedule and neighbor information, and learns its two-hop neighbor information. It also sends a list of receivers for the packets in a queue periodically. A flexible-schedule-based TDMA protocol called FlexiTP is proposed in [60]. The main idea of FlexiTP is to provide fault-tolerant and energy efficient data report using flexible time slot allocation.

The main purpose of low duty cycle is to avoid waste of energy during idle state. Although the Sparse Topology and Energy Management (STEM) protocol does not cover all MAC functions, it provides a solution for the idle listening problem in [9]. It defines Wakeup and Data channels and uses the wakeup channel as a control channel to identify if there is transmission activity. In STEM-B, a node to transmit a packet sends beacons on the wakeup channel periodically without prior carrier sensing. As soon as a receiver sends an acknowledgment frame for the beacon, the transmitter and receiver nodes perform the transmission on the data channel. In STEM-T, a node uses a simple busy tone to provide cheaper and less energy-consuming transmission. S-MAC(Sensor-MAC) is proposed in [67, 68] where sensors within a certain virtual cluster manage local synchronization and agree the same schedule performing fixed sleep and listen time. In S-MAC, the listen schedule is required to coordinated in the virtual cluster so a node exchanges its schedule with neighbors in a SYNCH field. The SYNCH field is subdivided into time slots and neighbors contend with backoff scheme. After setting up the schedule, a node uses RTS/CTS handshake mechanism to reduce collisions of data packets. In usual nodes on the border of a cluster are required to manage more than one schedule and spend more energy. The listen period of S-MAC can be used to both transmit and receive packets. Due to the delivery latency in [67], [68] proposes the adaptive-listening scheme to reduce per-hop latency using CSMA with RTS/CTS collision avoidance mechanism and [52] proposes Timeout-MAC (T-MAC) for a node to go to a sleep mode when there is no activity during short time listening period. Dynamic Sensor MAC (DSMAC) proposed in [39] suggests to use dynamic duty cycle mechanism to decrease the latency by sharing one-hop latency values, which means the time difference between enqueued time and transmitted time. In [7], AS-MAC is presented with the adaptation of node listening to the dynamic network traffic. The sender notices the receiver the next expected listen time and the listen time is decided based on the data flow. In [40], DMAC is proposed to construct unidirectional convergecast tree from sensor nodes to the sink for data gathering in WSNs. The purpose of DMAC is to reduce latency and also achieve energy-efficiency. Low latency is achieved by assigning subsequent slots to the nodes that are successive level in the tree. Even though [40] shows good results for decreased latency compared to other methods, it does not provide collision avoidance mechanism for the nodes using the same time schedule in the same level of the tree and attempting transmission to the same node in the successive level.

While other works presented above do not focus on collisions between nodes, [31] and [63] propose slot assignment mechanisms. They work for event-driven sensor network environments achieving only the subset of total packets. If no activity is sensed, sensor nodes increase their transmission probability exponentially for the next slot assuming that only small amount of traffic exists in the network. [63] further presents how to decide a non-uniform probability distribution.

Sequential Assignment Routing (SAR) [33] is the first protocol that includes a notation of QoS routing decision in WSNs. SAR uses QoS matric which has two parameters, energy and QoS factors, on each path and the priority level of a packet. SPEED [51] ensures a certain delay for each packet so that applications can estimate the end-to-end delay for the packet by considering the distance to the sink and the speed of the packet before deciding the admission.

In [20], a labeling technique for energy efficient MAC headers is proposed to present dynamically assigned short link labels that are spatially reused for the MAC header. The simulation result shows that the number of bits can be reduced using the non-uniform label selection distribution.

QoS provisioning in the MAC layer deals mainly with the scheduling of packets on the wireless channel subject to local constraints. Since the local constraints may change based on the needs of individual flows, the decisions are generally very dynamic and must be computed rather fast. The following protocols consider the time constraint requirements while scheduling medium access. First, a QoS-aware medium access control protocol (Q-MAC) [65] assumes an environment of multihop wireless sensor networks where nodes may generate packets with different priorities. The objective of Q-MAC is composed of intra-node and inter-node QoS scheduling mechanisms. The intra-node QoS scheduling scheme classifies outgoing packets according to their priorities, while the inter-node QoS scheduling solution handles channel access with the objective of minimizing energy consumption via reducing collision and idle listening. The intra-node scheduling mechanism employs multiple first-in first-out (FIFO) queues with different priorities. The inter-node scheduling mechanism provides self-generated and relayed packets to be classified to different queues with several QoS metrics, such as content importance and number of traveled hops. Data rate allocation between queues and serving packet selection are achieved through the MAX-MIN fairness algorithm and the GPS algorithm [3], respectively. Our data report scheduling also considers the size of queue of data reporting node and the priority of packets. While Q-MAC is designed for packet processing in multiple queues, our scheme is to find out the threshold of a given queue to avoid data overflow in a queue and further congestion control in a network.

The coloring-based real-time communication scheduling (CoCo) [23] is also designed for multihop wireless sensor networks that use IEEE 802.11 MAC protocol. The network model is based on uni-cast communication model assuming that node locations are available all the times, and a central scheduler running CoCo is in charge of communication scheduling. The objective of CoCo is to schedule real-time communication avoiding collisions and minimizing the overall packet transmission time. Our framework is also based on a locally centralized control for report scheduling. The proposed scheduling scheme is implemented in single-hop cluster-based network model so that each cluster head can operate as a centralized controller.

On-demand multihop routing algorithms such as AODV and TORA eliminate table updates in high-mobility scenarios. However, they cause high-energy cost during route setup phase. The sequential assignment routing (SAR) [33] uses the idea of multiple paths while taking parameters like energy resource, QoS on each path, and the priority of packets into consideration. In SAR protocol, a table-driven multi-path approach is used to improve energy efficiency in a low-mobility sensor network. The failure protection is addressed by having at least k-paths that have no common branches between a node and a sink. Also, localized path restoration procedures are used to decrease energy cost in failure recovery. Each node uses the following parameters to establish routing paths: (1) energy resource estimated by maximum number of packets that can be routed without energy depletion, assuming that the node has exclusive use of the path (2) QoS metrics where higher metric implies lower QoS. In our framework, each cluster head maintains two data reporting paths to support (1) energy efficient and (2) delay constraint data reporting requests. Also, the data reporting path construction is based on the local information exchanged with adjacent cluster heads.

2.3 Data Management

In [55], LEACH proposed local data aggregation supporting node collaboration to reduce the energy cost of data transmission rather than sending every raw data to the data sink in a cluster-based network. Sensor nodes randomly and densely deployed in a monitoring area forms clusters, and each cluster head aggregates data collected from its member nodes before transmitting to the data sink. In LEACH, the required data aggregation level should be specified by the application.

In our dissertation, the network model and scenario is very similar with the one proposed in LEACH in that (1) the operations of our proposed protocols and algorithms are based on a cluster-based topology, (2) each cluster head collects information from its member nodes and performs data aggregation by reducing energy consumption using in-network processing.

Direct diffusion [15, 30] is a task specific data-centric routing protocol which supports an event-driven applications. Intermediate nodes in direct diffusion are capable of caching and transforming data. SPIN [25, 36] has several similarities with [15, 30]. SPIN uses high-level data descriptors, called the *metadata*, to name its data.

The metadata and raw data have a one-to-one mapping and the format of metadata is application-specific.

Several researches have proposed the idea to cover a certain portion of the network with a certain number of sensor nodes at a given time. The fundamental assumption of these works is that the subset of nodes are capable of providing information about events of interest required by the applications within a given sensing range.

k-coverage problem ensures that a certain area from the entire monitoring space can be covered with at least k sensors. A certain number of sensors are chosen to cover the desired monitoring area, also known as k-coverage and some related works are proposed in [62, 57]. Similar work about the area coverage is discussed in [6] defining the percentage of a particular area A. If $f_a = 1$, the full area of A is covered. In this dissertation, we use f_a as one scenario for our data collection coverage rate χ. The details are discussed in next Section.

In [19] and [61], Ye et al. and Wang et al. proposed the proposed probing environment and adaptive sleeping protocol, called PEAS, and a coverage configuration protocol, called CCP, respectively. They are coverage-preserving protocols to turn off the nodes as much as possible to reduce energy consumption while maintaining the desired coverage over the entire monitoring area. The desired coverage is expressed by the detection probability P_d based on the maximum distance form the sensor to a certain point (that an event happens) and the spatial resolution discussed in [17, 43].

2.4 Quality of Service and Cross-Layer Design

QoS has been the target of many communication protocols. In order to provide QoS, the following characteristics should be considered.

- **Resource estimation** The estimation of available resource in a given network and nodes in the network is important. During the configuration state, the network connectivity information, the capacity of nodes and links, and the allocated resources can be learned.
- **Required resource** The required performance requirements for a particular task should be calculated to sustain the QoS expectations using allocated resources. Both the performance metric and the resource requirement estimation may be managed in the network. Based on the information, the required resource is allocated/reserved in a particular network entities and deallocated after the required service.

Fidelity and scalability are other important design consideration factors. Our integrated framework is based on the knowledge that local decision-making is sufficient for scalability. The sensor network problem is to extract information concerning some physical phenomenon to within some fidelity, given nodes with some constraint on resources. As the density of nodes increases, the possibilities for spatial correlation of sensing results increases, and the information that must be delivered will saturate according to the fidelity threshold, if only nodes have a mechanism for determining which ones will be involved in some form of local fusion and which ones will report nothing.

Clearly neither communications relay strategy nor the local decision rules are optimal in information theoretic senses. Rather they are merely sufficient to ensure scalability. However, this result is highly suggestive of what an optimal strategy might look like under a fidelity constraint. Once a sufficient number of nodes are identified that achieve mutual information between observations and source phenomena above some threshold, then no more nodes need be involved. The details of this is discussed in Chapter 4, and our scheme proposes the selection of data reporting node that satisfy the task-specific throughput fidelity specified by the end system.

Some previous works related to cross-layer design in a wireless sensor network are presented in [60, 4, 37, 47, 28]. In [60], a flexible-schedule-based TDMA protocol, called FlexiTP, provides fault-tolerant and energy efficient data reporting using flexible time slot allocation based on a data gathering tree. MERLIN (MAC and efficient routing integrated with support for localization), proposed in [4], divides a network into several timezones and performs synchronization towards the center. It exploits multicast streams and presents scheduling and routing schemes in a timezone-based network. A cross-layer transmission scheduling in [47] is based on one-hop clustered networks. It takes advantage of clustering such as data fusion in a cluster head. In [37], a low energy self-organizing protocol in a dense sensor network is discussed. The protocol focuses on the interactions between application and MAC layers and considers the tradeoff between QoS support and energy consumption. In [64], a dynamic MAC protocol integrates the channel state and the residual energy parameters to maximize the network life time. Employing the tradeoff between the two parameters, high priority packets access a channel of quality.

Quality-aware sensing architecture, called QUASAR [27], provides a quality-aware query (QaQ). QaQ illustrates quality requirements and the queries are specified by the applications. A network operates to satisfy the quality requirements while minimizing the costs.

CHAPTER 3

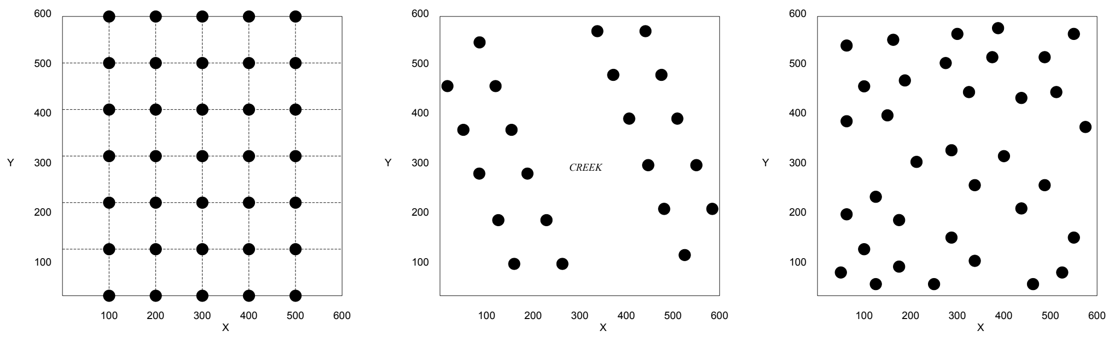
SENSOR NETWORK MODEL

In this chapter, we present sensor network scenarios where we can apply our integrated frame work for task-specific QoS-aware data reporting. We mainly consider dense networks in that a large number of sensors are deployed with high node density over a planned two-dimensional geographic area. The basic terminology and problems to be tackled for designing the framework are also described.

The rest of this chapter is organized as follows: In Section 3.1, we consider possible sensor deployment scenarios to cover the monitoring area in wireless sensor networks. Section 3.2 demonstrates homogeneous and heterogeneous network models that our framework focuses on. Section 3.3 presents basic notations and basic assumptions used to discuss the proposed schemes.

3.1 Sensor Deployment

In order to form a sensor network in the targeted monitoring area, a large number of sensors are deployed either *randomly* or *regularly*. In the regular deployment, a human operator or a system decides well planned fixed node positions and deploys sensors on the area. In the random deployment, a node position is unpredictable until a node has been deployed; for example, an aircraft may drop sensor nodes in the sky. Many existing research works assume that sensor nodes are randomly and uniformly deployed in the monitoring area in this kind of ways although the uniform random distribution of nodes is not easy to achieved practically. In fact, such random deployment can form various sensor deployments with different degrees of coverage



(a) (b) (c)

Figure 3.1. Sensor Deployment (a) Grid (b) Planned (c) Random.

that include sparse, dense, and even uncovered local network. Figure 3.1 illustrates three examples from the two deployment strategies.

Therefore, regular deployment is considered expensive in terms of the deployment time and required resources for the deployment, while random deployment sometimes generates problem such as coverage holes or isolated sensors, which are be able to communicate with neither other sensors nor the end system. In this dissertation, the applications of our integrated framework are not limited to any particular deployment. Two assumptions in a deployment scenario are (i) the nodes in a network form a connected graph even in the random deployment scenarios without isolation and (ii) every nodes in a network is static. Therefore, in order to solve coverage and connectivity problems in random deployments, we investigate a Poisson point process, which has been popularly assumed as a network model in the random deployment in existing research works [6, 56].

Where $\mathcal{N}$ number of nodes are deployed in a monitoring area $\mathcal{A}$, we apply the followings in a randomly deployed network model applying a Poisson point process with $\lambda > 0$. In a particular local area $\mathcal{A}^l \subseteq \mathcal{A}$, $\mathcal{N}^l$ is a random variable to represent the number of nodes in the local area $\mathcal{A}^l$ that follows a Poisson distribution with parameter $\lambda \cdot \mu(\mathcal{A}^l)$.

$$Pr[N(\mathcal{A}^l) = k] = e^{-\lambda \cdot \mu(\mathcal{A}^l)} \cdot (\lambda \cdot \mu(\mathcal{A}^l))^k / k!. \tag{3.1}$$

When $\mathcal{A}_1^l, \ldots, \mathcal{A}_n^l \subseteq \mathcal{A}$ are disjoint, the random variables $\mathcal{N}^1, \ldots, \mathcal{N}^l$ are independent. The k nodes are independent and uniformly distributed in $\mathcal{A}$ under the conditions of $\mu(\mathcal{A}) > 0$ and $N(\mathcal{A}) = k$. In order to find the coverage area $f_{\mathcal{A}}$, p and q are a randomly chosen point in the area $\mathcal{A}$. The probability that there is at least one sensor node s with $\|p_s - q_{\|_2}$ that is smaller than the sensing range r_s follows.

$$f_{\mathcal{A}} = Pr[\mathcal{N} \geq 1] = 1 - Pr[\mathcal{N} = 0] = 1 - e^{-\lambda \pi r_s^2} \tag{3.2}$$

According to Poisson distribution, each node has a random number of connected nodes and these variables are independent results of k. In order to validate the connectivity of a network, we define the distribution of the random variables k as g_k and the generating function as $G_1(s)$. Defining N_n is the number of nodes connected at the nth hop, $N_0 = 1$ when the initial node is a sink. In order to estimate the number of cluster members directly connected to a cluster head (in a single-hop), the proposed integrated framework defines the probability distribution g_k, and the corresponding generating function as

$$g_k = \frac{\mu^k}{k!} e^{-\mu} \tag{3.3}$$

$$G_1(s) = e^{\mu(s-1)} \tag{3.4}$$

respectively. The mean value of N_n is defined as

$$E\{N_n\} = \frac{dG_n(s)}{ds} \mid_{s=1} = \mu^n \tag{3.5}$$

Therefore, our integrated framework can estimate the number of number of nodes connected to a particular node (a cluster head) in both random and regular deployment models. As mentioned earlier, in regular deployment model the positions of sensor nodes are known before the deployment, and in random deployment model, on the other hand, our integrated framework calculates the estimated number using Eq. (3.4). Then, the framework updates the actual number of nodes by exchanging messages.

3.2 Network Model

In this section, we consider two network models with homogeneous sensor devices and different types of heterogeneous devices so that high capable device can operate as a cluster head with high communication and processing capabilities.

Assuming that a large number of sensor nodes are deployed in the monitoring area with high node density, as mentioned in the previous chapter, we define a wireless sensor network as an undirected connected graph $G = (V, E)$, where V is the set of nodes in the network and E is the set of bidirectional wireless links representing direct communication between sensor nodes or between sensor nodes and other wireless devices within the radio range. Each node learns its local connectivity at the network setup time and periodically adopts the dynamic topology changes, caused by node failures, environmental problems, and so on. An initial graph is formed from the end system during the network setup time, and each node $s_i \in V$ periodically updates the information of its local neighbors and learns the next hop destination based on the routing decision unless the transmission is broadcast. The local connectivity is defined based on the radio communication coverage. Two different forms of coverage are defined as follows.

- **Communication coverage** This coverage is based on the radio range r_{s_i}, typically transmission range, of a node s_i. In many cases, the communication coverage area, denoted by $\mathcal{A}^r_{s_i}$, is assumed as circles, but the physical barriers are hard to model. If two nodes are within their communication coverage range, the nodes are locally connected.
- **Sensor coverage** This coverage is important when the purpose of wireless sensor networks is considered. While communication coverage is used for data exchanges between two nodes, sensor coverage is used to represent the coverage of the monitoring area, denoted by $\mathcal{A}^s_{s_i}$, to detect required physical phenomena based on the sensing application objective. In other words, each point in the area of interest should exist within the sensor coverage of at least one sensor.

In the following, we will distinguish between communication coverage and sensor coverage. We also define the following basic terms of wireless sensor networks, used to describe a sensor network model in this dissertation.

- **Sensor** A sensor or sensor node, denoted by s_i, is a source of the information in the network.
- **Cluster head (CH) and cluster member (CM)** The entire network is divided into several clusters, each having a cluster head that is responsible for data reporting control. Each clust4er head collects data from its cluster members (CM),
- **Sink** A sink or a data sink, denoted by d_i, collects information from sources. We consider two options for a sink. The first option is that a sink operates as a sensor as well. For the second option, a sink can be other types of wireless devices; for example, a particular type of a wireless actuator node used to

interact with the sensor network or a gateway node to another types of network such as the Internet.

- **End system** The end system is defined as a final data collector. A sink might be the end system. In this dissertation, we distinguish between a sink and the end system assuming that a network may consist of more than one sinks and the sinks finally transmit the collected information to the end system.

For the rest of the discussion in the dissertation, we focus on the scenario that the entire network area is divided into several clusters, and each cluster takes the responsibility of local data reporting control in that area in a distributed manner. Each cluster collects data from its cluster members, performs data aggregation, and forwards the results to a sink. A cluster head and the members can communicate with each other in a single hop manner.

Let n_i be the number of member nodes belonging to the cluster head c_i. Each node learns its neighbor information during the initial network setup time and periodically updates the information. A node maintains a neighbor table until two-hop neighbors. Adjacent neighbors of a node s_i are defined as $V_{s_i} = \{k \mid k \in \{c_{s_i} \cup c_{s_j}\}\}$, where $c_{s_i} = \{s_j \mid d(s_i, s_j) \leq r_{s_i}\}$, $c_{s_j} = \{s_h \mid d(s_j, s_h) \leq r_{s_j}\}$, and r_{s_i}, r_{s_j}, and r_{s_h} represent each radio range of nodes s_i, s_j, and s_h, respectively.

As a cluster head operates as a slot allocation agent and membership management agent, it runs a conflict-resolution algorithm (e.g., graph coloring or schedule exchanges schedule with another member) when a member informs the overlapped schedule between the member and interfering nodes. Section 4.2 describes the detailed roles of a cluster head.

In this scenario, two types of communications exist as described illustrated in 3.2; the one is intra-cluster communications between nodes in the same cluster and the other is inter-cluster communications between nodes belonging to different clusters.

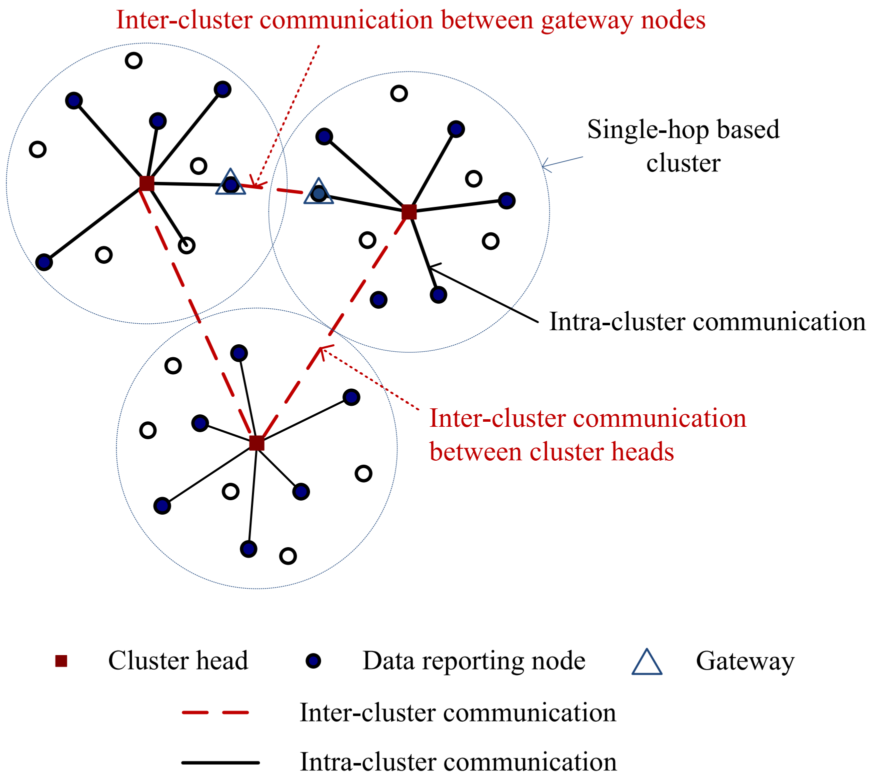
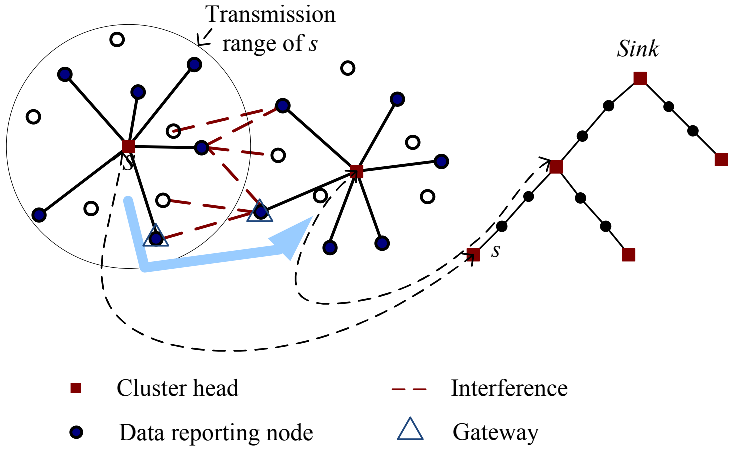


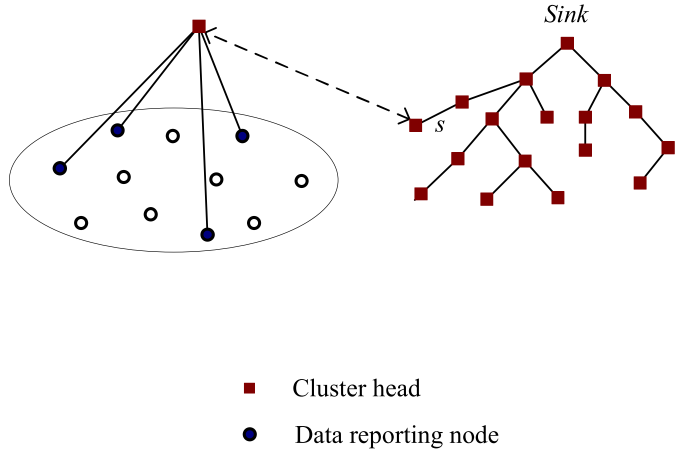
Figure 3.2. Communication Types in a Cluster-Based Network.

