Performance Evaluation on IEEE 802.15.4 Scatternet Sensor Networks

A thesis presented
by

Jun Fung

to
The Department of Computer Science
in partial fulfillment of the requirements
for the degree of
Master of Science
in the subject of

Computer Science

The University of Manitoba
Winnipeg, Manitoba
May 2007

Thesis advisor | Author

Jelena Misic | **Jun Fung**

Performance Evaluation on IEEE 802.15.4 Scatternet Sensor Networks

Abstract

LR-WPAN networks are equipped with battery-powered sensors and changing batteries is considered impractical. Therefore, energy saving is very important when building such networks. IEEE 802.15.4 is a standard for LR-WPAN, which specifies the Physical layer and MAC layer for LR-WPAN. Scatternet is a network structure where sensors are grouped into clusters using star topology - piconets. Each piconet has a coordinator as the communication center. Two piconets are connected by an intermediate node - bridge. In my thesis work, I have investigated the feasibility of building a large-scale LR-WPAN scatternet network using a simulation model.

My work can be divided into three phases. In the first phase, I studied the performance of the bridge communication by simulating a two piconets model where piconets are interconnected by a Master/Slave bridge. I evaluated the performance of the inter-cluster communication and found the performance saturation mode is reached when the traffic load is high enough in each cluster. I also compared the performance of bridges using different communication protocols and parameters. In the second phase, I applied a duty cycle technique to each node to let the network to achieve the prescribed event sensing reliability by a distributed activity management;

In order to balance the lifetime of all the clusters in the network, the nodes compensation method was proposed to extend the lifetimes of heavily loaded clusters. In the third phase, I proposed a scalable, energy-efficient and lifetime-balance routing algorithm for the network to extend the network lifetime as much as possible; My simulation results show that when the network is using the lifetime-balance routing algorithm, the workload of clusters are balanced and the network lifetime is extended effectively.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisor Dr. J. Misic, my committee, my parents, my family, and all the people who have supported me along the way.

This thesis is dedicated to my children Jerome and Timothy.

Chapter 1

Introduction

In recent years, low power, low data rate Wireless Personal Area Networks (LR-WPAN) are attracting much attention in the industry. WPAN is composed of battery-equipped sensors which communicate at a very low data rate. Each sensor is the combination of a sensing unit and a communication unit. The sensing unit is responsible for sensing environment data such as light, temperature or humidity and storing data into the data buffer. The communication unit is responsible for reading and processing data in the buffer and sending them out through wireless radio signals to the base station (sink). Data can be sent to the sink directly (single hop) or through other nodes (multi-hop). Figure 1.1 is an example of a typical multi-hop sensor network. In this figure, the laptop is the base station/sink for this network. In the following chapters of this thesis, I will use the term ”sink” instead of ”base station” to refer to the destination of all data.

In a LR-WPAN, The low cost sensors are scattered in unattended environments. Changing battery or recharging battery is considered impractical. In this situation,

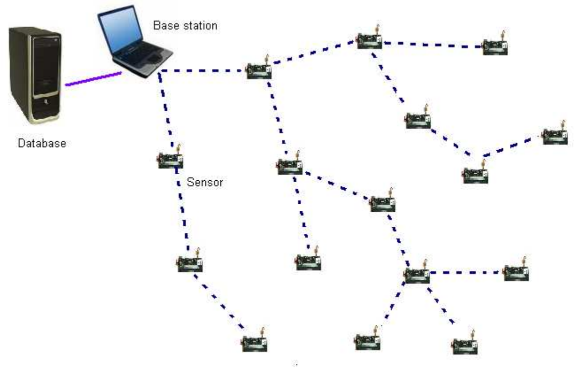


Figure 1.1: A sensor network structure

sensors should consume very little power so that they can survive for six months to two years with only two alkaline batteries in each sensor [9]. However, some testing reports have indicated that, if the sensors keep working, then the expected life time will only be around 72 hours. We must therefore expect that a sensor will spend most (99.5%) of its time in sleeping mode to achieve the expected life time.

LR-WPAN can be used in environmental, military, home, and commercial applications [4]. To make LR-WPANs marketable, much research was done to design energy-efficient communication protocols for LR-WPAN networks. There was no standardized protocol for LR-WPANs until October 2002. Philips, Motorola, Honeywell, Invensys and Mitsubishi Electric worked together and created the IEEE 802.15.4 standard [3], which specifies the physical layer (PHY) and media access control (MAC) for LR-WPANs.

Many research works were published based on the design and evaluation of such networks after the birth of the IEEE 802.15.4 standard. However, up to now, all the research have been based on a single cluster or on two or clusters. There is still no published work about research on larger scale IEEE 802.15.4 compliant networks. Large-scale sensor networks are useful in some applications such as environment disasters detecting or battle field surveillance. Cluster based network is considered to be more practical when building a large-scale dense network [25]. The reasons to use cluster structure include: cluster structure enables sensors to be free of routing duty so they can be put in sleep mode for a long time; the integration of data is much easier to be performed in the cluster heads; ease of the management of sensors. Our network will use clusters as the basic unit of the network. The topology inside a cluster uses star topology (Figure 1.2) (A simple star topology cluster is also called a piconet). The star topology structure is specified in [3]. However, is it practical to build a large scale LR-WPAN network? What are the problems to be solved to build such a network?

There are a few outstanding problems that need to be solved when building a large IEEE 802.15.4 compliant scatternet: first, the algorithm and performance of cluster interconnections; second, how to layout nodes in the network; third, energy efficient routing algorithms. In my thesis work, I provide solutions to address all the above problems and the solutions will be described briefly later in this chapter.

There are two types of nodes in a star topology network. One is called a coordinator and the other is called a device. A coordinator is the node which acts as the communication center of a cluster. The responsibility of a coordinator is to broadcast information through control signals (beacons), collect data from the device sensors and then process the collected data and send the data to the base station. There is only one coordinator in one cluster. Since a coordinator involves in all the com-