When a cluster head forwards data to an intermediate cluster, the cluster head that receives data may perform fusion depending on the application requirement. Each member sends data only to its cluster head with no peer-to-peer communication. Every cluster head is synchronized with each other and with its cluster members. The details are presented in Chapter 4.

We consider two network models as presented in Figure 3.3. One is a homogeneous network, which consists of one type of nodes with all the same functions and capabilities, and the other is a heterogeneous network, which has two different types of nodes.



(a) Homogeneous network



(b) Heterogeneous two-tier network

Figure 3.3. Network Model.

3.2.1 Homogeneous Networks

In a homogeneous network, a large number of homogeneous sensor nodes are deployed with high node density in the monitoring area. They have the same capabilities in terms of energy capacity, the transmission range, processing capacity, and so on. As cluster heads are required to control the data reporting in a local area, the energy consumption of them is much more than other member nodes. Therefore, the dynamic exchanges of the role as a cluster head is necessary in order to distribute the energy consumption.

The network model in our integrated framework is based on a single-hop cluster-based topology such that a cluster head manages the information of one-hop neighbors as members and two-hop neighbors as interfering nodes. In this scenario, a multi-hop communication is essential for a cluster head to reach to the other cluster head. A cluster member that is used for inter-cluster communications is defined as a gateway node.

3.2.2 Two-tier Heterogeneous Networks

The second model considered in this dissertation is heterogeneous wireless sensor networks, in which cluster heads with higher communication capabilities manage each of the clusters. Normal nodes have short transmission ranges and are inexpensive that allows for a large number of sensors to be deployed in the network. These nodes operate as cluster members and report their sensing results to the cluster head that has more energy, higher processing capability, and longer communication ranges so that each cluster head can directly reach to its adjacent cluster heads in a single-hop manner. In this dissertation, we limit the function of cluster heads on processing and communication capabilities in this work although they may have sensing capability depending on the types of devices.

3.3 List of Basic Notations and Summary of Assumptions

- A large number of sensors are deployed in the monitoring area forming a *dense network*.
- Every node in a network is *static*.
- A network forms a *connected* graph.

Table 3.1. Basic Notations used in the DRC Framework

Term	Description
$\mathcal{A}$	Area of interest (= monitoring area)
V	Set of nodes in a monitoring area
E	Set of edges between nodes
$\mathcal{N}$	Total number of sensor nodes in a monitoring area
V_{s_i}	Adjacent neighbors of node s_i
$\mathcal{A}^l$	Local area
$\mathcal{N}^l$	Number of nodes in local area $\mathcal{A}^l$
s_i	Sensor node where $1 \leq i \leq \mathcal{N}$
r_c	Radio communication range
r_s	Sensing range
$\mathcal{A}_{s_i}^{r_c}$	Communication coverage area
$\mathcal{A}_{s_i}^{r_s}$	Coverage of monitoring area
M_{c_i}	Set of cluster members belonging to cluster head i
θ	Number of prioritized sensing task classes in a network
$\mathcal{N}_i$	$(= \lvert M_{c_i} \rvert)$ Number of cluster members
n_i	Number of cluster members belonging to a class i $(n = \sum_{i=1}^{\theta} n_i)$
Φ_i^E	Energy parameter for a class i where $1 \leq i \leq \theta$
Φ_i^D	Delay parameter for a class i
Φ_i^T	Throughput parameter for a class i
θ	Number of priority groups in the network.
$\mathcal{T}$	Total number of slots in one cycle. $\mathcal{T} = \sum \mathcal{T}_i$
$\mathcal{T}_i$	Number of time slot for group i $(1 \leq i \leq \theta)$
$\mathcal{P}$	Set of grading patterns. $\mathcal{P}_k$ means kth grading pattern
ω_1, ω_2	Weighting values
α	Number of accessible slots assigned to a particular sensor
λ_i	Arrival rate at node i
μ_i	Service rate at node i
λ_a	Arrival rate after data aggregation
β	Number of data reporting node in a cluster
E_i^{res}	Remaining (residual) energy of node i
E^{ths}	Threshold of energy specified by the user
p_n^i	Probability that node i has n number of packets
L_i	Size of a packet generated from node i
R_i	Data rate of node i
ε	Dept of a tree

CHAPTER 4

FRAMEWORK OVERVIEW

The proposed integrated framework [21] for QoS-aware data reporting control, called DRC, separates data reporting control within a cluster and data deliver from a cluster to another cluster. In other words, intra-cluster data reporting control is designed for a cluster head to collect data from its cluster members while inter-cluster control is designed for each cluster to transmit the collected data or forwarded data from adjacent cluster heads to a sink. Our DRC scheme is based on the knowledge that local decision-making is sufficient for scalability. In this section, we first present the functional architecture and the basic operations of our DRC framework, as illustrated in Figure 4.1, and define the problems we address and assumptions for these problems.

4.1 QoS Provisioning

Depending on the types of sensing tasks, the end system in a sensor network requires different QoS parameters and the different levels of the parameters to collect data from the monitoring area. The QoS provisioning process is performed to learn the required QoS parameters and the level of quality. These can be defined by the users or the system itself. As an example, the QoS requirements are analyzed while being integrated with the modulation and transmission schemes [66]. During this process, various data collection requirements depending on different types of tasks are categorized into different classes. The proposed DRC framework focuses on three main QoS parameters: the throughput fidelity and the end-to-end delay constraint at the end system, and energy efficiency. Also the rate of data collection coverage,

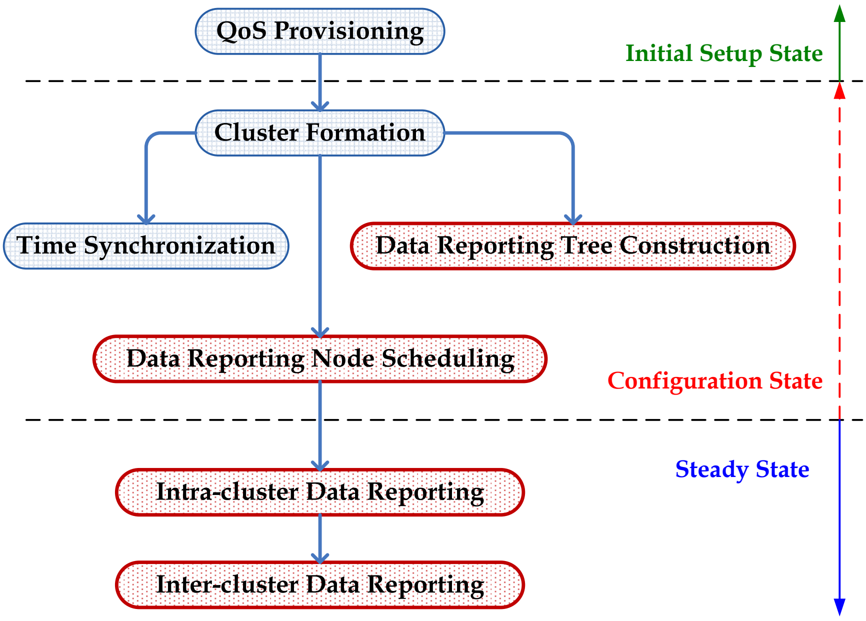


Figure 4.1. Functional Architecture.

and the frequency of data report, and fairness are considered depending on the types of tasks. The categorized QoS parameter values may be installed before nodes are deployed in a network or disseminated as global parameters.

QoS provisioning in the MAC layer deals mainly with the scheduling of packets on the wireless channel subject to local constraints. Since the local constraints may change based on the needs of individual flows, the decisions are generally very dynamic and must be computed rather fast. The following protocols consider the time constraint requirements while scheduling medium access. First, a QoS-aware medium access control protocol (Q-MAC) [65] assumes an environment of multihop wireless sensor networks where nodes may generate packets with different priorities. The objective of Q-MAC is composed of intra-node and inter-node QoS scheduling mechanisms. The intra-node QoS scheduling scheme classifies outgoing packets according to their priorities, while the inter-node QoS scheduling solution handles channel access with the objective of minimizing energy consumption via reducing collision and idle listening. The intra-node scheduling mechanism employs multiple first-in first-out (FIFO) queues with different priorities. The inter-node scheduling mechanism provides self-generated and relayed packets to be classified to different queues with several QoS metrics, such as content importance and number of traveled hops. Data rate allocation between queues and serving packet selection are achieved through the MAX-MIN fairness algorithm and the GPS algorithm [3], respectively. Our data report scheduling also considers the size of queue of data reporting node and the priority of packets. While Q-MAC is designed for packet processing in multiple queues, our scheme is to find out the threshold of a given queue to avoid data overflow in a queue and further congestion control in a network.

More specifically, the DRC framework considers throughput as the main QoS parameter in the intra-cluster data reporting control (IntraDRC) scheme while the end-to-end delay and energy efficiency are the ones for the inter-cluster data reporting control (InterDRC) scheme. IntraDRC guarantees that the local throughput at each cluster head after collecting data from its data reporting nodes satisfies the desired throughput. InterDRC focuses on delay and energy parameters by using the information on hop counts to a sink and the traffic amount in a cluster for maintaining the data reporting tree. If a particular delay bound is specified, a cluster head reports data using a shortest path. By calculating hop counts to a sink, a head estimates the reporting latency. If the latency exceeds the required delay bound, a cluster head reports an error to the end system. In other cases without delay constraints, InterDRC constructs a data reporting tree in an energy efficient manner by balancing the traffic load. InterDRC concentrates on learning routing paths from each cluster to a sink through one or more intermediate clusters. The details on IntraDRC and InterDRC are discussed in the following Chapters.

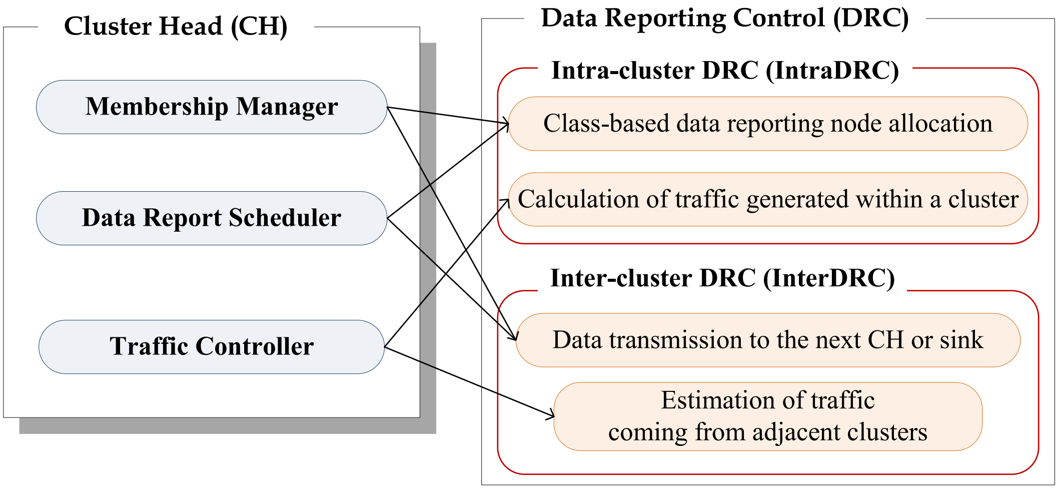


Figure 4.2. Roles of a Cluster Head.

4.2 Cluster Formation

Densely deployed sensor nodes are grouped together forming a cluster-based topology. By forming such clusters, the network can be managed in a distributed manner [55]. Within a cluster, each cluster head acts as a centralized controller to manage the data exchanges between the nodes in the cluster using IntraDRC. Then, it reports the collected data to a sink through one or more clusters using InterDRC.

In the DRC framework, each cluster performs three main functions as a local controller, as illustrated in Figure 4.2: a membership manager, data report scheduler, and a traffic controller.

As a membership manager, a cluster head maintains the neighbor table that includes the information of its member nodes, used in IntraDRC, as well as adjacent cluster heads, used in InterDRC. After collecting the cluster member information by exchanging 'hello' and 'reply' messages, a cluster head keeps the class information about prioritized tasks that a sensor node operates for in the neighbor table and calculates the number of sensor nodes belonging to each class. Where a cluster head cannot directly communicate with adjacent cluster heads, it also has to manage the gateway node information, which helps communication with a particular cluster head.

The functions performing as data report scheduler is the core part in DRC. In IntraDRC, a scheduler includes two functions: data reporting node selection and data report scheduling as discussed in Chapter 4. Briefly, a scheduler selects a subset of data reporting nodes among cluster members and schedules the data reporting of reporting nodes. The novel part of this operation is data reporting node allocation to a particular data reporting time slot. To the best of our knowledge, channel access scheduling is based on slot allocation/assignment in existing research works, but in our framework, scheduling is based on class-based node scheduling. The details are discussed in Section 5.3. In InterDRC, a scheduler manages data delivery from a cluster head to the next cluster head, eventually to a sink. While data report scheduling in IntraDRC operates based on either contention-based or schedule-based scheduling mode, the scheduling in InterDRC operates based on only schedule-based mode.

As a traffic controller, a cluster head analyzes the amount of traffic generated in the cluster and the traffic amount forwarded from other clusters in order to balance the traffic load and deal with the overall network contention [45, 54]. The DRC framework focuses on throughput fidelity in IntraDRC to decide the number of data reporting nodes in a cluster, while it focuses on end-to-end delay and energy efficiency in InterDRC to establish traffic-adaptive data reporting paths. As the throughput requirement is different depending on the types of tasks, the DRC framework first learns the level of throughput fidelity for each class during the QoS provisioning procedure. Although the practiced throughput fidelity is the required value at the end system, IntraDRC uses the requirement for local data report control information so that the locally satisfied throughput performance eventually provides the global

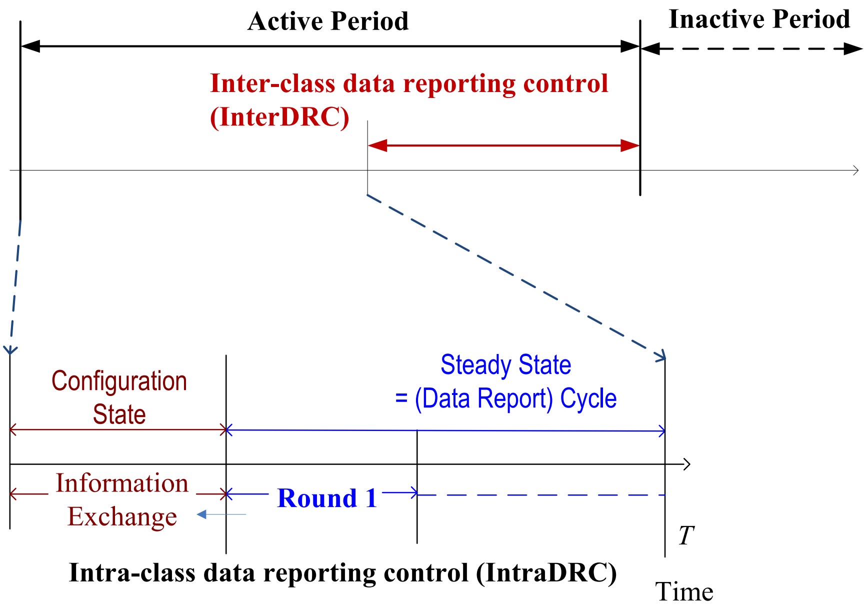


Figure 4.3. Cluster-Based Data Reporting.

throughput performance. If a specific delay constraint is specified, InterDRC builds data reporting paths to satisfy the delay requirement; otherwise, it constructs the data reporting tree in an energy efficient manner by balancing the traffic load, while the data reporting tree construction concentrates on learning routing paths from each cluster to a sink through one or more intermediate clusters.

4.3 Time Synchronization

Our DRC framework performs in time synchronized sensor networks, and Figure 4.3 describes the schedule of intra-cluster and inter-cluster data reporting. The overall network is loosely synchronized such that adjacent clusters are synchronized with each other and the synchronization is performed towards the end system. The IntraDRC scheme is to keep the cluster members as tightly synchronized as possible with the cluster head. For time synchronization, each member in a cluster is synchronized with its cluster head in advance. Then, a cluster head synchronizes with the other

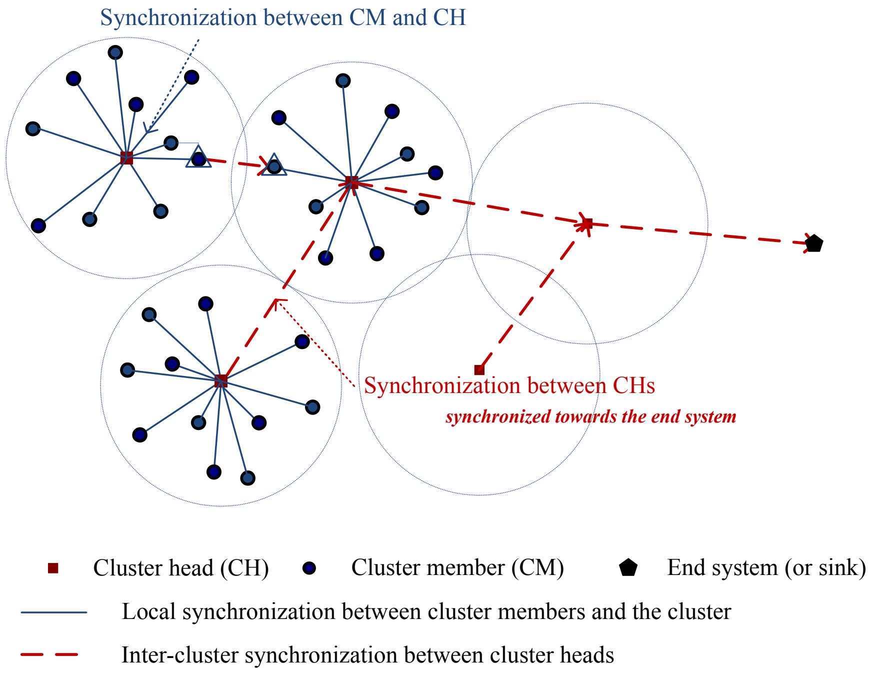


Figure 4.4. Time Synchronization in the DRC Framework.

nodes in a different cluster that is close to the end system. The time synchronization approaches can be applied by gradually synchronizing nodes towards a particular node that will be the end system [4, 42], as illustrated in 4.4.

CHAPTER 5

CLASS-BASED DATA REPORTING STRATEGY

As mentioned earlier, a sensor network can operate for multiple sensing tasks while each task requires its unique sensing operation configurations. We categorized the following example cases that a sensor network has prioritized multi tasks with a simple sensing application, Fire detection.

- Depending on the event, a sensor network has different data reporting priority, as described in Table 5.1. In the normal state, temperature sensors operate in a energy efficient mode to extend the network life time. The sampling rate is 1 per second; thus, the traffic in a network is low. The duty cycle of sensor nodes is low so that the nodes reduce the energy consumption cause by idle listening or data communications. However, once a fire has broken out, the sensors operate in a delay constraint mode in order to report the critical event as soon as possible to the end system. The sampling rate is changed to 1 per millisecond; thus, a network has burst traffic.

Table 5.1. Task-Specific Data Reporting

	Temperature monitoring	Fire detection
Data reporting type	periodic	event-driven
Sampling rate	low	high
Traffic	low	high
Data reporting mode	energy efficiency	delay constraint

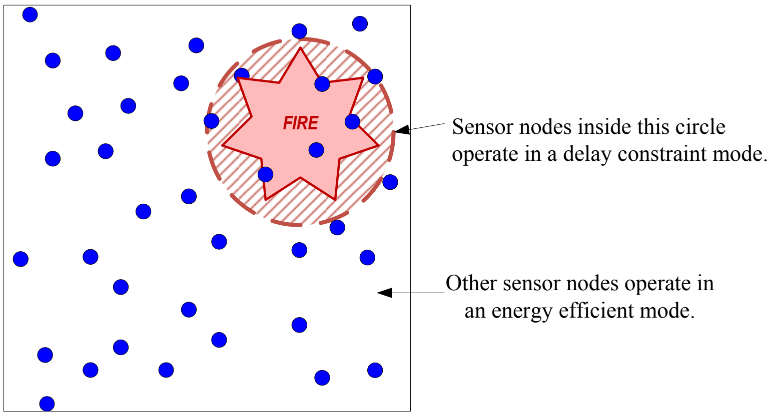


Figure 5.1. Data Reporting Based on the Area.

- Depending on the location, sensor nodes operate for different tasks, as illustrated in Figure 5.1. In the area that sensor nodes detect abnormal temperature, the nodes operate for the end system to decide whether a fire breaks out or not. The other nodes in different area operate for periodic temperature monitoring.
- Depending on the sensing type, a node has different data reporting priority. Temperature sensors need one time slot to transmit accumulated 10 sensing results by combining the values in one data packet. On the other hand, multimedia sensors such as image sensors and smart cameras require more than one time slots in order to transmit one application layer data. As the information extraction is delayed if even one data packet is missing, the nodes require to reserve multiple time slots for data transmission at one time.

The differentiated operation of nodes for data reporting, based on the sensing task, yields optimized task-specific QoS, further eliminating unnecessary energy consumption. Our class-based data reporting strategy inherently achieves the following two main advantages.

- Reduced competition: sensor nodes compete with other nodes in the same priority class only.
- Differentiated service: sensor nodes in different classes have differentiated operation configurations for data report scheduling.

We first describe a *two phase node scheduling* (TNS) scheme [22] that is developed as a basis of the proposed intra-cluster data reporting control (IntraDRC) strategy, which manages data reporting from cluster members to the cluster head within a cluster. The main goal of the TNS scheme is to extend the network lifetime and provide an adaptive data reporting strategy based on the task-specific requirements by reducing unnecessary competitions with all other nodes in a cluster. The scheme controls data reporting schedules of sensor nodes based on the priority class of the sensing task in a locally centralized manner within a cluster.

The TNS scheme consists of two phases. In phase I, called *class-based round allocation* (CRA), data reporting time slots are divided into separate data reporting round defined for each class. The time period of data reporting rounds defined for all classes at least one time is defined as data reporting cycle, and this cycle may be repeated as many as the sensing application requires. In phase II, called *QoS-aware node allocation* (QNA), the sensor nodes in the same class are scheduled to particular time slots depending on the given number of slots calculated in the first phase. Sensor nodes are scheduled in either a contention-based mode or schedule-based mode. Regardless the access mode, nodes may be scheduled more than one time in a data reporting round. As a default, a node will be scheduled more than one time in the case that the number of nodes is less than the number of time slots in the data reporting round. In TNS, we consider the following QoS parameters: throughput fidelity, the data collection coverage rate, and fairness. The categorized

QoS parameters for each class may be pre-installed before the nodes are deployed in the network or flooded after deployment.

Then, we describe the concept of β-coverage, which is designed to define the number of data reporting nodes in a cluster. The motivation comes from three characteristics of wireless sensor networks: the spatial correlation of sensing results in a cluster, dense deployment of wireless sensor networks, and task-specific throughput fidelity. Cluster members belonging to the same cluster head have high correlation, detecting the same phenomena in the local area. As IntraDRC is for data reporting control within a cluster, we focus on the trade-off between data redundancy and data accuracy required by the end system. Rather than maximizing throughput, wireless sensor networks require optimized throughput performance that means enough amount of data to extract information from while reducing redundant data as unnecessary redundancy requires extra energy consumption. Also, high data traffic causes high probability of collisions, which result in the waste of energy by failed data transmission and retransmission, if required by the application. In densely deployed wireless sensor networks, providing optimized amount of data to the end system is important. Therefore, in IntraDRC each cluster head controls the amount of data by deciding the number of data reporting node while satisfying the throughput fidelity specified by the sensing task/application.

The rest of this chapter is organized as follows: Section 5.1 introduces basic notations and assumptions. Section 5.2 presents how the DRC scheme decides the number of data reporting nodes. In Section 5.3, we describe the proposed two-phase node scheduling (TNS) scheme in detail. In Section 5.4, we investigate the performance of our class-based data reporting strategy, and Section 5.5 summarizes the chapter.

5.1 Preliminaries

As mentioned earlier, our system model is based on single-hop cluster-based sensor networks, and each cluster head, serving as a membership manager and data reporting scheduler as well as traffic controller, executes the TNS scheme. Sensor nodes join only one cluster by selecting the cluster head with stronger signal strength. When the signal strengths from more than one cluster heads is not comparable, a node selects a cluster head from which the node receives a message first. A sensor node s_i operating as a cluster head is denoted by c_i in this chapter. The set of cluster members of c_i is denoted by M_{c_i}. We define the number of cluster members $|M_{c_i}|$ as n. Table 5.2 summarizes the above notations. Now, let us introduce basic definitions used to describe the class-based data reporting strategy.

Definition 5.1.1: We define a *class* as a prioritized sensing task in the network, and the number of classes is defined as θ. When a sensor network performs single task, there exists only one class in the network and $\theta = 1$.

Definition 5.1.2: The QoS parameter for a class i, where $1 \leq i \leq \theta$, is defined as $\Phi_i^{\{description\}}$, where {description} is substituted for energy, delay, and throughput as Φ_i^E, Φ_i^D, and Φ_i^T, respectively.

Definition 5.1.3: The number of cluster members belonging to a class i, where $1 \leq i \leq \theta$, is defined as n_i, where $n = \sum_{i=1}^{\theta} n_i$.

Fidelity and scalability are important design consideration factors. Our integrated framework is based on the knowledge that local decision-making is sufficient for scalability. The sensor network problem is to extract information concerning some physical phenomenon to within some fidelity, given nodes with some constraint on resources. As the density of nodes increases, the possibilities for spatial correlation

Table 5.2. Summary of notations used in TNS

Term	Description
M_{c_i}	Set of cluster members belonging to cluster head i
θ	Number of prioritized sensing task classes in a network
$\mathcal{N}_i$	$(= \|M_{c_i}\|)$ Number of cluster members
n_i	Number of cluster members belonging to a class i $(n = \sum_{i=1}^{\theta} n_i)$
Φ_i^E	Energy parameter for a class i where $1 \leq i \leq \theta$
Φ_i^T	Throughput parameter for a class i
θ	Number of priority groups in the network.
$\mathcal{T}$	Total number of slots in one cycle. $\mathcal{T} = \sum \mathcal{T}_i$
$\mathcal{T}_i$	Number of time slot for group i $(1 \leq i \leq \theta)$
$\mathcal{P}$	Set of grading patterns. $\mathcal{P}_k$ means kth grading pattern
ω_1, ω_2	Weighting values
α	Number of accessible slots assigned to a particular sensor
λ_i	Arrival rate at node i
μ_i	Service rate at node i
β	Number of data reporting node in a cluster
E_i^{res}	Remaining (residual) energy of node i
E^{ths}	Threshold of energy specified by the user
p_n^i	Probability that node i has n number of packets
L_i	Size of a packet generated from node i
R_i	Data rate of node i

of sensing results increases, and the information that must be delivered will saturate according to the fidelity threshold, if only nodes have a mechanism for determining which ones will be involved in some form of local fusion and which ones will report nothing.

The degree of correlation for sensing results is related to the distance between two sensor nodes so that it is very likely for adjacent nodes to generate redundant data. Therefore, every node does not need to transmit all data to the data sink to decide a certain event. As an example, to determine whether a fire has broken out or not, only a certain number of sensors within an area can report their data to a cluster head and the head aggregates the data. In this way, the amount of data to be transmitted is reduced and the energy consumption can also be reduced since the number of communications is decreased and the probability of collisions is also decreased if a network uses a contention-based channel access mechanism. In some cases, each individual node should be fairly reliable. Optimization of the scheduling of sensing tasks and data reporting is important for the expected power consumption as well as the required QoS support.

5.2 Data Reporting Node Selection

In this Section, we consider the following QoS parameters for data report scheduling for QoS support at the end system: energy efficiency, the rate of data collection coverage, and the frequency of data report. The categorized QoS parameter values can be installed before nodes are deployed in a network or disseminated as global parameters. We first present class-based node selection used in the intra-cluster data reporting control strategy in order to decide class-based data reporting nodes from cluster members within a cluster. In some cases, applications may not require data collection from all nodes, especially in densely deployed networks [56]. Therefore, a cluster head collects data from a subset of members in the cluster as long as the data collection is enough to support the required information quality at the end system.

The main objective of data reporting node selection is providing a locally adaptive data reporting strategy based on the task-specific QoS requirement, which is defined as the throughput fidelity required by the end system depending on the types of sensing tasks. The optimized data reporting in each cluster eventually improves the overall network performance. Although the report scheduling decision is made by a cluster head, these operations can be initiated by either a cluster member or a cluster head. During the configuration state, a cluster head probes its cluster members. When a cluster member sends back a probe reply message, the member can set the scheduling request field to '1' if necessary. The details are discussed below. If a cluster head is able to assign the required reporting schedule, it updates the schedule matrix and broadcasts the matrix; otherwise, it sends no extra message.

As IntraDRC is performed on a cluster basis, the scheme efficiently learns, applies, and controls the local situation without any communication overheads with other nodes. IntraDRC defines the degree of data correlation and the threshold of a queue as local parameters in order to help node selection and report scheduling. The detailed usage of each parameter is discussed in the following subsections.

IntraDRC controls the amount of traffic generated in a cluster by selecting a certain number of reporting nodes while meeting the required coverage rate, which is defined as task-specific throughput fidelity, denoted by Φ_i^T for class i. As the reliability of reporting paths may not be 100%, the acquired amount of traffic from sensor nodes will be greater than the user-specific coverage rate considering the delivery error rates. Assuming that each node i generates the same amount of data $L_i(t)$ at a given time t, the total offered traffic during the time period T is $\int_0^T L_i(t)dt$. A cluster head groups the member nodes in a cluster into β blocks and selects one node from each block at each time slot as a reporting node so that β nodes report data to the cluster head and the total offered traffic during the reporting period $\mathcal{T}$ is $\int_0^{\mathcal{T}} (\sum_{i=1}^{\beta} L_i(t))dt$.

Assuming that each node i generates the same amount of data $L_i(t)$ at a given time t, let p_i be the probability that i packets are in the node. The mean number of packets in the node is

$$\overline{L_i(t)} = \sum_{i=0}^{L_i} (i * p_i). \tag{5.1}$$

The total offered traffic during the time period T is

$$\int_0^T L_i(t)dt, \tag{5.2}$$

and throughput of node i, denoted as TH_i, is calculated as follows.

$$TH_i = \frac{\overline{L_i(t)}}{R}, \tag{5.3}$$

where R is the system response time.

The mean queueing delay to be referred for the expected end-to-end delay measurement is also calculated as follows. Where $\overline{L_i^{queue}(t)}$ is the mean number of packets in the queue of node i,

$$\overline{L_i^{que}(t)} = \sum_{i=1}^{L_i} ((i-1) * p_i). \tag{5.4}$$

Therefore, the mean queueing delay D_queue is

$$D_{que} = \frac{\overline{L_i^{que}(t)}}{TH_i}. \tag{5.5}$$

A cluster head groups the member nodes in a cluster into β blocks and selects one node from each block at each time slot as a reporting node so that β nodes report data to the cluster head and the total offered traffic during the reporting period T is

$$\int_0^T (\sum_{i=1}^{\beta} L_i(t))dt. \tag{5.6}$$

The goal of data reporting node selection is to maintain intra-cluster traffic load based on the user-specified requirements on the sender side in a distributed manner and eventually control the overall amount of traffic in a network. By controlling the number of reporting nodes, a cluster head can estimate the expected level of collisions or manage the report schedule for each node and avoid unnecessary extra reporting by adopting the required QoS information, defined as the collection coverage rate in this scheme. As the IntraDRC strategy is performed in a cluster basis, the scheme efficiently learns, applies, and controls the local situation without any communication overheads with other nodes.

5.2.1 β-Coverage

In order to decide β reporting nodes in a cluster, we consider three aspects that might be given from the applications.

Firstly, throughput fidelity of the application is defined as the expected data collection rate χ. Let us L_i be the amount of data reported by node i. When the applications require to have at least L_{app} throughput performance, cluster heads need at least $\sum_{i=1}^{\beta} L_i$, from β nodes since the reliability of reporting paths may not be 100%. Therefore, $L_{req} = L_{app} + ERR_{path}$, where L_{req} is the required throughput collected by a cluster head from sensor nodes in a cluster and ERR_{path} is the path error rate. In this case, β reporting nodes to satisfy $\int_0^T (\sum_{i=1}^{\beta} L_i(t))dt \geq L_{app} \cdot (1 + ERR_{delivery})$ are selected to transmit sufficient information even after suffering some deliver errors.

Secondly, a certain number of reporting nodes β or a certain percentage of nodes χ can be specified by the application. In some case, data reporting from only a subset of nodes will be enough for the end-system to learn the information about the area. In this case, $\beta = \lceil \mathcal{N} * \chi \rceil$. For example, when a fire breaks out in a forest, the end-system does not need to collect data from all nodes as long as the amount of data received is enough to decide whether a fire has broken out or not. The transmission attempts from a small number of nodes will also help the probability of successful transmissions minimizing collisions [63]. For the second example, sensing results from 10 nodes in each local area needs to be collected by the application to learn the statistical information.

Thirdly, a certain number of sensors are chosen to cover the desired monitoring area, also known as k-coverage problem [62, 57, 58, 6]. For example, only 80% of the entire monitoring area can be required to cover at a given time and k nodes are sufficient to cover the desired area. In this case, the desired monitoring area is defined as χ and β is equal to k. When $\chi = 1$, every node is expected to report data.

A cluster head analyzes the throughput from β data reporting nodes for three data reporting rounds as default, where one data reporting round consists of β time slots. If the throughput performance is less than Φ_i^T, it increase β by one.

5.2.2 Node Selection

Each cluster head analyzes the sensing results collected from members and maintains the spatial/temporal correlation between nodes as discussed in [8] in order to select β data reporting nodes among $\mathcal{N}_{c_i}$ cluster members. For the decision of node selection, three parameters can be considered based on the end system requirement: correlation of sensing values, the remaining energy level, and fairness in terms of data report chance. Block design has been applied for wakeup scheduling in the field of communications in WSNs [69, 46, 70]. In IntraDRC, a cluster head creates a distinguished cluster member node sets using block designs and the nodes in the same group are defined as potential contenders, which may attempt data transmission in the same time slot.

Figure 5.2 illustrates the data reporting node selection example, where $\mathcal{N} = 11$, $\chi = 0.6$, and $\beta = 8$ in a homogeneous network. When nodes are homogeneous, a cluster head is always included as one of β reporting nodes so that the energy consumption caused by the communications with one member is reduced. Therefore, a cluster head chooses β - 1 reporting nodes that is 7 in the example. In the first round, $\{1, 3, 4, 6, 7, 9, 10\}$ are selected among 11 member nodes while, in the second round, other 7 nodes $\{2, 3, 4, 5, 8, 9, 11\}$ including 3 duplicate ones $\{3, 4, 9\}$ are selected.

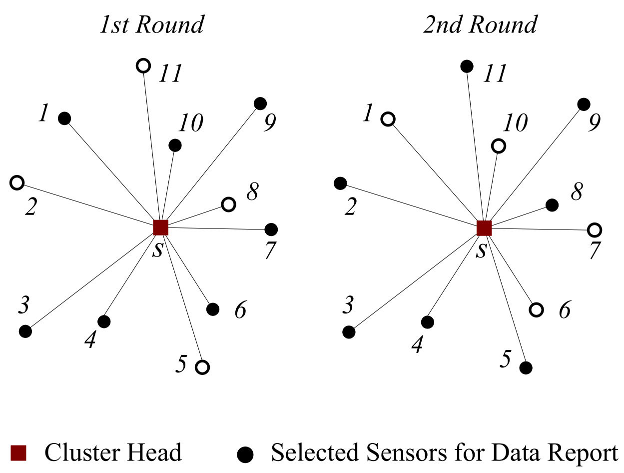


Figure 5.2. Data Reporting Node Selection.

5.2.2.1 Block Design

Now let us recall a block design in Combinatorial theory with n distinct objects into β' blocks such that the jth block contains the k number of distinct objects, and each object occurs in α different blocks. In WSNs, block design theory has been applied for wakeup scheduling in communication field as proposed in [69, 46]. Applying the block design concept into β reporting node selection, n members in a cluster are grouped into β' blocks such that jth block $\mathcal{B}_j$ contains the k nodes in each block and each node occurs in r different blocks. Each pair of node can appear α times among β' blocks.

r is set to 1 when the applications do not want to have redundant nodes in different blocks, and α is set to 1 when each node is required to belong to only one block. In this case, the number of nodes assigned in each block is $\lceil n/\beta \rceil$ or $\lceil (n/\beta)) - 1 \rceil$.

Where k is fixed in (n, β, r, k, α) balanced block design, a possible design drawback is possible combinations of each parameter value is limited such that $k = (nr)/\beta$. To solve this problem, we apply the level of correlation between nodes to design reporting node blocks. Let $y_{ij} = corr(x_i, x_j)$ be the function to calculate the correlation between two nodes i and j. A cluster head chooses a block with low correlated nodes first and groups the remaining nodes in the blocks with high correlated nodes. Another approach is to fix the value r as one so that each member appears in a block at one time and a k value for each block varies. The number of nodes belonging to each group is calculated as follows. A cluster head calculates the initial value of $k_{\mathcal{B}_i} = \lfloor n/\beta \rfloor$ for ith group. Then, from the group that has the highest correlation, $k_{\mathcal{B}_i}$ is increased by one until the $n\%\beta$ groups so that the group with high correlation has one more group member.

Now, let us define a specific block design applied in IntraDRC. The block design based node set selection design is defined below.

Definition 5.2.1: Consider $\mathcal{S}_t$ as the tth set that contains k_t distinct nodes. π-set selection is an arrangement of η nodes into π sets from $\mathcal{S}_1$ through $\mathcal{S}_\pi$ such that each node occurs in α_i different sets, and every pair of distinct node i and j occur together in exactly σ_{ij} sets.

In IntraDRC, η corresponds to $\mathcal{N}$ whereas π corresponds to β. The details of these mappings are discussed later. Each set $\mathcal{S}_t$ contains the subset (k_t nodes) of cluster members that are assigned to a particular time slot. In other words, the k_t nodes are potential contenders, that compete with each other to report data at a given time slot. When a pair of nodes i and j belong to σ_{ij} sets, two nodes compete with each other for data reporting σ_{ij} times. When node i belongs to α_i sets, i has α_i chances for data reporting during a data reporting round.

A specific block design form and the different parameter values result in various node sets. When each node gets a data reporting opportunity only once during the

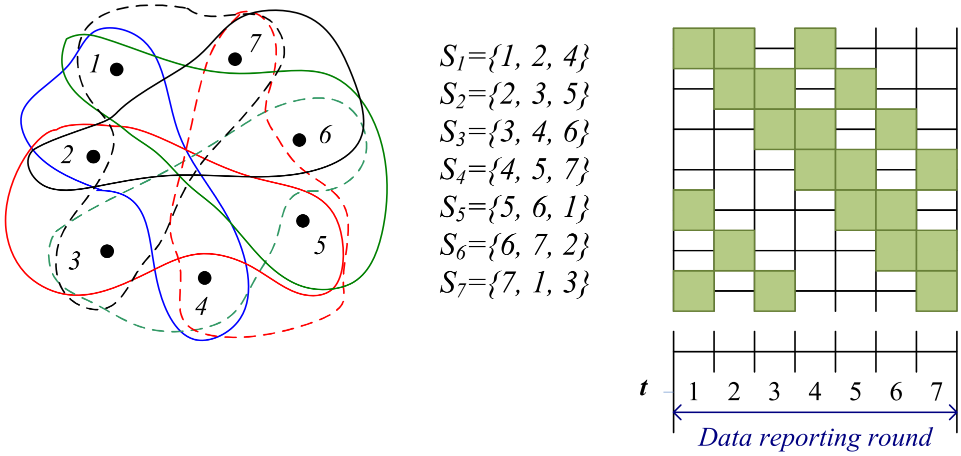


Figure 5.3. $\mathcal{N} = \pi = \eta = 7$, k=3, $\alpha = 3$, $\sigma = 1$.

data reporting round, all α_i, where $1 \leq i \leq \pi$, is set to 1, and there are no redundant nodes in the different sets. In this case, k_t in $\mathcal{S}_t$ is set to either $\lfloor \eta/\pi \rfloor$ or $\lfloor (\eta/\pi)) + 1 \rfloor$. IntraDRC calculates the initial $k_t = \lfloor \eta/\pi \rfloor$ for each $\mathcal{S}_t$ and distributes the remaining $\eta \% \pi$ nodes into each set one by one.

Figure 5.3 shows a cyclic design, which is a special case where $\eta = \pi$. In the figure, $\eta = \pi = 7$, $\forall_{t=1}^{\pi} k_t = k$, $\forall_i^{\pi} = \alpha = 3$, and $\forall^{\sigma_{ij}} = \sigma = 1$. Node 1 has data reporting chances at 1st, 2nd, and 4th time slots. Similarly other six nodes also have reporting chance for at least three slots as $\alpha = 3$. Node 1 belongs to the same set as Node 2 only once, as $\sigma = 1$. In our framework, IntraDRC applies the cyclic block design for the initial node set configuration to guarantee that all nodes are evenly distributed in the π sets. The design uses $(\eta, \pi, \alpha, k, \sigma)$ parameters (all α, k, and σ have the same values respectively) and $\eta = \pi = \beta = \mathcal{N}$. Depending on the channel access mechanism used in a system, the node will compete with other two nodes at each slot using a contention-based mechanism or a cluster head will assign this node at one of three slots.

As mentioned earlier, η corresponds to $\mathcal{N}$ in IntraDRC. While applying the cyclic block design, however, it is possible that the block design parameters may not properly follow the elementary relations in block designs in that not all parameters may generate integer values. The parameter relations as follows.

$$k = \frac{(\eta \cdot \alpha)}{\pi}. \tag{5.7}$$

$$\alpha(k-1) = \sigma(\eta - 1). \tag{5.8}$$

To solve this problem, IntraDRC omits $\mathcal{N} - (k^2 + k + 1)$ nodes. These omitted nodes get into a sleep mode and does not participate in sensing tasks for the predefined time, i.e., one data reporting round. In other words, IntraDRC manages $(k^2 + k + 1)$ active nodes in each cluster such that

$$\mathcal{N} = k^2 + k + 1. \tag{5.9}$$

Table 5.3 shows the example.

Table 5.3. Relationship Between $\mathcal{N}$ and k

$\mathcal{N}$	k
3	1
7	2
13	3
21	4
31	5
43	6

To omit particular nodes, each cluster head analyzes the sensing results of cluster members and maintains the data correlation between nodes [8]. Then, it omits the nodes that report highly correlated sensing data with other nodes in advance. When the usage of correlation is not feasible, i.e., before analyzing the correlation between nodes, IntraDRC randomly selects the nodes to omit.

5.2.3 Data Report Scheduling

The IntraDRC scheme supports the contention-based channel access mode using CSMA/CA, the contention-free mode using TDMA, and the hybrid mode, which combines both contention-based and contention-free modes in one data reporting round. The access mode may be pre-configured by the end system. The default operation of IntraDRC is based on the contention-based mode. In Figure 5.3, with the contention-based mode node 1 will compete with nodes 2 and 4 at slot 1, nodes 5 and 6 at slot 5, and nodes 3 and 7 at slot 7 to transmit data if the nodes have data to report. As IntraDRC reduces the potential contenders by grouping nodes into $\mathcal{S}_g$, it can use a smaller back-off delay than traditional CSMA/CA. On the other hand, with the schedule-based mode IntraDRC decides when node 1 transmits its data among the slot-set $\{1, 2, 4\}$.

Although IntraDRC operates with the contention-based mode, a data reporting node can send a contention-free slot access request based on the status of its queue. If node i has data either that cannot be transmitted in the previous data reporting round(s) or that has delay constraint, it sends the slot request message to a cluster head. A cluster head gives priority to delay constraint data to reserve a time slot.

When the operation is based on a schedule-based mode, a cluster head considers the residual energy of a node the age of data to schedule transmission. The aging parameter is defined to represent fairness in terms of the frequency of data reports. If a node has a chance to transmit data, it sets the age to zero. If it did not get a chance to transmit, it increases the value by 1 whenever a data report cycle is passed.

Then, a cluster head gives the higher priority for transmission to the node with older age.

A cluster head sets a threshold $\mathcal{E}^{ths}$ for the residual energy $\mathcal{E}_i^{res}$ of its member i as proposed in [64]. When $\mathcal{E}_i^{res} \leq \mathcal{E}^{ths}$, the aging value is increased by 0.5 so that the delivery chance is reduced by half.

A cluster head s estimates the mean number of packets and the mean residence time, and it sends the request message when the values reach a certain threshold. Following describes the estimation. λ_s represents the average arrival rate of packets generated to a queue and μ_s represents the service rate at s. We assume that λ_s follows Poisson distribution, μ_s follows an exponential distribution, and $\lambda_s < \mu_s$. Let us define p_n^s is the probability that n packets are in a queue at s. We obtain $p_0^s = 1 - \rho_s$ and $p_n^s = (1 - \rho_s)(\rho_s)^n$, where $\rho_s = \frac{\lambda_s}{\mu_s}$. Then, the residence time at node i is calculated as

$$D_s^{res} = D_s^{srv} + D_s^{que}, \tag{5.10}$$

where $D_s^{srv} = \frac{\rho_s}{\lambda_s}$, and $D_s^{que} = \frac{\rho_s D_s^{srv}}{1-\rho_s}$.

As illustrated in Figure 5.4, D_i^{res} sharply increases after a certain point when λ_s increases. (λ_s is presented as *lambda* in Figure 5.4.) Therefore, a cluster head regulates the data reporting rate from the reporting nodes as it may need to manage the traffic from other clusters as well. This regulation is done by updating the report scheduling matrix configuration. We describes the scheduling matrix as follows.

A cluster head maintains a matrix to represent the data reporting schedule for its members. The matrix is defined as $\mathcal{A} = [a_{it}]$, where $i = 1, \ldots, \eta$ and $t = 1, \ldots, \pi$. Whether node i is assigned to the tth time slot depends on the following:

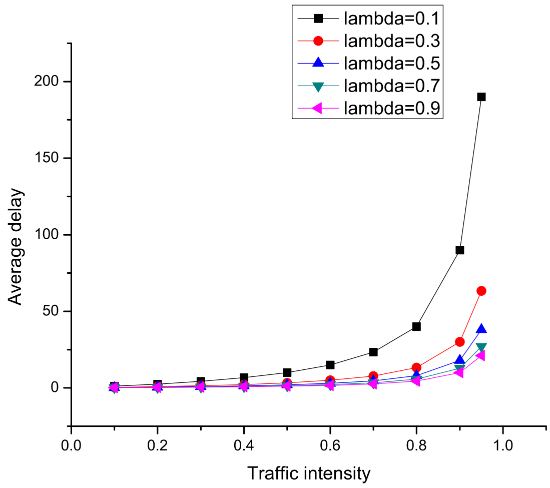


Figure 5.4. Relationship between the Residence Time and Traffic Intensity.

$$a_{it} = 1, \text{ if } i \in \mathcal{S}_t,$$

$$a_{it} = 0, \text{ if } i \notin \mathcal{S}_t.$$

Each cluster head broadcasts the schedule matrix $\mathcal{A}$ after probing its members during the configuration state. Following is the schedule matrix $\mathcal{A}$ of Figure 5.3.

$$\mathcal{A} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.11)$$

Using this matrix, a cluster head can estimate k_t, which is the number of potential contenders for slot t, and α_i, which is the frequency of data reporting chances for node i, using the following equations, respectively.

$$k_t = \sum_{i=1}^{\eta} a_{it} \tag{5.12}$$

$$\alpha_i = \sum_{t=1}^{\pi} a_{it} \tag{5.13}$$

Note that IntraDRC uses the same values for each of k, α, and σ parameters in the initial cyclic block design as mentioned earlier. This initial design matrix will be updated if a cluster head assigns a reserved time slot to a particular data reporting node after receiving the request message.

The overlap matrix $\mathcal{L} = [\sigma_{lm}] = \mathcal{A}\mathcal{A}'$ with size $\mathcal{N} \times \mathcal{N}$, where $\sigma_{lm} = \sum_{u=1}^{\eta} \alpha_{ul}\alpha_{um}$ ($\mathcal{A}[a_{it}]$ with size $\mathcal{N} \times \beta$ and $\mathcal{A}'[a_{ti}]$ with size $\beta \times \mathcal{N}$) signifies the number of occurrences of nodes l and m being assigned to the same time slot.

$$\mathcal{L} = \begin{pmatrix} \alpha_1 & \sigma_{12} & \ldots & \sigma_{1\eta} \\ \sigma_{21} & \ldots & \ldots & \sigma_{2\eta} \\ \ldots & \sigma_{ij} & \ldots & \ldots \\ \sigma_{\eta 1} & \ldots & \ldots & \alpha_{\eta} \end{pmatrix} \tag{5.14}$$

Using the matrix $\mathcal{L}$, it is shown that $\sigma_{ll} = \alpha_l$, which denotes the number of channel access chances of node l. Note that $\sigma_{lm} = \sigma_{ml}$ as both denote the frequency with which the two nodes l and m are assigned for the same slot. By controlling the number of reporting nodes, a cluster head can estimate the expected level of collisions.

5.3 Two-Phase Node Scheduling

A two-phase node scheduling (TNS) scheme is a QoS-aware distributed data report scheduling scheme for heterogeneous networks with multi class sensing tasks. Referring to Chapter 3.2, note that this protocol is not limited to heterogeneous scenarios. Sensor nodes monitor the targeted environmental phenomena periodically and deliver the result to a sink, and each node maintains a neighbor table considering up to two-hop neighbors. Adjacent neighbors of a node s_i are defined as $V_i = \{j \mid d(i,j) \leq r_i\}$, where r_i is the communication range of i and $d(i,j)$ is the Euclidean distance between s_i and s_j. The TNS scheme includes class-based data reporting round allocation and QoS-aware node allocation algorithms, and the scheme is not limited to contention-based and schedule-based channel access mechanisms; however, sensor nodes transmit and receive data only at the time slot(s) assigned and sleep at the other slots even in a contention-based mode.

As mentioned earlier, at the initial network setup stage a network is self-configured and a node s_i learns the information of its neighbors up to two-hops away, $\mathcal{C}_{s_i}$, and the corresponding Φ_* of one-hop neighbors. Nodes execute a QoS-aware routing algorithm and construct a routing tree. The routing algorithm is responsible for establishing connectivity among nodes while creating paths from a source to the base station in order to satisfy QoS requirements, such as end-to-end delay and the number of hops to a base station. After this step, each node knows its parent to report and its children to receive data. Any global parameter such as QoS parameters Φ^E, Φ^D, and Φ^T and class information, can be pre-stored in a node before the deployment or flooded to the nodes in a network.

A data report stage consists of two states: the configuration state and the steady state. During the configuration state, a cluster head performs the IntraDRC strategy including the TNS scheme operating as a membership manager, data reporting scheduler, and traffic controller. If a new node is added or an existing node is deleted, the information is updated during the configuration stage. If a cluster head is informed from a cluster member about the overlapped schedule interfering nodes belonging to an adjacent clusters, it runs a conflict-resolution algorithm (e.g., exchanging schedule with another member). Figure 5.5 illustrates the result after performing the TNS

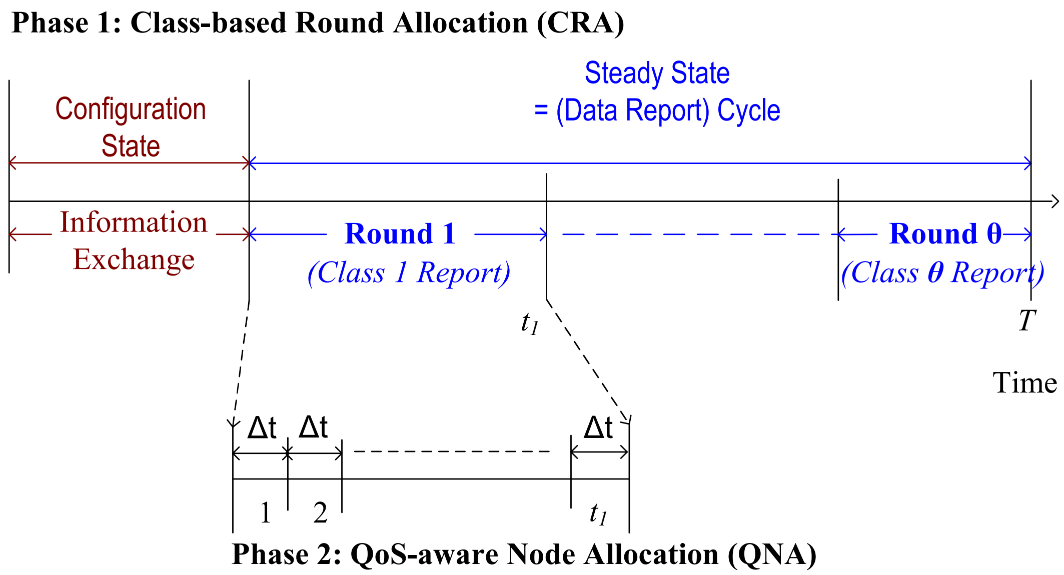


Figure 5.5. Two-Phase Node Scheduling (TNS).

scheme and the details are presented in the following two Sections. During steady state, a node transmits and receives data at the scheduled time slots; otherwise, it goes to sleep mode.

5.3.1 Phase 1: Class-Based Round Allocation

The main objective of this phase is to assign the well-proportioned time slot sets to each class to provide differentiated QoS.

Definition 5.3.1.1: Let $\mathcal{T}$ be the total number of time slots in a data report cycle and t_i be the number of slots assigned to the class i. Each time slot has a length of Δt which is defined as a duration to transmit one single packet.

Definition 5.3.1.2: Let ω_1 and ω_2 be the predefined weighting values to address flexible credits between the QoS parameter and the number of nodes in a particular class. The sum of ω_1 and ω_2 is equal to 1.

Definition 5.3.1.3: The set of time slots t_i assigned to the class i is termed as a *round* and denoted by $\mathcal{R}_i$. A *data report cycle* includes $\mathcal{R}_1$ through $\mathcal{R}_\theta$ defining the time period for every class to have opportunity to report data in turn.

Two cases are considered to decide the value of $\mathcal{T}$. In the case that the end system is required to collect data from every node in the network,

$$\mathcal{T} = \sum_{i=1}^{\mathcal{N}} ts_i, \tag{5.15}$$

where ts_i is the number of required time slot for s_i to transmit data packet(s). In the other case that data collection from a subset of nodes is demanded by an application,

$$\mathcal{T} = \lceil \sum_{j=1}^{\theta} \sum_{i=1}^{n_j} ts_i \dot{\gamma_j} \rceil, \tag{5.16}$$

where γ_j is the required data collection coverage for the class j. A cluster head executes the class-based round allocation (CRA) algorithm to distribute the total number of time slots $\mathcal{T}$ to each different priority class i using the preassigned report precedence value, Φ_i, and the number of nodes in the class, $\mathcal{N}_i$. As a result, a high priority packet has less probability of colliding than low priority ones and is transmitted in advance. A node will compete with the others only in the same group and goes to sleep mode during the time slots with no communication (transmit or receive) schedules and during the rounds assigned for other groups to save energy.

The CRA algorithm includes two rules; the first rule of CRA is

$$t_i = \lfloor (\omega_1 \Phi_i + \omega_2 \frac{\mathcal{N}_i}{\mathcal{N}}) \cdot \mathcal{T} \rfloor. \tag{5.17}$$

After applying Eq.(5.17), the number of remaining slots which have not been assigned to any class yet ranges from 0 through $(\theta-1)$. When the number of remaining slots is not zero, the second rule is applied and the slots are assigned to each class one by one in order of priority. Figure 5.6 describes the CRA algorithm.

```
Algorithm CRA (Φ_i, N_i, ω_1, ω_2) Begin
    NumSlotLeft ← T;
    for i ← 1 to the number of groups;
      t[i] ← ⌊(ω_1 · Φ[i] + ω_2 · N[i]/N) · T⌋;        /* First rule */
     NumSlotLeft ← NumSlotLeft − t[i];
    for i ← 1 to NumSlotLeft;
      Increase t[i] by 1;                       /* Second rule */
    return t[i];        /* Number of slots assigned to each group */
End-Algorithm
```

Figure 5.6. Class-based Round Allocation (CRA) Algorithm.

Table 5.4 shows the example result for the number of slots assigned for data report rounds after running the CRA algorithm. Two classes are defined, where Φ^T are 0.6 and 0.8, and 0.4 and 0.2 for Class 1 and Class 2, respectively, and the total number of time slots in a cycle is 8, 16, and 32. The total number of nodes managed by a cluster head is 8, 16, and 32, where the number of nodes in Class 1 is fixed as 2. The values of ω_1 and ω_2 are the same as 0.5.

Table 5.4. Numerical Results of CRA algorithm

		Φ_1=.6	Φ_2=.4	Φ_1=.8	Φ_2=.2
$\mathcal{N} = 8$	$\mathcal{T} = 8$	4	4	5	3
$\mathcal{N}_1 = 2$	$\mathcal{T} = 16$	7	9	9	7
$\mathcal{N}_2 = 6$	$\mathcal{T} = 32$	14	18	17	15
$\mathcal{N} = 16$	$\mathcal{T} = 8$	3	5	4	4
$\mathcal{N}_1 = 2$	$\mathcal{T} = 16$	6	10	8	8
$\mathcal{N}_2 = 14$	$\mathcal{T} = 32$	15	20	15	17
$\mathcal{N} = 32$	$\mathcal{T} = 8$	3	5	4	4
$\mathcal{N}_1 = 2$	$\mathcal{T} = 16$	6	10	7	9
$\mathcal{N}_2 = 32$	$\mathcal{T} = 32$	11	21	14	18

5.3.2 Phase 2: QoS-Aware Node Allocation

The main objective of SAG is to provide an adaptive QoS to the class with the given number of time slots in $\mathcal{R}_i$. As the second phase is performed for the nodes in the same class, we simplify the notation n_i to n and t_i to t.

5.3.2.1 Grading

Grading was first proposed for a switch design in telecommunication networks [11]. The idea is that not all of the inputs get through to the same number of outgoing circuits and the outgoing circuits could be made available to a number of possible permutations, called gradings, at a particular time. Applying the idea of grading for scheduling data report in wireless sensor networks, an incoming circuit is matched to a set of one or more sensor nodes and an outgoing circuit is a particular time slot.

Figure 5.7 illustrates grading chart examples, where $\mathcal{N}$ = 4, t = 4, and ts =1, and presents how each chart is used for a particular slot allocation mechanism based on contention-free, contention-based, and hybrid medium access control mechanisms.

Definition 5.3.2.1: Let α be the availability of s_i, where $1 \leq \alpha \leq t$ (t replaces the notation t_i which is defined as the number of slots assigned for the class i at the first-phase). Availability is defined as the number of time slots assigned to s_i for data report in a round.

The grading chart in Figure 5.7 (a) presents the configuration of a contention-free mechanism (e.g., TDMA); each time slot is dedicated to a particular node and, therefore, $\alpha = 1$ per every four slots. By contrast, the grading chart in Figure 5.7 (b) illustrates the configuration of a contention-based mechanism (e.g., CSMA); a slot

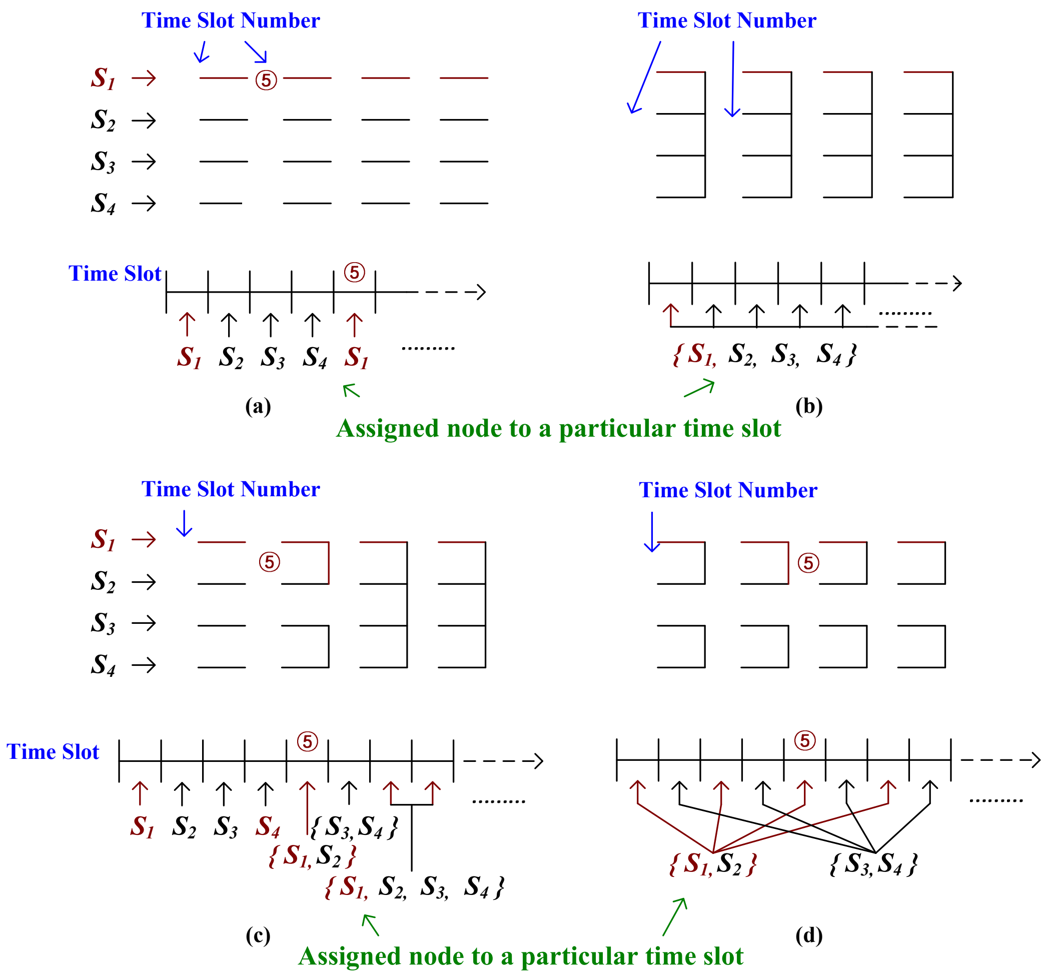


Figure 5.7. Grading Chart (a) TDMA (b) CSMA (c) Gradient (d) Balanced.

can be access by any of the contenders in the class. Finally, Figure 5.7 (c) and (d) show example configurations of a hybrid mechanism.

As mentioned earlier, two cases are considered depending on the task-specific data collection coverage, where $\gamma_i = 100\%$ and $\gamma_i < 100\%$. In the first case, a schedule-based slot allocation mechanism is used. In the second case, as the number of the given time slots t is less than n, both schedule-based and contention-based slot allocation mechanisms are proposed and either of one can be used depending on the application requirement. The background to apply grading in the slot allocation

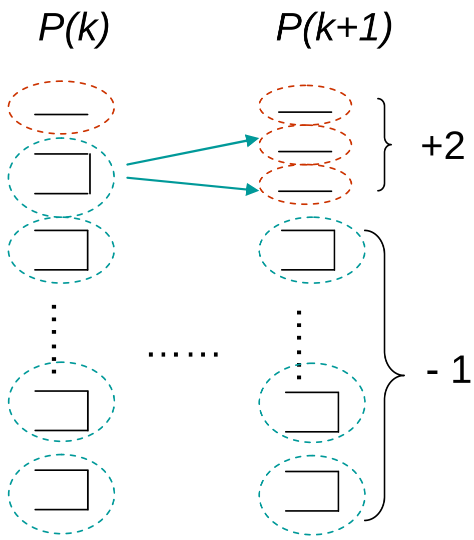


Figure 5.8. Grading Patterns.

algorithm is for managing the number of nodes to report at each time slot with a certain pattern so that a slot is accessed by preassigned sensors, especially, in the case when $\gamma_i < 100\%$ and $t < n$. As sensing results from the sensor nodes in a local area have the high probability to have spatial and temporal correlation, grouping the nodes with correlated data and reporting data among the nodes in a group facilitate efficient data report operation. s_1 and s_2, and s_3 and s_4 in Figure 5.7.(d), for example, are grouped together and either node of the groups transmits data at the given time slot; s_1 transmits at the first slot and s_4 transmits at the second slot. If the nodes do not measure the correlation or applying the correlation is not possible for grading, the nodes are randomly grouped.

A grading chart is described by its incidence matrix such as the description of a block design as presented in [41]. Now let us recall a block design in Combinatorial theory with n distinct objects into t blocks such that the jth block contains the k_j number of distinct objects, where $1 \leq j \leq t$, and each object occurs in α different blocks. Applying the block design concept into a grading chart, the terminology block

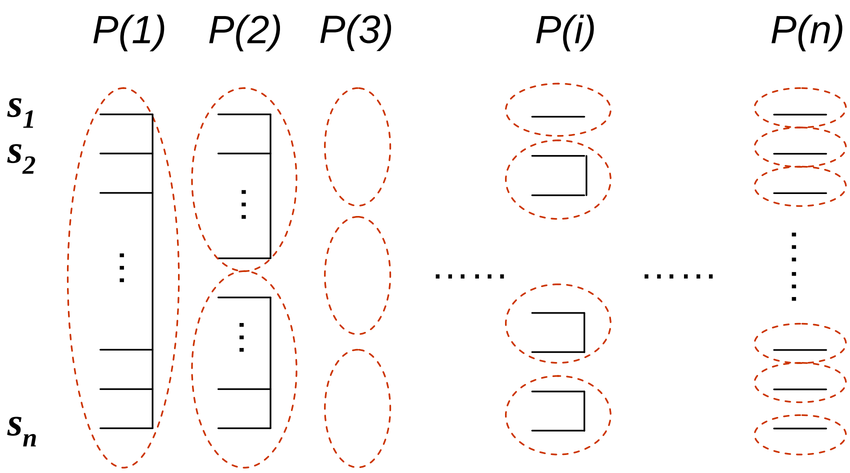


Figure 5.9. Example Grading Pattern.

maps to a time slot t, an object to a sensor node s_i, the number of objects contained in a particular block to the number of nodes grouped for the jth time slot k_j, and the number of occurrence of an object in blocks to an availability α. The matrix to represent a grading chart is defined as $\mathcal{A} = (a_{ij})$, where $i = 1, \ldots, n$ and $j = 1, \ldots, t$.

Using two matrix $\mathcal{A}[a_{ij}]$ and $\mathcal{L}$ discussed in Block design, a cluster head manages the potential contenders at a given time slot.

We define two types of grading patterns, Gradient Grading Pattern (GGP) and partially Balanced Grading Pattern (pBGP), and focus on pBGP in this dissertation. GGP is the pattern with the gradually increased number of contenders as shown in Figure 5.9.(c) with k increased from 1 to 2 and 4. Using this type of patterns, the expected probability of collisions is increasing gradually as well. This pattern is useful when the sampling rate is not very often. pBGP is the pattern with the balanced number of contenders as presented in Figure 5.9.(d) with the equal value of $k = 2$. The term *partially* is used because the number of contenders n may not be exactly divided by the given number of slots t. The details of pBGP is presented in next section with the optimized slot allocation algorithm with two conditions.

5.3.2.2 Channel Access Mechanism

Medium access control(MAC) protocols, which specify how sensor nodes share the communication channel, have been considered as an important area to decide the performance of sensor nodes and further network lifetime of WSNs while devoting to development of energy efficient mechanism including sleep/wake-up mode and adaptive listening to reduce idle listening time, control channel approaches, etc. Our integrated framework focuses on a channel access algorithm which can be used with other components of existing MAC protocols. Channel access schemes for WSNs can be classified into contention-free and contention-based protocols. In contention-free protocols, since there is no collision, which occurs when more than one sensor node in overlapped transmission range try to communicate via shared media, sensor nodes can reduce energy consumption caused by transmission failures and retransmissions and further provide increased accuracy of decision making from collected data. In addition, it is easy to provide fairness for each sensor to send its data with contention-free protocols. In contention-based protocols, the probability to waste idle slot dedicated to a particular node will be less than TDMA based mechanism and reduce the transmission delay and provide more flexible and efficient resource share in irregular event occurrence and frequent topology changes. However, energy consumption is greater than contention-free protocols since a sensor wastes energy for failed transmission caused by collisions and sometimes it is required to spend more energy for retransmission. Our design of the MAG scheme is based on hybrid mechanism managing contention-free and contention-based portion depending on different types of application. The main part of this work is to find optimized grading pattern to handle the tradeoff between energy efficiency and delay to support application-specific sensor operation to collect sensing data.

Both schedule-based and contention-based channel access mechanisms are considered for data report scheduling.

Schedule-based mechanism: When the amount of data traffic is heavy, e.g., event-driven scenario like fire detection, or collision-free data report is required, a cluster head executes a schedule-based mode. If one single node is assigned for a particular time slot, the node is scheduled for the slot; otherwise, a cluster head makes a decision which one among the nodes assigned for a particular slot will transmit its data. The node selection can be initiated by either a source node or a cluster head. If a node is alerted by the limited queue size or event detection, the node send a slot request message to the cluster head. When the cluster head can allow the required time slot(s) to the sensor, it sends the schedule; otherwise, it sends no extra message. A cluster head performs a node selection based on the information of the nodes assigned, such as energy level, a slot request, or an aging value. An aging value is increased whenever a data report cycle is passed and initialized to zero whenever the node reports data.

Contention-based mechanism: When the amount of data traffic is low or contention-based channel access is assigned, a cluster head does not need to perform the further node selection step after nodes are assigned to the given number of time slots using gradings. In contention-based mode, nodes use CSMA/CA to transmit data. As the number of potential contenders for the jth slot is decreased from $\mathcal{N}_i$ to k_j when more than one time slot is assigned to a class, the maximum random backoff value in CSMA/CA is decreased depending on the k_j that reduces the end-to-end delay.

5.4 Simulation Experiments

The performance of the QRS scheme is evaluated using a discrete time event simulator NS-2 [2, 1, 38]. The simulation results are based on randomly deployed static wireless nodes in a squared area and the simulation parameters are shown in Table 5.5. Various network density is simulated with the nodes of 100, 200, and 300 in the area of 300 x 300 m. The simulations have been performed based on a contention based mode. The required data collection coverage γ is 0.6 and both weighting values ω_1 and ω_2 are 0.5 for calculating the total number of time slots in a cycle. A cluster head is randomly elected regardless the type of classes and the role is not exchanged with other nodes during simulations. After the configuration state, a cluster head acts like a normal node to transmit data; in other words, a cluster head is the same with others except that it executes a two-phase slot allocation algorithm.

Table 5.5. Simulation Parameters

Simulation parameter	Value
The size of an area	300 m X 300 m
Bandwidth	19 Kbps
Transmission range	60 m
Transmit mode power	60 mW
Receive mode power	30 mW
Idle mode power	30 mW
Sleep mode power	0.003 mW
Transition power	30 mW
Transition time	2.45 ms
Packet size	96 bytes
Time slot size	42 ms
Simulation time	1000 seconds

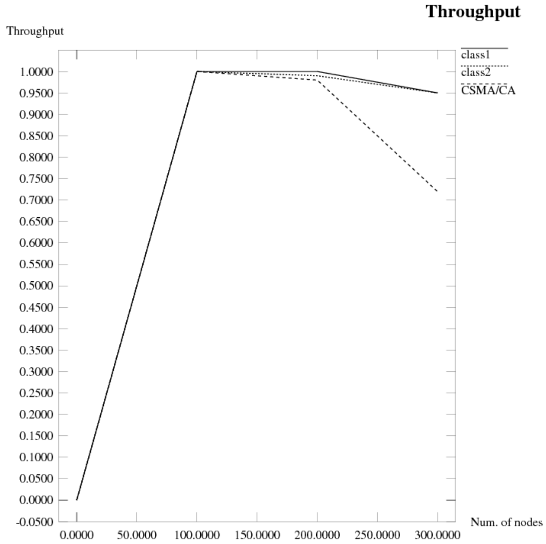


Figure 5.10. Throughput when $\Phi_1 = \Phi_2 = 0.5$.

5.4.1 Throughput

Throughput is evaluated for different priority classes by (the total number of bits received at a base station) / (the total number of bits transmitted at each source node) with various node density. From the results, the IntraDRC strategy shows better performance in terms of throughput in high density networks.

In Figure 5.10, the IntraDRC strategy is compared with CSMA/CA. As the QoS parameter is the same for each class as 0.5, every node is treated equally and the number of potential contenders at each time slot is also partially balanced for each class using the *pB-SAG* algorithm in the IntraDRC strategy. When the number of nodes in the network is 100 or 200, the throughput of the QRS scheme is almost 100 percent. CSMA/CA also shows the good performance as the network does not suffer from collisions. When the network density is increased with the nodes equal to 300, however, the throughput of CSMA/CA drops dramatically to around 72 percent. On the other hand, the throughput of the IntraDRC strategy is slightly reduced because

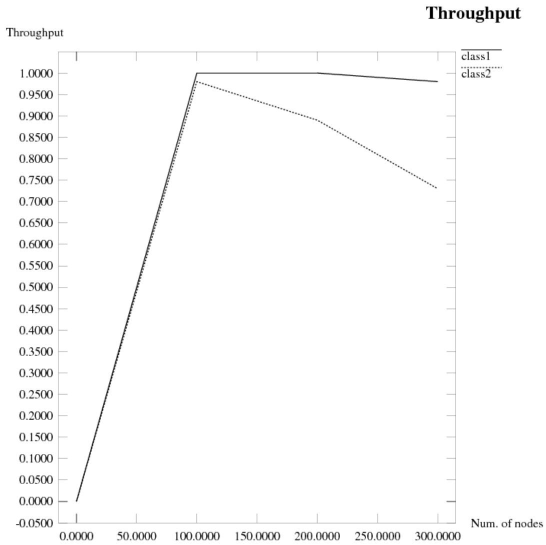


Figure 5.11. Throughput when $\Phi_1 = 0.8$ and $\Phi_2 = 0.2$.

the number of nodes attempting data transmission is distributed into the several time slots assigned to a class.

In Figure 5.11, different QoS parameter values are set to each class. As the number of time slots assigned to class 2 is decreased to 0.2, the throughput of class 2 keeps decreasing to around 89 percent and 73 percent as the number of nodes is increased to 200 and 300. On the other hand, the throughput of class 1 shows the better results in comparison to Figure 5.10 when the number of node is 300, as the number of time slots for the class is increased with the high QoS parameter value and the potential contenders are separated into the given slots.

5.4.2 Energy Consumption

The IntraDRC strategy consumes energy for data report scheduling. At the initial network setup stage, the cluster head election and cluster construction require energy consumptions for nodes to communicate with other adjacent nodes. Then, a

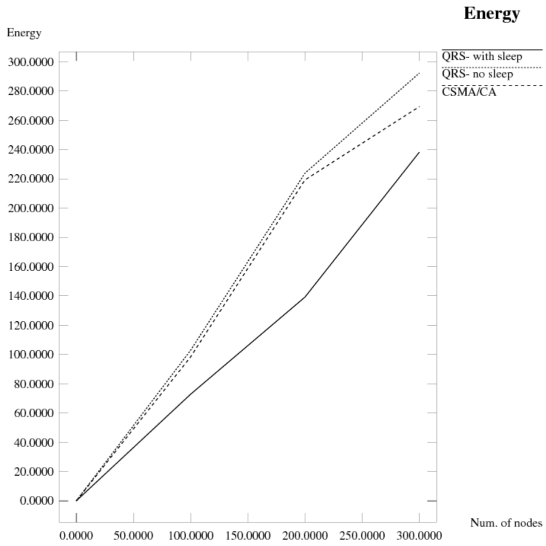


Figure 5.12. Energy Consumption.

cluster head consumes energy to manage the information of its members, to construct data report schedules, and to distribute the schedule information to its members.

Figure 5.12 shows the average energy consumption of cluster heads in a network. The value is calculated by $(\sum_{i=1}^{s} \varepsilon_i)/s$, where ε_i is consumed power by a cluster head and s is the total number of cluster heads in a network. As the time slot scheduling is performed once at the configuration state and the simulations remains at the steady state, the energy consumption for a node acting as a cluster head may not show a big difference with other normal nodes.

As each sensor node in the TNS scheme exchanges periodic messages with a cluster head (one per 10 seconds) for the membership management purpose as well as scheduling information, the simulation result shows that a node using the TNS scheme without sleep mode consumes more energy compared with the result of CSMA/CA; around 5 W with 100 nodes, 20 W with 200 nodes, and 25 W with 300 nodes. Using sleep mode, the TNS scheme shows good performance reducing the energy consumption for idle listening.

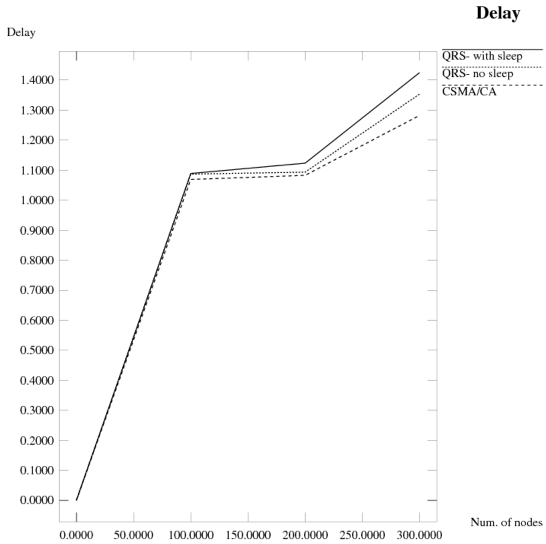


Figure 5.13. End-to-end delay.

5.4.3 End-to-End Delay

The performance of end-to-end delay is measured with one node deployed at the left most bottom part of the network in three scenarios; the TNS scheme with sleep mode, the TNS scheme without sleep mode, and CSMA/CA. The average end-to-end delay is defined as $(\sum_{i=1}^{pkt} \mathcal{D}^i{}_{rcv} - \mathcal{D}^i{}_{snd})/(pkt)$, where $\mathcal{D}^i{}_{rcv}$ is the arrival time for the packet i at the destination, $\mathcal{D}^i{}_{snd}$ is the packet transmission time at a source, and pkt is the number of packets generated by a source. Figure 5.13 shows that when the network density is not high (with the number of nodes equal to 100), the differences among three scenarios are not significant with the maximum delay being less than 20 ms. In the results, we find that the effect of sleep mode in the TNS scheme results in a delay difference of 2 ms. The performance of the TNS scheme, however, in terms of end-to-end delay is affected as the network density is high with 300 nodes. The results show that the delay of CSMA/CA is less than that of the TNS scheme and the TNS scheme without sleep mode is less than the one with sleep mode. This result, however, does not distinguish the data delivery between different classes.

5.5 Discussion

The TNS scheme is a novel QoS-aware data report scheduling scheme performed including a two-phase slot allocation protocol in a cluster designed for heterogeneous wireless sensor networks. The ultimate goal of this scheme is to provide adaptive QoS with the given resources depending the different types of tasks. The definition of quality of service (QoS) and the metrics to evaluate the performance of a wireless sensor network are different from traditional networks in that the QoS attributes highly depend on the specific sensing tasks and applications. While energy efficiency is an important consideration for designing algorithms and protocols for wireless sensor networks, other QoS parameters such as the coverage rate, the end-to-end delay, fairness, throughput, and error rates for delivery or sensing may be equally important depending on the application objectives. Thus, an important issue in a sensor network is to design task-specific QoS-aware data reporting algorithms and protocols that optimize resource consumption and extend the network lifetime. In this dissertation, we propose an integrated framework for QoS-aware data reporting in wireless sensor networks. By classifying prioritized sensing tasks based on the task-specific QoS parameters and the levels, high priority data has the higher probability of accessing the channel and the higher probability of successful transmission in a contention-based mode compared to low priority data.

The IntraDRC strategy is based on the selection of data reporting nodes that applies the block design concept from combinatorial theory and a novel two-phase node scheduling (TNS) scheme that defines class-based data reporting rounds and node assignment for each time slot. The objective of IntraDRC is to provide optimized data reporting control in a distributed manner. In this strategy, a certain number of data reporting nodes are selected in each cluster in order to satisfy the throughput fidelity specified by the applications while reducing redundant data reporting by selecting a subset of cluster members. This intra-cluster reporting control eventually helps control the overall amount of traffic in the network. The TNS scheme schedules data reporting while considering the priority of data, yet guaranteeing that sensor nodes compete with each other in the same class only.

Experimental results show that our scheme can achieve better performance of throughput by reducing the probability of collisions as the number of potential contenders are distributed into the time slots assigned. In terms of throughput and energy efficiency, the TNS scheme strategy using sleep mode shows good performance in a dense network; however, end-to-end delay may be extended to low class data report. Currently, we are enhancing the grading pattern scheme block designs to build a more flexible slot allocation protocol.

CHAPTER 6

DATA REPORTING TREE MANAGEMENT

In this chapter, we discuss the inter-cluster data reporting control scheme called QoS-aware data reporting tree construction (QRT) for data reporting control from each cluster head to a sink after the cluster head performed data aggregation after collecting sensing results from β data reporting nodes. The goal of the QRT scheme is to offer reporting paths based on the characteristics of traffics while considering the trade-off between the end-to-end delay and energy efficiency. In order to minimize the end-to-end delay, the fewest hop count is considered for tree construction; otherwise, a load balanced tree is constructed to reduce the energy consumption of the nodes that are close to a sink or the ones that have heavy children. Each cluster head analyzes the traffic load of its cluster for load balancing and congestion control to improve the overall network performance [54, 45, 12]. We modify the spanning tree construction algorithm proposed [13]. While network layer protocols offer a best-effort service, the transport layer protocol is responsible for achieving reliable end-to-end service and congestion/flow control.

6.1 Preliminaries

In the InterDRC strategy, the QRT scheme maintains data reporting paths from each cluster head to a sink based on the traffic characteristics while considering the trade-off between the end-to-end delay and energy efficiency. As a default, QRT constructs a fewest hop spanning tree for minimizing the end-to-end delay and minimum weight spanning tree for distributing energy consumption of a node with heavy

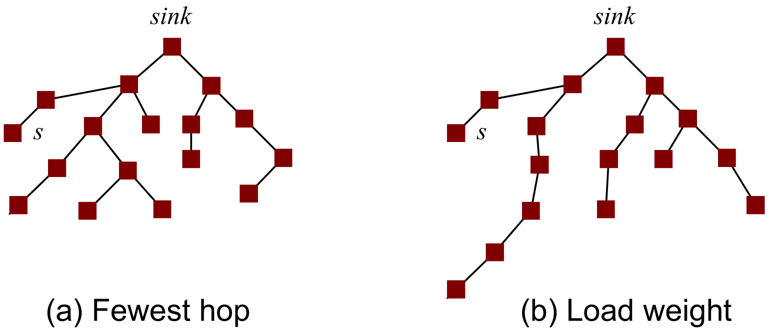


Figure 6.1. Data Reporting Tree.

children as presented in 6.1. A cluster head maintains reporting paths constructed by these spanning trees. In a homogeneous network, each cluster head finds the reporting paths to a sink, exchanges the path information with adjacent cluster heads, and operates as a backbone node for a back-bone based tree construction. Similar to the core-extraction distributed algorithm for a back-bone based routing path [48], one or two gateway nodes are included to connect two cluster heads.

6.1.1 Problem Definition

In the QRT scheme, we consider two QoS parameters. During the network setup time, QRT first establishes a fewest hop-based reporting tree and this tree is used for delay constraint traffic delivery in the InterDRC strategy. This tree is updated with traffic load information to construct a traffic-adaptive reporting tree and this tree is used for energy efficient delivery. In a homogeneous network, each cluster operates as the backbone node for a back-bone based tree construction, using a gateway to exchange data between two adjacent clusters.

In the inter-cluster data reporting strategy, we focus on three QoS metrics: throughput fidelity, end-to-end delay, and the amount of energy consumed. We define Φ as a system parameter whose value is defined by the end system during the QoS provisioning procedure, presented in Section 4.1, to specify the required QoS level. Φ^c_{TH}, Φ_D, and Φ_E are respectively defined to specify the desired throughput for a task group c, the end-to-end delay constraint at the end system, and the residual energy threshold of a node. In this dissertation, we define two task groups. One operates to minimize energy consumption while the other operates to minimize end-to-end delay. Consider $\mathcal{N}$ cluster members exist in a cluster and β nodes among $\mathcal{N}$ are selected as data reporting nodes such that node i generates a packet with the size L_i. The data rate of node i is denoted by R_i. We assume that R_i is equal for all nodes in homogeneous networks, and R_s of a cluster head s is greater than R_i of a member node i in heterogeneous networks.

Throughput

Throughput of a task c at a cluster head s, denoted as TH^c_s, is defined as the average rate of successful transmission in packets/second. This throughput should be at least the desired throughput at the end system, such that $TH^c_s \geq \Phi^c_{TH}$.

End-to-End Delay

The end-to-end delay to report data with delay constraint from a cluster head s to a sink, denoted by D_s, should be less than or equal to the desired delay constraint, such that $D_s \leq \Phi_D$. The detailed equation for D_s is presented in Section 5. The DRC scheme manages data reporting paths so that the data delivery with delay constraints can be performed in h_s hops to satisfy the given delay condition.

Energy consumption

If the residual energy of node i, denoted by E^{res}_i, is less than Φ_E, a node gets into the energy saving mode. The E^{res}_i estimate follows the approach presented in [64].

Operating in this mode, a node only transmits its own data and does not participate as a forwarding node to deliver packets generated from other nodes.

While adopting energy consumption as a QoS parameter, a node only transmits its own data and does not participate as a forwarding node to deliver packets generated from other nodes. In order to provide better data reporting opportunity to prioritizing sensor nodes with by choosing sensor nodes with a certain residual energy, the network can extend the network lifetime.

6.2 Fewest Hop-Based Reporting Tree

For the fewest hop-based reporting tree construction, each cluster head finds the shortest path to a sink and exchanges the information with its adjacent cluster heads. With h_s denoting the number of hops to the root from node s, we have:

$$h_1 = 0, \tag{6.1}$$

$$h_s = \min_{j \in V_s}(h_j) + 1, s \geq 2. \tag{6.2}$$

Node s selects node j as a parent with the minimum number of hops to the root. Figure 6.2 shows a data reporting path from the cluster head s to the destination d in a homogeneous network. In this case, one or two gateway nodes are included in a reporting path to connect two cluster heads and help the data exchanges between them. Figure 6.3 presents a data reporting path in a heterogeneous network.

For the tree construction, the QRT scheme executes Dijkstra's algorithm from a sink to every cluster head during the configuration time. As the complexity of Dijkstra's algorithm is $O(|V|^2)$ and the average time complexity is $O(a \times |V|)$, where

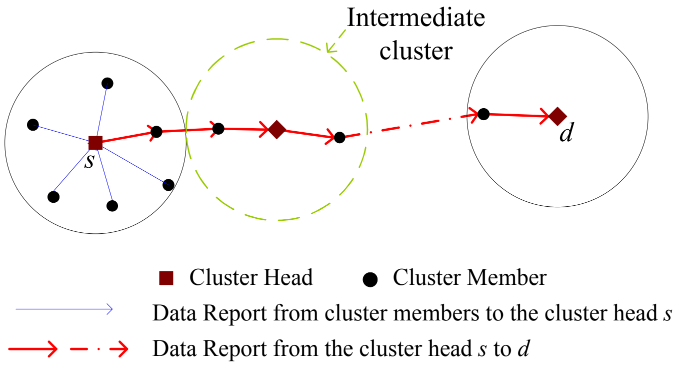


Figure 6.2. Data Reporting in a Homogeneous Network.

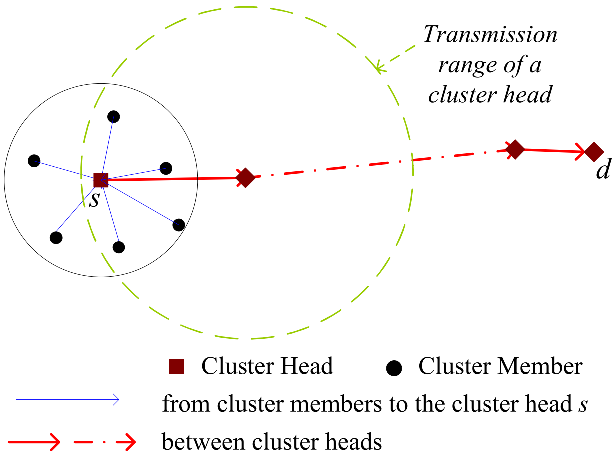


Figure 6.3. Data Reporting in a Heterogeneous Network.

a is the average node degree, the scalability issue should be considered. Our DRC framework runs Dijkstra algorithm once during the initial setup time and updates as the traffic-adaptive tree in a distributed manner. Figure 6.4 presents the initial data reporting tree construction algorithm.

The QRT scheme simplifies the end-to-end delay computation assuming that the residence time, denoted by D^{res}, and the propagation time, denoted by D^{prop}, for one-hop transmission are the same for all nodes. Assuming that λ follows Poisson distribution, μ follows an exponential distribution, and $\lambda < \mu$, where λ represents the average arrival rate of packets generated to a queue and μ_s represents the service

Initial Tree Construction Algorithm (graph $G = V, E$) **Begin**

compute the ϵ;

initialize the tree with a sink;

while a node exists that is not added to the tree **do** ;

  for node i not in the tree **do**

    compute $\varphi_s = 1 - \frac{h_s}{\epsilon}$;

    compute $x_s = \varphi_s * h_s + (1 - \varphi_s) * \lambda_s$;

    store the minimum x_s;

  end

  add the node i that achieves the minimum cost;

End-Algorithm

Figure 6.4. Data Reporting Tree Construction Algorithm.

rate. Let us define p_n is the probability that n packets are in a queue. We obtain $p_0 = 1 - \rho$ and $p_n = (1 - \rho)(\rho)^n$, where $\rho = \frac{\lambda}{\mu}$. Then, the residence time at node i is calculated as

$$D^{res} = D^{srv} + D^{que}, \tag{6.3}$$

where $D^{srv} = \frac{\rho}{\lambda}$, and $D^{que} = \frac{\rho D^{srv}}{1-\rho}$.

D^{agg} is as the time delay for each cluster to perform data fusion and this delay depends on the data fusion procedure. Let $\mathcal{H}_{int}$ be the number of intermediate clusters in the data reporting path between the source cluster head s and the destination sink d. Then, the data reporting time from s to a sink in both scenarios is given by:

$$D_s = D^{prop} + (D^{res} + D^{prop}) * h_s + D^{agg} * \mathcal{H}_{int}. \tag{6.4}$$

The average node degree at each cluster of a fewest hop spanning tree tends to be large while the depth of the tree is shorter. The nodes that have many children in a tree will consume more energy than others and forfeit their functions due to energy depletion causing holes in the network. In the next section, we discuss how the tree is updated considering the trade-off between delay and energy problems.

6.3 Traffic-Adaptive Data Reporting Tree

In the case where the end-to-end delay requirement is not specified, InterDRC manages another traffic-adaptive reporting tree to distribute the energy consumption of the nodes. In other words, a cluster head maintains two data reporting paths. One is the reporting path from the fewest hop-based tree, and the other is that from the traffic-adaptive tree. For the traffic-adaptive data reporting tree construction, we modify the spanning tree construction algorithm in [13]. The algorithm considers a combination of a hop count and a path weight. Each cluster head analyzes the traffic load of its cluster to balance the traffic and to provide congestion control for the overall network performance. In InterDRC, the QRT scheme measures the traffic load at each cluster and use this traffic information as a path weight.

The data reporting tree construction scheme in InterDRC is to provide robustness so that the reporting tress that are immune to data loss when nodes or links fail while maintaining good performance. The basic metrics in the QRT scheme as follows.

$$\varphi \times \{hop - count\} + (1 - \varphi) \times \{path - weight\}, \tag{6.5}$$

where $0 \leq \varphi < 1$. Where more importance is placed on the *hop count*, the shape of data reporting tree is tends to be fat and shallow. On the other hand, more importance is placed on the *path weight*, the shape of tree is skinny and deep. Therefore, the type of data reporting tree depends on the metric of interest. While considering the traffic from other clusters as well as the traffic from its members as

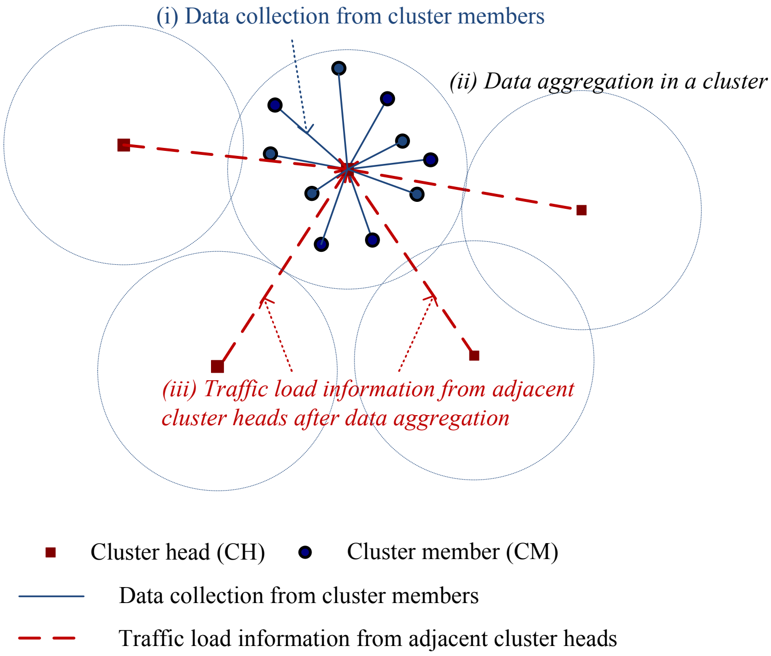


Figure 6.5. Traffic Load Estimation.

illustrated in Figure 6.5, the total traffic load received at a cluster head is measured by

$$\lambda_s = \sum_{j \in M_s} \mu_j (1 - p_0^j) + \lambda_a, \tag{6.6}$$

where M_s is a set of cluster heads that are children of s in a data reporting tree, μ_j is the service rate of node j, p_0^j is the probability of an empty queue of j, and λ_a is the arrival rate after data aggregation at s. In order to balance the trade-off between the delay constraint and the traffic load (related to energy consumption), the cost value to construct the tree is defined as

$$x_s = \varphi_s * h_s + (1 - \varphi_s) * \lambda_s, \tag{6.7}$$

where $\varphi_s = 1 - \frac{h_s}{\epsilon}$ $(0 \leq \varphi_s < 1)$ and ϵ is the depth of the deepest leaf in the tree and h_s is the hop count of a cluster head s. As a result, we can update the initial fewest hop based data reporting tree while considering traffic load. After the initial configuration stage, each cluster head will know

- its parent node in the fewest hop-based reporting tree
- its estimated weights
- the number of hops to a sink.

Then, a node s selects a parent j based on the form of the weighted cost in Eq. 6.7. Note that the number of hops of node i is the number of hops to a sink if node i selects node j as its parent. This method builds a data reporting tree as long as (1) no two nodes choose each other as a parent and (2) no cycles are constructed. In order to solve these possible problems, a simple method by limiting the number of messages exchanged for the tree construction is implemented. In this tree construction scheme, learning the value of ϵ could be expensive in the case that nodes in the network do not have any knowledge. However, our integrated scheme assumes that a network has basic knowledge such as the number of prioritized tasks, the QoS parameters, the node density of a network, and the capability of sensor nodes, i.e., transmission rage and power level.

This traffic-adaptive data reporting tree has a smaller average node degree at a cluster head compared to that in a fewest hop data reporting tree. Let us define $\mathcal{C}_s$ is a set of adjacent cluster heads of cluster head s. s selects its adjacent cluster head w as a parent if

$$w = \min_{u \in \mathcal{C}_s}(x_u). \tag{6.8}$$

A cluster head updates its parent when it receives the less cost information from another cluster head.

6.3.1 Traffic Generation Probability

In order to calculate *dynamic* p, we assume that a network is steady-state which means the arrival rate λ_i is less than service rate μ_i in the queuing system of the network and infinite queuing space is provided. We also assume that processing delay is ignorable. Suppose that s_i generates data packet with a Poisson distribution and transmits with an arbitrary distribution and packets in a system are served in first-come-first-serve (FCFS) manner. Assuming that a node is a single-server system with only one wireless link, we analyze M/G/1 model to derive delay equation. The total time spent in a node is the sum of queuing delay and service time. First we find the average time $\mathcal{W}$ spent in a queue using the Pollaczek-Khinchin mean value formula defined as

$$\mathcal{W} = \frac{\rho\bar{x}(1 + C_b^2)}{2(1-\rho)},$$

where $\bar{C}_b^2 = \frac{\sigma_b^2}{(\bar{x})^2}$. To derive p using the mean service time $\bar{x}$ and the second moment $\bar{x^2}$, Eq. (6.9) is redefined as

$$\mathcal{W} \approx \frac{\lambda\bar{x^2}}{2(1-\rho)}.$$

From Eq. (6.9), the mean values of each delay system has been driven.

$$E[B] = P(B = b) = \sum_{b=0}^{CW} bP(b) = \bar{B} \tag{6.9}$$

$$E[C] = (1-\rho)\bar{K}\sum_{i=0}^{n} i\rho^i = \frac{\bar{K}\cdot\rho}{1-\rho} \tag{6.10}$$

$$E[A] = (1-\rho)p\sum_{i=0}^{n} i(1-p)^i = (1-\rho)\cdot\frac{1-p}{p} \tag{6.11}$$

Then, the average service time $\bar{x}$ and the second moment $\bar{x^2}$ are given as

$$\bar{x} = \bar{B} + \frac{\bar{K}\rho}{1-\rho} + \frac{(1-\rho)(1-p)}{p} \tag{6.12}$$

$$\bar{x^2} = \bar{B}^2 + \frac{\bar{K}\rho}{(1-\rho)^2} + \frac{(1-\rho)(1-p)}{p^2} \tag{6.13}$$

(6.14)

6.4 Simulation Experiments

The performance is evaluated through simulations in C++. The simulation results are based on randomly deployed static wireless nodes in a squared area. The simulation parameters are shown in Table 7.7. The data collection coverage rate χ is 0.8 and the number of reporting nodes β is calcuated by $n \cdot \chi$ where n is the number of cluster members. The fixed 12 cluster heads have been deployed in the monitoring area regardless the network density with the nodes of 100, 200, and 300. The data reporting cycle consists of two rounds. At the first round, β nodes transmit data to the cluster head either in a contention-based mode using CSMA/CA or in a schedule-based mode using TDMA. At the second round, a cluster head transmits the collected sensing results without data aggregation to a sink located in the right top side of the monitoring area in a schedule-based mode using TDMA.

6.4.1 Throughput

Throughput is measured by (the number of bytes received at a base station) / (the total number of bytes transmitted at each source node) with the nodes of 100, 200, and 300 in the fixed area. Throughtput is measured using both contention-based and schedule-based data reporting in a cluster.

In a contention-based mode, the throughput is slightly affected by the network density where the number of nodes is increased to 300 as illustrated in Figure 6.6. In fact, the throughput performance is also affected by the data collection coverage rate. Where the coverage rate is high, more reporting nodes will try to report data

Table 6.1. Simulation Parameters

Simulation parameter	Value
The size of an area	300 m X 300 m
Bandwidth	19 Kbps
Transmission range	60 m
Transmit mode power	60 mW
Receive mode power	30 mW
Idle mode power	30 mW
Sleep mode power	0.003 mW
Transition power	30 mW
Transition time	2.45 ms
Packet size	96 bytes
Time slot size	42 ms
Simulation time	100 seconds

at a given time such that the probability of collisions becomes high. Therefore, the performance of throughput in a contention-based mode will be decreased high with high coverage rate in high d

Where reporting is performed in a schedule-based mode, throughput was 100% regardless network density because the simulation assumes that there is no delivery error except collisions as presented in Figure 6.7. In real environment, however, the performance of throughput may not be perfect with other environmental effects such as unstable wireless channel and signal strength, and unreliable reporting paths.

6.4.2 Energy Consumption

Figure 6.8 shows the average energy consumption of cluster heads using the equation $\sum_{i=1}^{n} \varepsilon_i / n_h$, where n_h is the total number of cluster heads deployed in a network and ε_i is the total energy consumed at cluster head i. n_h is fixed as 12 in the simulations.

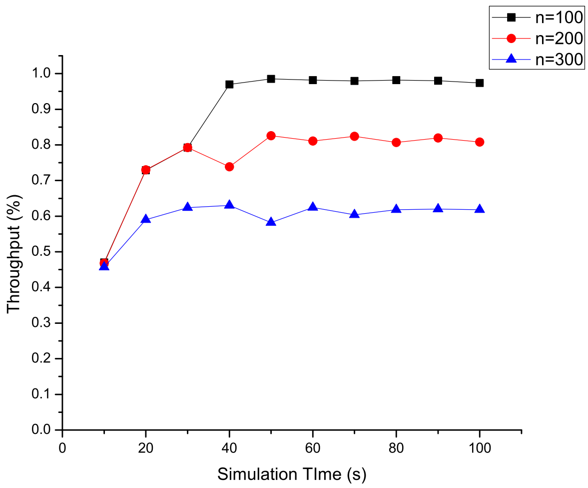


Figure 6.6. Throughput When a Contention-Based Mode.

Figure 6.8 shows the average energy consumption of cluster heads with the nodes of 100 and 300 deployed in the monitoring area. These results are compared with data reporting using a shortest path spanning tree, denoted as ST in a graph, where the edge weight is fixed. We have simulated both contention-based and schedule-based mode for the proposed DRC scheme to compare with each other, but the plots are very similar although there are slightly different values. The reason is that the difference of operation between two modes is only whether a cluster head performs node scheduling to assign a particular node at a given time slot and distributes the information to its members or not. Although the contention-based mechanism does not include the process for scheduling, it does not affect the energy consumption because a cluster head directly starts collecting data without sleep mode. This affects the performance of the end-to-end delay.

Each cluster head needs to learn its member nodes as well as its adjacent clusters and construct a data reporting tree during the network setup time. After that, it also needs to continuously wake up to collect data from its members and forward the

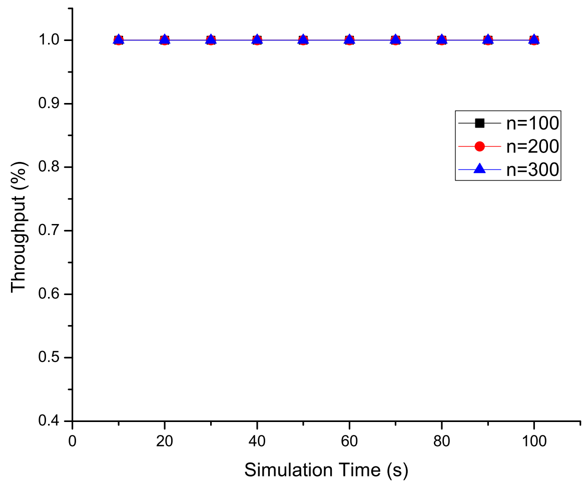


Figure 6.7. Throughput When a Schedule-Based Mode.

results to the base station. Therefore, the time that cluster heads can save energy depends on the inter-cluster reporting schedule, the duty cycle, and the sampling rate specified by the user.

6.4.3 End-to-End Delay

Figure 6.9 illustrates the average end-to-end delay performance by $\sum_{i=1}^{m} \mathcal{D}_i/m$. The average delay of a node is calculated by $\mathcal{D}_i = (\sum_{j=1}^{pkt} \mathcal{D}^j{}_{rcv} - \mathcal{D}^j{}_{snd})/(pkt)$, where $\mathcal{D}^j{}_{rcv}$ is the arrival time for the packet j at the destination, $\mathcal{D}^j{}_{snd}$ is the transmission time at a source, and pkt is the number of packets generated by a node i.

We have compared the results while changing the network density with the nodes of 100 and 300 in both schedule-based and contention-based modes as presented in Figure 6.9. These results had been compared the ones using a shortest path reporting tree. The comparisons did not show big differences because the data reporting from each cluster head to a sink does not affect the performance of the end-to-end delay in the given simulation scenarios. Rather than that, the performance

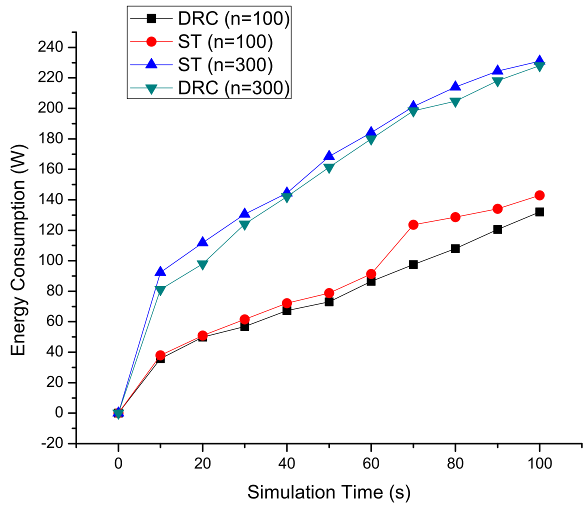


Figure 6.8. Energy Consumption (W) vs. Simulation Time (second).

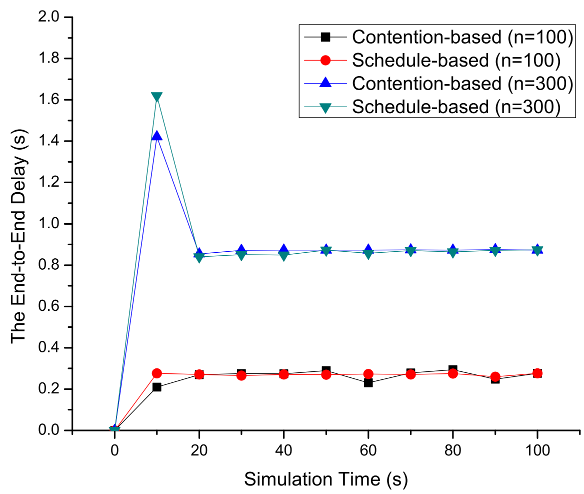


Figure 6.9. End-to-End Delay (second) vs. Simulation Time (second).

was affected by the network density. When the network density is high, the average packet delivery time is increased because the number of required reporting nodes β is also increased and the data collecting time of a cluster head is linearly increased.

6.5 Discussion

In this chapter, we describe a QoS-aware data reporting tree construction scheme, called the QRT scheme, performed for inter-cluster data reporting control. This scheme is based on two QoS parameters: end-to-end delay and energy efficiency. In Intra-cluster data reporting scheme, the traffic amount generated in a cluster is controlled based on the user-specific data collection coverage rate. The scheme selects a certain number of data reporting nodes based on block design and correlation of nodes. The goal of this scheme is to generate at least the desired throughput considering the data delivery error rate while reducing the overall mount of traffic in a network. In the QRT scheme, the data reporting paths are modified from a shortest path spanning tree in order to balance the traffic load to save energy consumption of the nodes with heavy children and eventually extend the network lifetime. The scheme uses the hop count information from a cluster head to a sink and the traffic load information generated within a cluster and forwarded from adjacent clusters that are linked as children in a data reporting tree. The goal of this scheme is to offer reporting paths considering the traffic characteristic. By interacting with the application-specific requirements to acquire the desired information from the monitoring area, the data collection coverage rate, correlation between nodes in a cluster, and the remaining energy level of nodes are considered to balance the trade-off between the end-to-end delay and energy efficient operations.

On-demand multihop routing algorithms such as AODV and TORA eliminate table updates in high-mobility scenarios. However, they cause high-energy cost during route setup phase. The sequential assignment routing (SAR) [33] uses the idea of multiple paths while taking parameters like energy resource, QoS on each path, and the priority of packets into consideration. In SAR protocol, a table-driven multi-path approach is used to improve energy efficiency in a low-mobility sensor network. The failure protection is addressed by having at least k-paths that have no common branches between a node and a sink. Also, localized path restoration procedures are used to decrease energy cost in failure recovery. Each node uses the following parameters to establish routing paths: (1) energy resource estimated by maximum number of packets that can be routed without energy depletion, assuming that the node has exclusive use of the path (2) QoS metrics where higher metric implies lower QoS. In our framework, each cluster head maintains two data reporting paths to support (1) energy efficient and (2) delay constraint data reporting requests. Also, the data reporting path construction is based on the local information exchanged with adjacent cluster heads.

The QRT scheme requires a global information about the depth of the initial spanning tree. This might degrade the performance of the QRT scheme, but in the network model of our proposed DRC framework, the end system (or a sink) will learn the basic network information such as the network density during the initial setup time. Therefore, the QRT scheme may use the estimate value, rather than the exact tree size, in order to first construct initial data reporting tree and update while exchanging messages during the configuration state.

We intend to further investigate the practicality of the QRT scheme in two directions: improving the adaptability to a wide variety of task-specific classes and enhancing the throughput fidelity level with more task classes.

Considering that an event occurrence in a monitored area is a spatial stochastic process, ti is better to have a more flexible data reporting control mechanism depending on the location of events occurred so as to optimize the use of the limited resources, especially in case that the periodic reporting is not or intermittently required. In light of this, the proposed cluster-based integrated framework needs to be extended to cover the monitored area with more granulated local information for each specific cluster. Such granularity can be decided with the localization information of sensor nodes or information exchanges between adjacent cluster heads. More energy-efficient and flexible to the user requirements will facilitate the optimized task-specific QoS supports.

After the node deployment, sensor nodes fulfill a given monitoring task periodically. They may be required not only to periodically report what they sensed with delay constraint requirement, but also to immediately report when they detect a specific event. It is very likely that due to high node density, the number of sensor nodes detecting an event is usually larger than the number of sensors actually required for the event detection.

Although this scheme is based on a cluster-based two-tier topology that has high capable cluster heads, this scheme can be applied with simple modification for various topologies such as homogeneous networks and more hierarchical topology like three-tier networks. As the proposed cross-layer design is a distributed centralized scheme based a cluster-based topology, it requires more computation and communication overheads compared to the other network models without the topological structure. Also, the energy consumption of a cluster head is greater than that of other nodes.

CHAPTER 7

CLASS-BASED LOCAL ADDRESSING

A wireless sensor network is an application-specific information gathering platform. The application in sensor networks may be able to divided into several sub-tasks/applications and each sub-application can run at a different base station in different location to distribute load. For example, the forest management system may include several applications for temperature, sound, humidity, chemical value and image sensing to protect a forest fire and pollution or for statistical researches as shown in Figure 7.1.

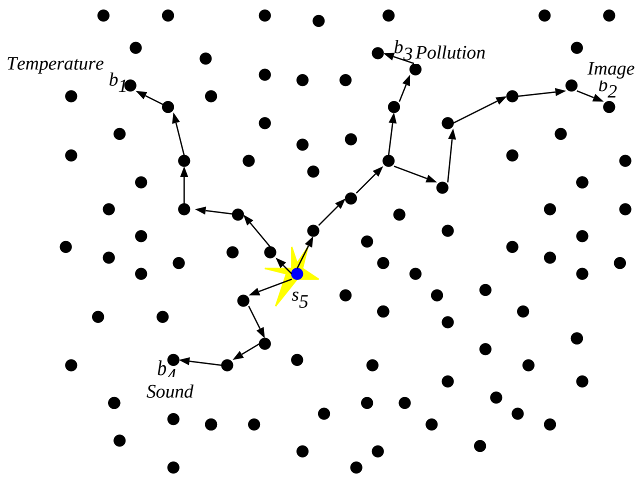


Figure 7.1. Example Network Scenario with Multiple Sinks.

These different type of applications may require different quality of service(QoS) while gathering information from sensors as illustrated in Figure 7.2. For example, a temperature sensing value which is greater than a certain threshold should be delivered as soon as possible due to the possibility of a forest fire. However, humidity sensing values may not need to be considered as critical data since even an abnormal value might not notify any critical situation like change temperature. An image sensing data may need to be delivered with bandwidth constraint to a base station to let a user to analyze the situation.

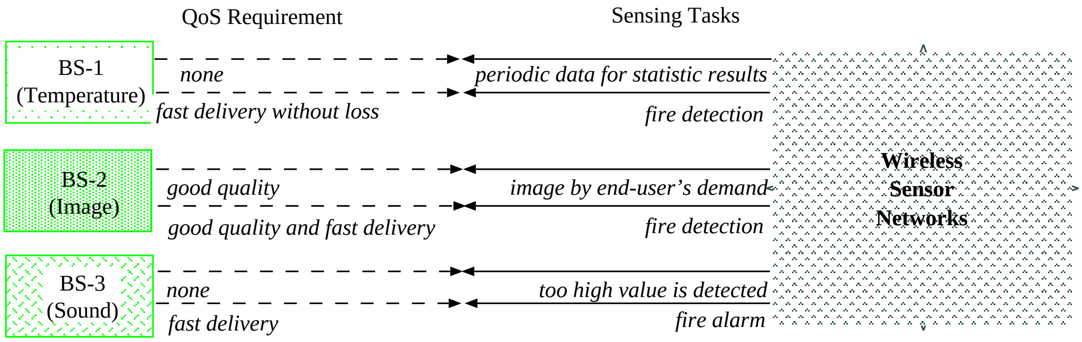


Figure 7.2. QoS support Based on Sensing Tasks.

7.1 Preliminaries

Our motivation lies in the fact that application-specific QoS should be supported depending on different type of tasks/applications. We focused on query-based routing mechanism and attempted to make a category for application-specific queries more simple and efficient. We define a sensor, also called a source, s_i where $1 \leq i \leq n$ as a node which has sensing and relaying function. A sink, also called a base station, s_{*l} where $1 \leq l \leq m$ is defined as a node which has data gathering function and runs m number of applications at each base station. The location information of base stations are flooding at the network deployment time so each node know where each base station is placed. The proposed network shows n-to-m delivery model, covering n-to-*one* delivery either.

7.1.1 Problem Definition

We summarize the problem as follows:

Problem Definition: To support application-specific QoS in terms of data gathering transmission, each packet should be processed depending on different type of applications and service requirements at each intermediate node in a routing path.

To solve this problem, we use hierarchical classification to describe a query as illustrated in Figure 7.3. One high-level application, i.e., the forest management system, includes m number of sub-applications, i.e., temperature sensing, image sensing, etc., which run at m base stations separately.

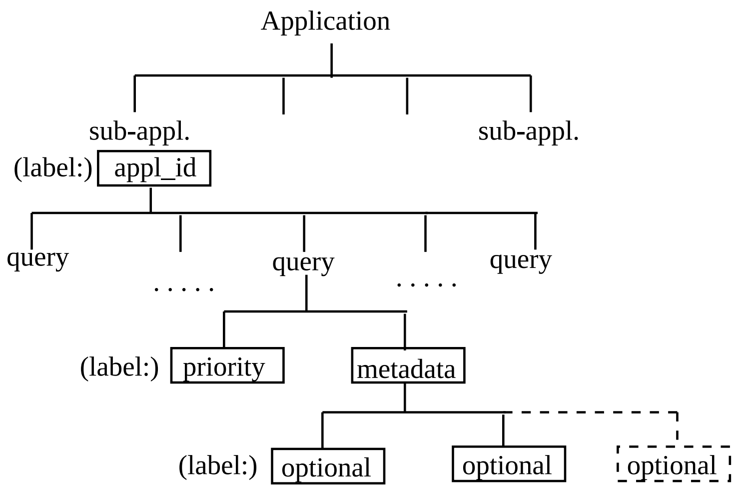


Figure 7.3. Hierarchical Labeling for Query Classification.

We define an *application ID* field in a label to distinguish sub-applications(or base stations). Each application uses its own queries to collect information from sensors and may require some level of service depending on type of queries. These are distinguished using a *query ID* and *priority* fields. We also define a *type* field to notify if a packet is for request or reply message etc. A *Metadata* field contains an abstract form of a query. Figure 7.4 illustrates some basic fields used in labels. Details of each field are presented in Section 7.3.

Next Hop ID	Application ID	Priority	Type	Metadata

Figure 7.4. Basic Fields used in Labels.

7.2 Network Model

A wireless sensor network is defined as a undirected connected graph $G = (V, E)$, where V is the set of the number of $\mathcal{N}(= n + m)$ nodes and E is the set of edges representing bidirectional communication path with several channels between sensors or between sensors and base stations. Each $\mathcal{N}$ node learn its local position and notice its one hop neighbors in its transmission range $\mathcal{R}$ at the network deployment time.

Definition 2.1: The set of one hop neighbors of s_i is defined as $N_{s_i} = \{n_j \mid$ where n_j is another sensor or base station in $\mathcal{R}$ of $s_i\}$.

Definition 2.2: Let t_{ij} be the response time which s_i receives a reply message from n_j after sending a message to n_j(i.e., T_{reply} - T_{hello}).

Figure 7.5 shows the process at the network deployment time.

We propose our AQR scheme based on the following assumptions:

- A network is covered by a large number of static and homogeneous sensors which are deployed randomly and uniformly. Sensor nodes send their data packet using multihop transmission if a base station is not in their transmission range.

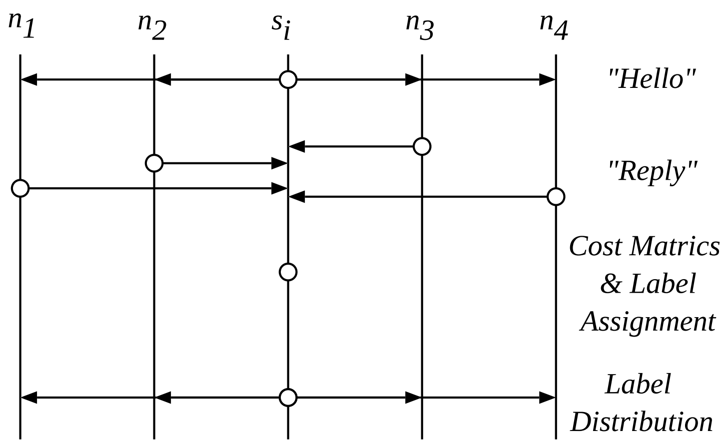


Figure 7.5. Process at the Network Deployment Time.

- A network has one or more base stations which run different tasks/applications and only base stations can send queries to sensor nodes.
- Metadata have been defined before the network deployment time.

7.3 Labels

Labels can represent lots of information such like data type, metadata, attributes, node ID, coordination or encryption/decryption key, etc. In order to support application-specific service differentiation, the AQR scheme should classify not only applications but queries used in each application. The proposed hierarchical labels make packet description more simple and efficient by reducing the number of bits used to represent each packet field and further reduce energy consumption. Using a label, a packet can be secured in some level since a packet description is changed with new bit sequence not in relation to the raw data(except a *next hop ID* field since it is exposed due to the label distribution process).

In this section, we present basic labeling fields, whose values are defined as F_f where f = {*next*, *appid*, *pri*, *type*, *meta*, ... } and lengths are as L_{F_f}.

7.3.1 Next Hop ID

F_{next} carries the information to which node a packet should be forwarded. A next hop is decided by F_{appid} and F_{pri} using a cost matric described in next Section.

NEXT_HOP_ID_SIZE The $L_{F_{next}}$ may have variable sizes depending on the size of a set of N_{s_i}, defined as k_{s_i}, and used as a system variable, NEXT_HOP_ID_SIZE in our scheme. Since we assume that sensors are uniformly distributed and every sensor is homogeneous, the expected number of N_{s_i}, $E[k_{s_i}]$, will be decided by the network density δ and transmission range $\mathcal{R}$. $E[k_{s_i}]$ is obtained by

$$E[k_{s_i}] = \lceil \pi R^2 \times \delta \rceil . \tag{7.1}$$

Then the number of bits to represent next hop ID field, $L_{F_{next}}$, is calculated by:

$$L_{F_{next}} = \lceil \log_2 k_{s_i} \rceil . \tag{7.2}$$

The number of N_{s_i} is not exactly $E[k_{s_i}]$ and F_{next} may need more bits than $L_{F_{next}}$ if k_{s_i} is greater than $E[k_{s_i}]$. However, the AQR scheme assigns the $L_{F_{next}}$ number of bits for NEXT_HOP_ID_SIZE to manage the maximum $E[k_{s_i}]$ of neighbors. Figure 7.6 shows the algorithm to manage one hop neighbors at a node s_i.

Label Assignment/Distribution Figure 7.7 shows the steps for F_{next} label distribution example at the network deployment time.

First, s_5 broadcasts *hello* message. The active nodes in $\mathcal{R}$ of s_5 receives this and sends *reply* message to s_5. However, the message from s_7 may not reach to s_5 due to the signal power or other obstructions. Then s_5 will receives *reply* messages from $N_{s_5} = \{s_1,\ s_2,\ s_3,\ s_4,\ s_6,\ s_8,\ s_{11}\}$ and makes a cost matric for the maximum

Algorithm *neighbors_managemnet*() **Begin**

receives a message from n_j;

if (table size $\geq$ NEXT_HOP_ID_SIZE)

  check the energy level of n_j; /* using signal power */

  if (energy level $\geq$ threshold)

    check the t_{ij};

    if ($t_{ij} \geq t_{ih}$) (n_h is already in the table)

      remove n_h;

      add n_j;

End-Algorithm

Figure 7.6. One-hop Neighbor Management Algorithm.

$E[k]$ neighbors. Based on the cost matric, s_5 assigns F_{next} and distributes the label to each neighbor. Label distribution process may be done using piggybacking, i.e., with a *reply* message, since it makes extra overhead. For example, they know s_5 is their neighbor when N_{s_5} receives *hello* message from s_5, so they can send a label for s_5 with a *reply* message.

Now, Figure 7.8 shows the example node s_2 which receives its labels from its N_{s_2}. s_2 may get duplicate label values, like 1001, as shown in Figure 7.8. Therefore, a node s_i should manage its distributed local labels from neighbors as well as assigned labels for neighbors. Consequently, each node in the AQR scheme has the extra overhead for memory and energy for the table for its label and its neighbors' label and for label distribution.

Data Forwarding A sensor node s_i may be a source to send its sensing data or an intermediate node to forward data from other sources. In the case that s_i is a source, s_i sets F_{next} with a label for the next hop to send a packet. In another case, s_i checks F_{next} whether the packet is for itself or not after receiving a packet from n_j. If the value is matched with the one for s_i which is previously distributed from n_j,

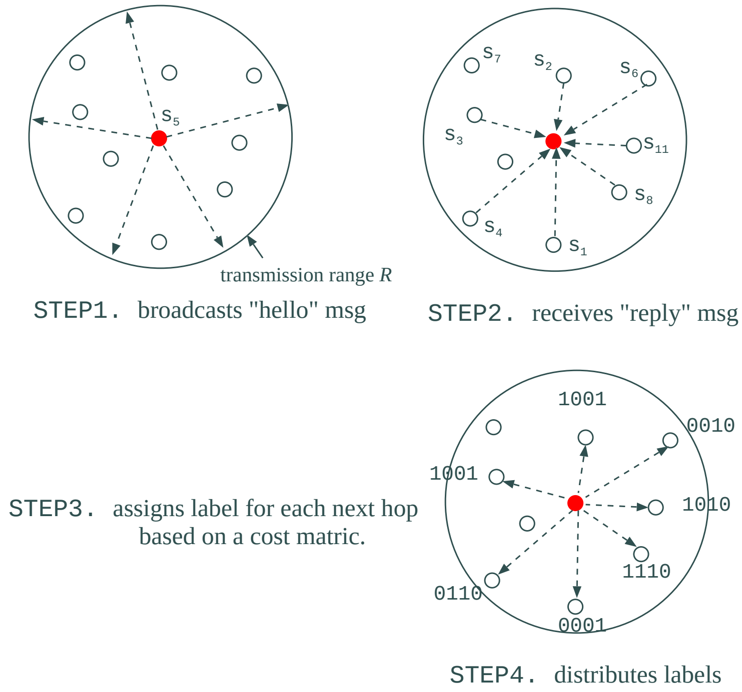


Figure 7.7. Next hop ID Label Distribution at s_5.

s_i changes F_{next} with a label for the next node using a label swapping algorithm and forwards the packet as illustrated in Figure 7.9.

7.3.2 Application ID

Based on the example network model in Section 7.2, we make the example labels for F_{appid} in Table 7.1.

F_{appid} is used to distinguish the final destination (note that we assume each application runs at a different base station) and make a perfect query with F_{meta} as shown in Figure 7.10. For example, if F_{meta} means "Send the current sensing task", a sensor sends the current temperature when F_{appid} indicates temperature sensing application or sends the snapshot when F_{appid} indicates image sensing application. Figure 7.11 shows the algorithm to process packets.

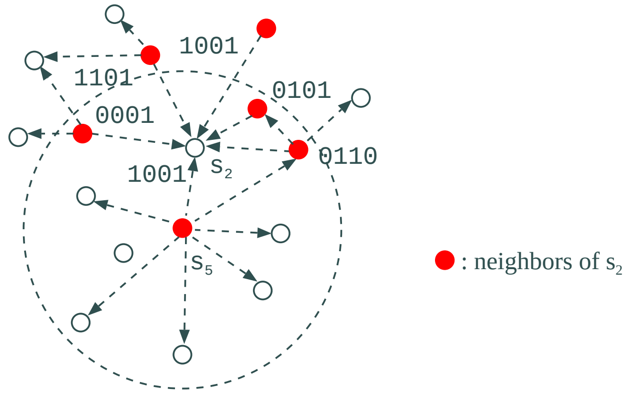


Figure 7.8. Label Distribution from neighbors to s_2.

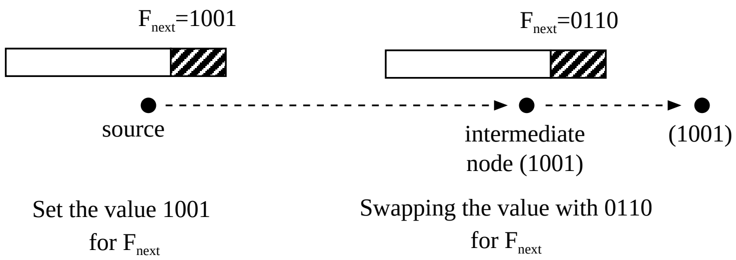


Figure 7.9. Example for Data Forwarding Process.

7.3.3 Priority

F_{pri} is based on the category of service differentiation such like critical, real-time, bandwidth constraint or best-effort service as shown in Table 7.2. For example, if a sensor node detects higher value than a certain threshold for temperature, the information should be delivered to the base station as soon as possible due to the possibility of a fire. At the same time, the user may require to capture an image or record a video to analyze the situation. In this case, intermediate nodes in the path from the source to the base station running the image sensing application may need to use more number of channels than that for normal case to forward the multimedia data packets with good quality in proper time.

Table 7.1. Example of F_{appid} based on the Network Model

Application	F_{appid}
Temperature	00
Image	01
Pollution	10
Sound	11

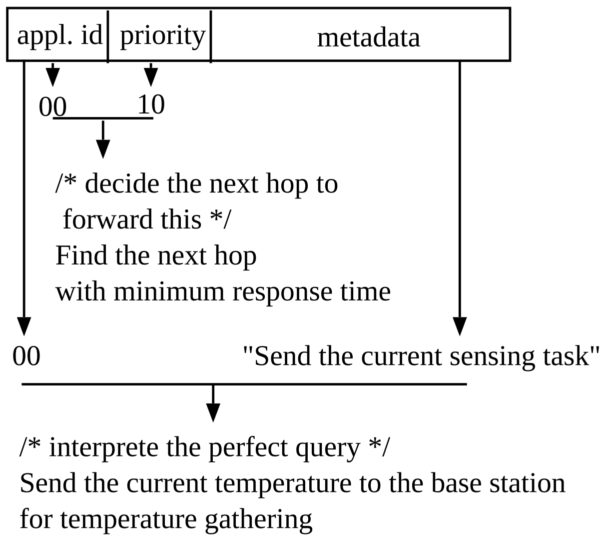


Figure 7.10. Label Combination for Packet Process.

When a network has m number of applications and p number of priority levels, each node needs to consider the maximum $m \times p$ number of cases to support application-specific QoS. However, not every application requires all p types of different services. For example, other applications except image sensing do not have bandwidth constraint in the example network model. Therefore, the number of cases to be considered to decide forwarding path will be less than $m \times p$.

F_{pri} can be decided when a query is classified as the metadata or when a user sends a query request. The default value of F_{pri} is *best effort*.

Algorithm *Packet_Process*() **Begin**

if (F_{next} == local ID for the source)
  lookup(F_{appid} & F_{meta});
    execute a requested job;
  lookup(F_{pri});
    check cost matric;
    switch F_{next};
else ignore the packet;

End-Algorithm

Figure 7.11. Algorithm for Packet Process.

Table 7.2. Example of Priority Field

Delivery Type	Label
Best effort	00
Bandwidth Constraint	01
Real-time	10
Critical	11

7.3.4 Metadata

Application providers can predict a user's interest depending on type of applications. In other words, queries are restricted on an application and an application provider can make a list of queries for a certain sensor network. Queries are usually described in natural language or high-level query language, such like SQL. In the AQR scheme, we propose to classify queries based on their expected operation results and change them as a shorten abstract form, called *metadata*. Defining metadata, the AQR scheme can reduce the number of bits to represent queries described in hierarchical labels. Metadata can be installed on sensor motes as system parameters with an application and provide some level of security meaning. That means even though an unauthorized node catches a packet in a network, he/she does not have any idea of the meaning of the packet without the mapping information between metadata and a query.

Table 7.3. Metadata Labeling

Query	Optional	F_{appid}	F_{meta}
Send current temperature	-	00	00101
Send current chemical value	-	10	00101
Warn a forest fire	200	00	11010
Warn an air pollution	2 × threshold	10	11010

A specific query from one application may require the same result with that of another application. For example, the query "If the temperature is greater than 200 degree, send the data as soon as possible" by a temperature sensing application, and the query "If the chemical value is two times greater than the standard one, send this data as soon as possible" by a pollution detection application can use the same metadata and priority, "Send sensing value which is greater than a certain threshold" with a critical priority. In this case, the term 'a certain threshold' can be used as '200 degree' for temperature sensing and 'two times greater than a standard value' for chemical sensing. This example is illustrated in Table 7.3.

Table 7.4. Definition of Packet Type

Packet Type	Example	Generator
request	Send a current value	base station
reply	95	sensor
command	Stop sending data	base station
event	200	sensor

Now we present some extra fields, as illustrated in Figure 7.12, which are related to metadata field below.

Type We define the type of packets, request, reply, command and event, in Table 7.4 and show the packet format for each type in Figure 7.12.

- *request* packet contains query request from a base station to sensors. It require for sensors to send data as a reply message.
- *reply* packet is a response for a query request message sent from a base station. It contains information for the query.
- *command* query is to ask some operations to sensors. Sensors do not need to send a reply message after finishing their job.
- *event* is generated from sensors. It is a kind of response for the query that sensors have known since their deployment time.

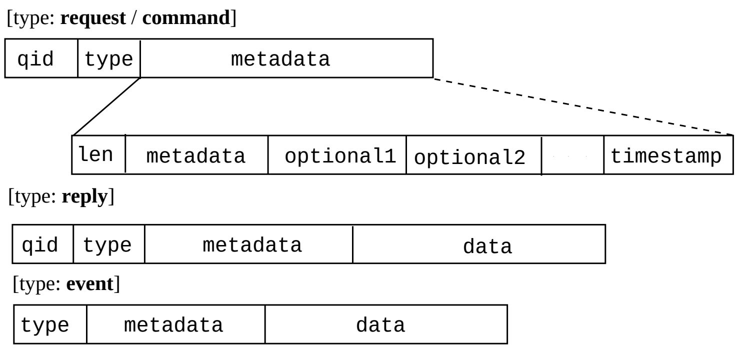


Figure 7.12. Packet Format.

Length/Optional The AQR scheme defines two more subfields, *length* and *optional*, since some metadata may require extra fields to include specific values, as shown in Table 7.5. *Optional* field will be added depending on the number of extra information as shown in Table 7.5. F_{len} is the number of optional fields and also describes the overall size of optional fields since the AQR scheme defines the size of one optional field as one byte.

Table 7.5. Sub-field and Metadata Example

Metadata	Extra Info	Num. of Optional field(F_{len})
Send a current value	-	0
Send a value greater than x	x	1
Send a value between x and y	x, y	2

In this dissertation, we do not consider optional fields for the performance of the AQR scheme since the number of bits consumed to describe the specific values is not different with other schemes. For example, AQR scheme defines *"unsigned 16"* to express a sound value which is the same format in an application of Crossbow product [29].

Query ID/Timestamp The AQR scheme uses *qid* field to distinguish a query sequence sent from a base station to sensors since a base station may send more than one queries before receiving a reply for the previous query from sensors. Whenever a base station sends a query(request) packet, it specifies sequential F_{qid} for each query. When sensors send a reply packet for a specific query, they also use F_{qid} as shown in Figure 7.12 and the base station can reuse the F_{qid} after receiving the reply message for the query.

timestamp field is used to specify how long the query request is available. If F_{time} is expired when a sensor receives the request message, the sensor just ignore the packet and the base station which sent the request removes the query request information from its cache and reuses the *qid* value.

$L_{F_{meta}}$ can be very flexible depending on the function of applications and this value is one important factor for performance of the proposed AQR scheme. Therefore, we need to analyze the overhead with different values of METADATA_FIELD_SIZE. In fact, if $L_{F_{meta}}$ is one byte, AQR scheme can define the 256 number of metadata and this number may be enough for most applications. In Section 7.5, we analyze the overhead of metadata field.

7.4 Dynamic Labeling(DL)

In this section, we present dynamic labeling(DL). The purpose of DL is to improve energy conservation by reducing the number of bits to describe a specific label value whose transmission frequency is very high. In general term, data delivery type can be classified as continuous, event-driven and on-demand(query-based) [30, 44, 26]. One example of continuous data is sending the current sensing value periodically. The frequency of this type of task may be much higher than that of event-driven or on-demand as the data is called *data stream* [24, 44, 35]. DL can be applied for *next hop ID* and *metadata* fields.

7.4.1 DL of Next Hop ID

Now our proposed AQR scheme introduces DL to describe F_{next}. First, a node s_i assigns F_{next} with variable length for each neighbors. We may use Huffman encoding proposed in [20]. While transmitting a packet, each node collects how many times the link has been used continuously and sets F_{next} with different size depending on this frequency value. The AQR scheme makes F_{next} of the link has been used more than a certain threshold, FREQUENCY_THRESHOLD, shorter by cutting off its most significant bit (MSB).

Figure 7.13 shows the example model. Each node has a variable FREQUENCY whose value is equal to FREQUENCY_THRESHOLD. If a node s_5 start sending packets to s_6, it decreases the value of FREQUENCY by 1. If this value is equal to 0 in certain time, s_5 cuts off the MSB of the original F_{next} for s_6 and sets the value of FREQUENCY to FREQUENCY_THRESHOLD again and repeat the process. When the original F_{next} is cut off, for example, the value 0010 is changed to 010, if there is the same value with 010 for s_8, s_5 switches the values each other, that means s_5 has the value 010 and s_8 does 0010.

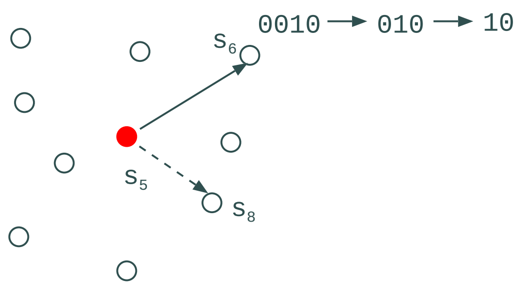


Figure 7.13. Dynamic Labeling of Next Hop ID.

7.4.2 DL of Metadata

As we mentioned earlier, transmission frequency of periodic data is very high. Therefore, we can assign shorter F_{meta} for the queries which require frequent transmission from sensors. Table 7.6 shows an example for dynamic labeling of metadata.

Table 7.6. Dynamic Labeling of Metadata

Metadata	F_{meta}
Send a current sensing value	10
Send an abnormal value	1101

7.5 Simulation Experiments

The performance of the DRC scheme using the local addressing scheme, called AQR, is evaluated through simulations in C++. The simulation results are based on randomly deployed static wireless nodes in a squared area with the fixed size of 300m X 300m. The simulation parameters are shown in Table 7.7. We used the collection coverage rate $\chi = 80\%$ and β is calculated by $(\mathcal{N} \cdot \chi)$ at each cluster head. A fixed 12 number of cluster heads have been deployed in the monitoring area regardless of the network density with number of nodes set to 100, 200, and 300 respectively to evaluate the effect of the cluster member size. The data reporting repeats a cycle, which consists of two data reporting rounds; each round is IntraDRC and InterDRC respectively. In the first round, β nodes transmit data to the cluster head in a contention-based mode using CSMA/CA while in the second round, a cluster head transmits the collected sensing results after data aggregation (in this simulation, we simply calculate the average of the collected data) to a sink located in the right top side of the monitoring area. 30% of packets during the simulations are randomly configured as delay constraint data. The performance of the DRC scheme is compared with IEEE 802.15.4* that uses the IEEE 802.15.4 medium access control mechanism and a fewest hop-based data reporting tree for data delivery from a cluster head to a sink. In IEEE 802.15.4*, we defined the superframe format with the guaranteed time slot (GTS).

7.5.1 Throughput

The average throughput is measured with 100, 200, and 300 nodes respectively, and the desired throughput is set to 70% for all simulations. As illustrated in Figure 7.14, the DRC scheme shows good performance when the number of nodes are 100 and 200. Although the performance gap between 200 nodes and 300 nodes is large, the

Table 7.7. Simulation Parameters

Simulation parameter	Value
The size of an area	300 m * 300 m
Bandwidth	19 Kbps
Transmission range	50 m
Transmit mode power	60 mW
Receive mode power	30 mW
Idle mode power	30 mW
Sleep mode power	0.003 mW
Transition power	30 mW
Transition time	2.45 ms
Packet size	96 bytes
Time slot size	42 ms
Simulation time	100 seconds

DRC scheme offers the desired throughput. The DRC scheme extremely improves the throughput performance by limiting the data reporting attempts of cluster members at each time slot, which is done by distributing the nodes into separate block sets. Although the data reporting chance is reduced, the DRC scheme still guarantees the desired throughput by managing β reporting nodes. In addition, the results also show that the DRC scheme is more tolerant to an increased number of nodes than IEEE 802.15.4*. The throughput performance of IEEE 802.15.4* is decreased sharply due to a variation in the network density.

In fact, the throughput performance is also affected by the data collection coverage rate. When the coverage rate is high, more reporting nodes will try to report data at a given time such that the probability of collisions increase. As other environmental factors (e.g., unstable wireless channel, signal strength, and unreliable reporting paths) increase the data reporting error, real experimentations will help in a more precise performance evaluation of the proposed scheme.

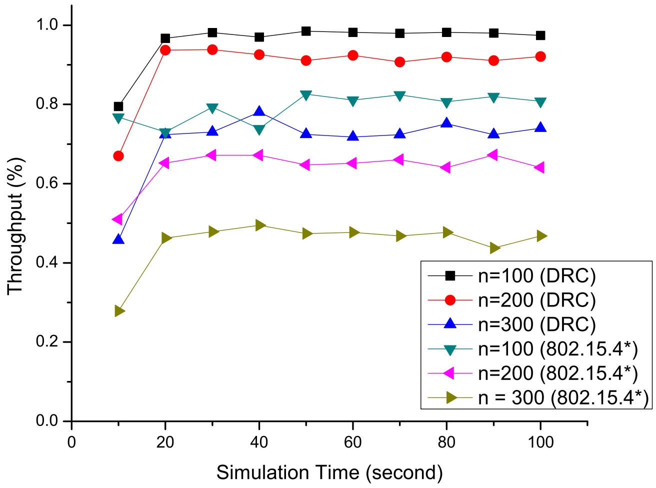


Figure 7.14. Throughput vs. Simulation Time.

7.5.2 Energy Consumption

Figure 7.15 shows the average energy consumption of cluster heads and member nodes with 100 and 200 nodes respectively. As mentioned earlier, the total number of cluster heads is fixed to 12 in the simulations. We have simulated both contention-based and schedule-based modes using the DRC scheme to learn how the channel access mode affects the results. The plots, however, are very similar albeit with slightly different values. This is because the DRC scheme provides a novel mechanism for node scheduling after grouping data reporting nodes for distributed channel access. Therefore, the probability of collisions is reduced as the number of potential contenders decreased. The difference in the operation between two channel access modes lies in whether a cluster head performs node scheduling to assign a particular node at a given time slot and it distributes the information to its members or not.

Each cluster head needs to learn about its member nodes as well as its adjacent clusters and constructs a fewest hop-based data reporting tree during the network setup time. After that, a cluster head continuously wakes up to collect data from

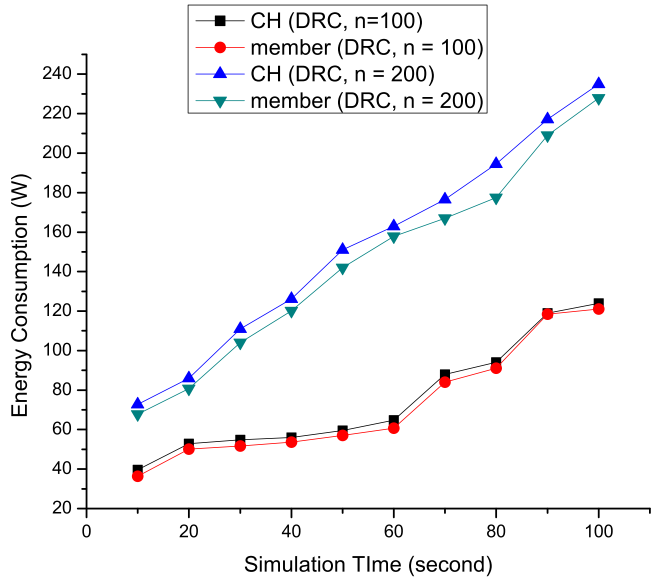


Figure 7.15. Energy Consumption (W) vs. Simulation Time (second).

its members and report to a sink. Therefore, the times at which the cluster heads can save energy are based on the inter-cluster reporting schedule and the duty cycle of sensor nodes. In this simulation, we did not adopt these kinds of sleeping mode mechanisms not to have the effects on the DRC scheme in the current stage.

In Figure 7.16, the average energy consumption of all the nodes in a monitoring area are compared with IEE 802.15.4* with two different network densities resulting from 100 and 200 nodes respectively. Although cluster heads consume more energy than cluster members for data reporting control, the simulation results show that nodes operating with the DRC scheme consumes less energy than the ones using IEEE 802.15.4*. This is because of the scheduling mechanism of the DRC scheme. Although the scheduling procedure requires extra energy, the nodes need not compete with other nodes once the schedule configuration is completed.

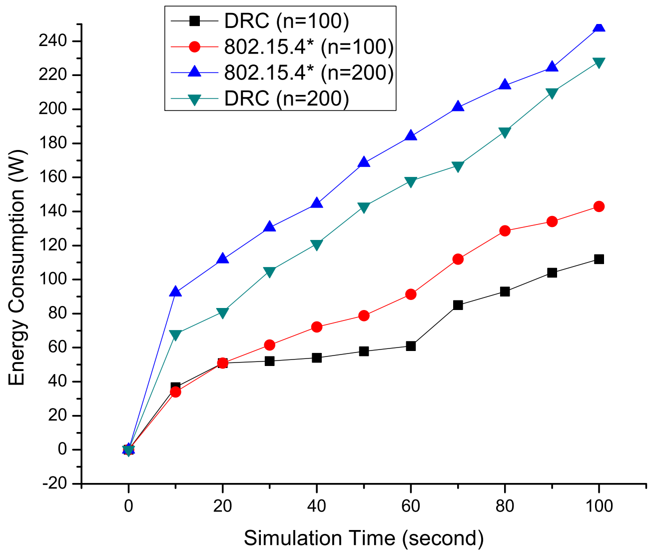


Figure 7.16. Energy Consumption (W) vs. Simulation Time (second).

7.5.3 End-to-End Delay

Figure 7.17 illustrates the average end-to-end delay performance of the DRC scheme with 100 and 300 nodes in both schedule-based and contention-based modes, and that of IEEE 802.15.4*. The comparisons did not show appreciable differences although we expected that the end-to-end delay of IEEE 802.15.4* would be slightly smaller than that of our scheme. However, the processing time for data reporting node selection and report scheduling at each cluster does not spend significant time. Another finding is in the comparisons of a contention-based mode and a schedule-based mode in the DRC scheme. Although the DRC scheme operates in the contention-based channel access mode, the required data reporting node selection and scheduling procedures in IntraDRC are the same with that in a contention-free mode. This comparison of two channel access modes also did not show perceived differences. The end-to-end delay was affected by the network density in both the DRC scheme and

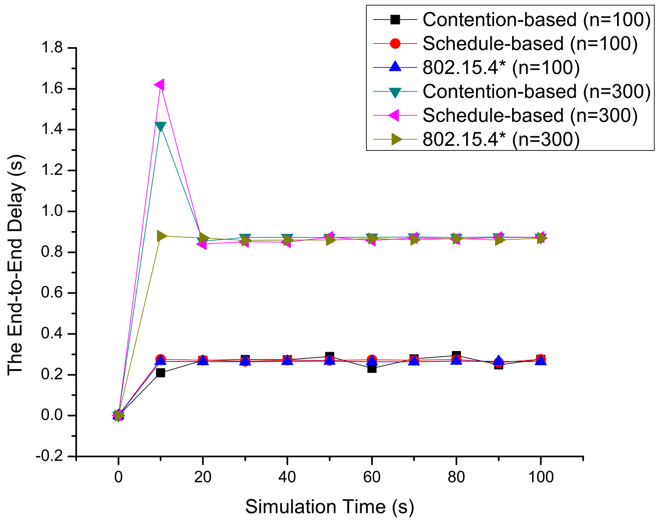


Figure 7.17. End-to-End Delay (second) vs. Simulation Time (second).

IEEE 802.15.4*. As the number of β data reporting nodes increased with 300 nodes in the network, the intra-cluster communication delay also increased resulting in an increase in the data collection time of the cluster head. As mentioned earlier, the IEEE 802.15.4* establishes a fewest hop-based data reporting tree. Simulation results show that the end-to-end delay of the DRC scheme offering energy-efficient data reporting is similar to that of IEEE 802.15.4.

7.6 Discussion

In this chapter, we proposed a local addressing scheme to reduce the energy consumption for data transmission. To improve energy conservation, the AQR scheme proposes a hierarchical query description using labeling. This scheme is designed for the scenario with different data sinks collecting different types of data to distribute load and define metadata which is a shorten abstract form of queries, but the system model with single data collection pointer does not degrade the performance.

The objective of the AQR scheme is to reduce the energy consumption used in intra-communications. As mentioned earlier, our integrated DRC framework separates the controls of data reporting within a cluster and that from one cluster to another. For the inter-cluster data reporting strategy, each cluster requires periodic data communications with its cluster members. As the energy consumption used for communication is the most expensive part in wireless sensor networks, our AQR scheme categorizes the types of control and query messages and designs hierarchical packet frame definition.

The number of metadata is an important factor to affect the performance of AQR scheme. Overhead analysis shows that label format in AQR scheme uses much less number of bits than a general packet format (e.g., "AM" type defined in TinyOS).

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

This dissertation addresses an integrated cross-layer framework to support QoS-aware data reporting in wireless sensor networks. We focus on the main characteristics of wireless sensor networks. Firstly, the operations of wireless sensor networks are application-specific. Secondly, one main role of a sensor network is to monitor physical environment, offers the interface between the end system and the networks, and controls the environment based on the user's requirements. Thirdly, a wireless sensor network defines its unique QoS parameters based on the applications; for example, in many cases the end system does not require maximized throughput.

Our framework is based on hierarchical topology based on single-hop clusters. Each cluster head collects information from its members, performs data aggregation, and forwards the results to a base station. Based on this network model, we separate data reporting control strategies into intra-cluster and inter-cluster data reporting schemes.

We considered the problem of data reporting control in a cluster-based wireless sensor network. Given a network topology, we discussed an intra-cluster data reporting control (IntraDRC) scheme and an inter-cluster data reporting control (InterDRC) scheme by adopting different QoS parameters for each scheme. The goal of IntraDRC is to collect the desired amount of data in a cluster from a subset of cluster member nodes and reduce the energy consumption during data reporting caused by competitions between data reporting nodes. IntraDRC selects data reporting nodes using block designs and schedules them by adopting the slot reservation request based on the status of a queue in a reporting node. Using the InterDRC scheme, each cluster head maintains two data reporting paths to offer differentiated delivery paths for an energy efficient reporting and a delay constraint reporting based on traffic characteristics. The reporting path decision is based on a fewest hop-based tree and a traffic-adaptive tree.

In the IntraDRC strategy, we proposed a class-based node allocation scheme including the selection of data reporting nodes that applies the block design concept from combinatorial theory and a novel two-phase node scheduling (TNS) scheme that defines class-based data reporting rounds and node assignment for each time slot. The TNS scheme includes a QoS-aware node selection strategy using block design and a two-phase data report scheduling algorithms. The scheduling algorithm is designed to support multiple priority classes. In this IntraDRC strategy, a certain number of data reporting nodes are selected in each cluster in order to satisfy the throughput fidelity specified by the applications while reducing redundant data reporting by selecting a subset of cluster members. This intra-cluster reporting control eventually helps control the overall amount of traffic in the network. The TNS scheme schedules data reporting while considering the priority of data, yet guaranteeing that sensor nodes compete with each other in the same class only.

In the InterDRC strategy, we proposed the QoS-aware data reporting construction scheme, called QRT, that balances the trade-off between the end-to-end delay and energy efficiency.. The QRT scheme maintains the flexible data reporting tree that considers the end-to-end delay constraints and energy efficiency. The idea of this QRT scheme is to manage variants of the data reporting tree based on two information, such as the hop counts to a data sink and the traffic amount generated from local area. For this purpose, each cluster head analyzes the traffic scenario of its cluster for load balancing and congestion control, thus improving the overall network performance. For the energy efficient operation, QRT measures the traffic load and balances the degree of a reporting tree to avoid heavy children. In InterDRC, the proposed spanning tree construction algorithm first builds the fewest hop-based reporting tree, used for delay constrained data delivery. This tree is updated with traffic load information in order to construct a traffic-adaptive reporting tree, used for energy efficient data delivery.

As the above two schemes, TNS and QRT, require communication overheads between a cluster head and its members, a local addressing scheme is proposed. The proposed local addressing scheme is designed to support QoS in a packet level by including the application-specific QoS information inside a packet while reducing the energy consumption caused by the communications for packets with a heavy header.

By separating the controls of data reporting within a cluster and that from one cluster to another, the proposed integrated framework can define different levels of various QoS parameters in each intra-cluster data reporting as well as inter-cluster reporting. To the best of our knowledge, we are the first to propose node arrangement using block designs in order to design task-specific data report scheduling in wireless sensor networks. This node arrangement strategy facilities an efficient local data collection in a cluster.

Compared to the other network models that does not consider the topological structure, the proposed schemes require more computation and communication overheads as it is based on a cluster-based topology and each cluster manages intra-/inter-cluster data reporting. Therefore, the energy consumption of a cluster head is greater than that of other ordinary nodes. In order to reduce the communication overheads, a local addressing scheme can be applied for local communications. For example, using binary number based addressing can simply include specific information by defining bit sequences.

Simulation results demonstrate that the proposed framework results in a significant conservation of energy by reducing the competition between data reporting nodes and establishing traffic-adaptive data reporting paths. The results also show that the throughput performance of our integrated framework is especially good due to stable data reporting independent of the network density.

Although our proposed integrated framework is based on a single-hop cluster-based topology, our scheme can be applied to various other topologies including more hierarchical topologies such as three-tier networks. When cluster heads have more powerful capabilities such as longer transmission range, the proposed DRC framework can achieve better performance. On the other hand, the failure of a cluster head may gracefully degrade the performance. Therefore, we plan to study possible solutions for cluster head failures in our future work. We also plan to integrate a local addressing scheme to save energy consumption for intra-cluster communications. More detailed performance analysis while changing network and QoS parameters can throw additional light on the efficacy of our strategies.

REFERENCES

[1] Drand: Distributed tdma scheduling for sensor networks.

[2] The network simulator - ns-2.

[3] A. Parekh and R. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. In *In Proceedings of International Conference of the IEEE Computer and Communications Societies (INFOCOM)* (1992).

[4] A.G. Ruzzelli, G.M.P. O'Hare, and R. Jurdak. Merlin: Cross-layer integration of mac and routing for low duty-cycle sensor networks. In *Ad Hoc Networks* (2008), vol. 6, pp. 1238–1257.

[5] Akyildiz, I., and Wang, X. A survey on wireless mesh networks. In *IEEE Radio Communications* (September 2005), pp. 23–30.

[6] B. Liu and D. Towsley. A study on the coverage of large-scale sensor networks. In *In proceeding, 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)* (2004).

[7] Bing, L., Lin, Z., and Huimin, Z. An adaptive schedule medium access control for wireless sensor networks.

[8] C. Liu, K. Wu, and J. Pei. An energy-efficient data collection framework for wireless sensor networks by exploiting spatiotemporal correlation. In *IEEE Transactions on Parallel and Distributed Systems* (July 2007), vol. 18, pp. 1010–1023.

[9] C. Schurgers and V. Tsiatsis and S. Ganeriwal and M. Srivastava. Optimizing sensor networks in the energy-latency-density design space. In *IEEE Transactions on Mobile Computing* (2002), vol. 1, pp. 78–80.

[10] Chong, C. Y., and Kumar, S. P. Sensor networks: Evolution, opportunities, and challenges. In *Proceedings of the IEEE* (August 2003), vol. 91, p. 1247ff.

[11] Clark, M. P. *Networks and Telecommunications: Design and Operation*. John Wiley and Sons Ltd, Baffins Lane, Chichester, West Sussex, England, 1991.

[12] C.Y. Wan, S.B Eisenman, and A.T. Campbell. Coda: Congestion detection and avoidance in sensor networks. In *ACM SenSys* (November 2003).

[13] D. England, B. Veeravalli, and J.B. Weissman. A robust spanning tree topology for data collection and dissemination in distributed environments. In *IEEE Transactions on Parallel and Distributed Systems* (May 2007), vol. 18, pp. 608–620.

[14] D. Estrin, D. C., and Pister, K. Connecting the physical world with pervasive networks. In *IEEE Pervasive Computing* (January 2002).

[15] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. New century challenges: Scalable coordination in sensor networks. In *ACM Mobicom* (1999).

[16] D. Rakhmatov and S. Vrudhula. Energy management for battery-powered embedded systems. In *ACM Transactions on Embedded Computing Systems* (August 2003), vol. 2.

[17] Deb, B., Bhatnagar, S., and Nath, B. ReInForm: Reliable Information Forwarding Using Multipaths in Sensor Networks. In *Proceedings of IEEE Conference on Local Computer Networks (LCN)* (October 2003).

[18] E. Cayirci, R. Govindan, T. Znati, and M. Srivastaval. Wireless sensor networks. In *Computer Networks: Intl. Journal of Computer and Telecommunications Networking* (November 2003), vol. 43.

[19] F. Ye. et al. Peas: A robust energy conserving protocol for long-lived sensor networks. In *In proceeding, 23rd International Conference on Distributed Computing Systems* (2003).

[20] G. Kulkarni, C. S., and Srivastava, M. Dynamic Link Labels for Energy Efficient MAC Headers in Wireless Sensor Networks. In *IEEE SENSORS* (June 2002), vol. vol. 2, pp. 1520–1525.

[21] H. J. Choe, P. Ghosh, and S. K. Das. Cross-layer design for adaptive data reporting in wireless sensor networks. In *The 1st International Workshop on Information Quality and Quality of Service for Pervasive Computing (IQ2S) 2009 in conjunction with IEEE PerCom 2009* (March 2009).

[22] H. J. Choe, P. Ghosh, Kalyan Basu, and S. K. Das. Class-based data reporting scheme in heterogeneous wireless sensor networks. In *The 18th International Conference on Computer Communications and Networks (ICCCN)* (August 2009).

[23] H. Li, P. Shenoy, and K. Ramamritham. Scheduling communication in real-time sensor applications. In *In Proceedings of the 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)* (May 2004).

[24] Hammad, M., Aref, W., and Elmagarmid, A. Stream Window Join: Tracking Moving Objects in Sensor-Network Databases.

[25] Heinzelman, W., Kulik, J., and Balakrishman, H. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proceedings of ACM Mobile Computing and Networking (MOBICOM)* (1999), pp. 174–185.

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

[27] I. Lazaridis, et al. Quasar: Quality-aware sensing architecture. In *SIGMOD Record* (2004), vol. 33, pp. 26–31.

[28] I.F. Akyildiz, and I.H. Kasimoglu. Wireless sensor and actor networks: Research challenges. In *Ad Hoc Networks* (2004), vol. 2, pp. 351–367.

[29] Inc., C. T. http://www.xbow.com.

[30] Intanagonwiwat, C., Govindan, R., and Estrin, D. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proceedings of ACM Mobile Computing and Networking (MOBICOM)* (2000), pp. 56–67.

[31] K. Jammieson and H. Balakrishnan and Y. C. Tay. Sift: A mac protocol for event-driven wireless sensor networks. In *Technical Paper, MIT Laboratory for Computer Science* (May 2003), p. 894.

[32] K. Sohrabi and G. J. Pottie. Performance of a novel self-organization protocol for wireless ad-hoc sensor networks. In *IEEE Vehicular Technology Conference (VTC)* (Fall 1999), pp. 1222–1226.

[33] K. Sohrabi and J. Gao and V. Ailawadhi and G. J. Pottie. Protocols for self-organization of a wireless sensor network. In *IEEE Personal Communications* (2000), vol. 7(5), pp. 16–27.

[34] Kahn, J. M. Next century challenges: Mobile networking for smart dust. In *Proceedings of the 5th ACM Intl. Conference on Mobile Computing and Networking (MobiCom)* (Seattle, WA, August 1999), pp. 270–278.

[35] Krishnamachari, B., Estrin, D., and Wicker, S. *Modeling Data-Centric Routing in Wireless Sensor Networks.* USC Tech. Report CENG 02-14, 2002.

[36] Kulik, J., Heinzelman, W. R., and Balakrishnan, H. Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks. In

Proceedings of ACM Mobile Computing and Networking (MOBICOM) (1999), pp. 61–69.

[37] L. Song, and D. Hatzinakos. A cross-layer architecture of wireless sensor networks for target tracking. In *IEEE/ACM transactions on Networking* (February 2007), vol. 15, pp. 145–158.

[38] LEE, W. Flexitp implementation in ns-2, 2007.

[39] Lin, P., Qiao, C., and Wang, X. Medium access control with a dynamic duty cycle for sensor networks. 1534–1539.

[40] Lu, G., Krishnamachari, B., and Raghavendra, C. S. An adaptive energy-efficient and low-latency mac for data gathering in wireless sensor networks. 224, 26–30.

[41] M. Hall, Jr. *Combinatorial Theory.* Blaisdell Publishing Company, Waltham, Massachusetts and Toronto, London, 1967.

[42] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi. Robust multi-hop time synchronization in sensor networks. In *Proceedings of International Conference on Wireless Networks (ICWN)* (June 2004).

[43] M. Perillo, Z. Ignjatovic, and W. Heinzelman. An energy conservation method for wireless sensor networks employing a blue noise spatial sampling technique. In *In proceeding, 3rd International Symposium on Information Processing in Sensor Networks (IPSN)* (2004).

[44] Manjeshwar, A., Zeng, Q.-A., and Agrawal, D. P. An Analytical Model for Information Retrieval in Wireless Sensor Networks Using Enhanced APTEEN Protocol. *IEEE Transactions on Parallel and Distributed Systems vol. 13*, 12 (2002), 1290–1302.

[45] O.B. Akan, and I.F. Akyildiz. Esrt: Eveint-to-sink reliable transport for wireless sensor networks. In *IEEE/ACM Transactions on Networking* (October 2005).

[46] Optimal Block Design for Asynchronous Wake-Up Schedules and Its Applications in Multihop Wireless Networks. Optimal block design for asynchronous wake-up schedules and its applications in multihop wireless networks. In *IEEE Transactions on Mobile Computing* (September 2006), vol. 5, pp. 1228–12410.

[47] Q. Liang, D. Yuan, Y. Wang, and H.H. Chen. A cross-layer transmission scheduling scheme for wireless sensor networks. In *Computer Communications* (2007), vol. 30, pp. 2987–2994.

[48] R. Sivakumar, P. Sinha, and V. Bharghavan. Cedar: A core-extraction distributed ad hoc routing algorithm. In *IEEE Infocom* (2004), pp. 202–209.

[49] Rajendran, V., Obraczka, K., and Garcia-Luna-Aceves, J. J. Energy-efficient, collision-free medium access control for wireless sensor networks.

[50] R.L. Cruz and A.V. Santhanam. Optimized routing, link scheduling and power control in multi-hop wireless networks. In *In proceedings of the IEEE INFOCOM* (March 2003).

[51] T. He, J. A. Stankovie, C. L., and Abdelzaher, T. SPEED: A Stateless Protocol for Real-time Communication in Sensor Networks.

[52] T. V. Dam and K. Langendoen. An adaptive energy-efficient mac protocol for wireless sensor networks. In *SenSys* (Los Angeles, CA, USA, November 2003).

[53] T. Znati, C. Raghavendra, and K. Sivalingam. Special issue on wireless sensor networks. In *Mobile Networks and Applications* (August 2003), vol. 8.

[54] V.C. Gungor, M.C. Vuran, and O.B. Akan. On the cross-layer interactions between congestion and contention in wireless sensor and actor networks. In *Ad Hoc Networks* (2007), vol. 5, pp. 897–909.

[55] W. B. Heinzelman and A. P. Chandrakasan and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. In *IEEE Transactions on Wireless Networking* (2002), vol. 1(4), pp. 660–670.

[56] W. Choi and S. K. Das. A Novel Framework for Energy-Conserving Data Gathering in Wireless Sensor Networks. In *Proceedings of IEEE Conference on Computer Communications (INFOCOM)* (2005).

[57] W. Choi, and S.K. Das. Coverage-adaptive random sensor scheduling for application-aware data gathering in wireless sensor networks. In *Computer Communications* (2006), vol. 29, pp. 3467–3482.

[58] W. Choi, S. K. Das, and H. J. Choe. Constrained random sensor selection for application-specific data gathering in wireless sensor neetworks. In *2nd Workshop on Embedded Networked Sensors (EmNetS-II)* (May 2005).

[59] W. Heinzelman. Middleware to support sensor network applications. In *IEEE Network Magazine, Special Issue* (2004), vol. 18, pp. 6–14.

[60] W. L. Lee, Amitava Datta, and R. Cardell-Oliver. Flexitp: A flexible-schedule-based tdma protocol for fault-tolerant and energy-efficient wireless sensor networks. In *IEEE Transactions on Parallel and Distributed Systems* (June 2008), vol. 19, pp. 851–864.

[61] X. Wang, et al. Integrated coverage and connectivity configuration in wireless sensor networks. In *In proceeding, Second ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2003).

[62] X. Wang, G. Xing, Y. Zhang, R.P.C. Lu, and C. Gill. Integrated coverage and connectivity configuration in wireless sensor networks. In *in Proceedings*

of ACM Conference on Embedded Networked Sensor SYstem (SenSys) (2003), pp. 28–39.

[63] Y. C. Tay and K. Jamieson and H. Balakrishnan. Collision-minimizing csma and its applications to wireless sensor networks. In *IEEE Journal on Selected Areas in Communications* (August 2004), vol. 22(6), pp. 1048–1057.

[64] Y. Chen, and Q. Zhao. An integrated approach to energy-aware medium access for wireless sensor networks. In *IEEE Ttransactions on Singal Processing* (July 2007), vol. 55, pp. 3429–3444.

[65] Y. Liu, I. Elhanany, and H. Qi. An energy-efficient qos-aware media access control protocol for wireless sensor networks. In *In Proceedings of International Conference on Mobile Ad Hoc Sensor Systems (MASS)* (November 2005).

[66] Y. Yuan, Z. Yang, Z. He, and J. He. An integrated energy aware wireless transmission system for qos provisioning in wireless sensor network. In *Computer Communications* (2006), vol. 29, pp. 162–172.

[67] Ye, W., Heidemann, J., and Estrin, D. An energy-efficient mac protocol for wireless sensor networks. In *INFOCOM* (New York, June 2002).

[68] Ye, W., Heidemann, J., and Estrin, D. Medium access control with coordinated, adaptive sleeping for wireless sensor networks. In *IEEE/ACM Transactions on Networking* (2004).

[69] Y.F. Wong, L.H. Ngoh, and W.C. Wong. An adaptive wakeup scheme to support fast rouing in sensor networks. In *International Workshop on Modeling Analysis and Simulation of Wireless and Mobile Systems, Proceedings of the 2nd ACM international workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks* (2005), pp. 18–24.

[70] Y.F. Wong, L.H. Ngoh, W.K.G. Seah, and W-C. Wong. Dual wakeup design for wireless sensor networks. In *Computer Communications* (September 2009), vol. 32, pp. 1–13.

BIOGRAPHICAL STATEMENT

Jay V. Smith was born in Atteel, Palestine, in 1965. He received his B.S. degree from Baghdad University, Iraq, in 1988, his M.S. and Ph.D. degrees from The University of Texas at Arlington in 1993 and 1999, respectively, all in Electrical Engineering. From 1988 to 1991, he was with the department of Electrical Engineering, Baghdad University as an Instructor in the Communications and Electronics Labs. In 1995, he joined Nortel Networks, Wireless Engineering Networks as RF engineer then as a technical advisor for the design and optimization of CDMA mobile and fixed wireless access systems. His current research interest is in the area of CDMA communications for mobile cellular systems. He is a member of several IEEE societies.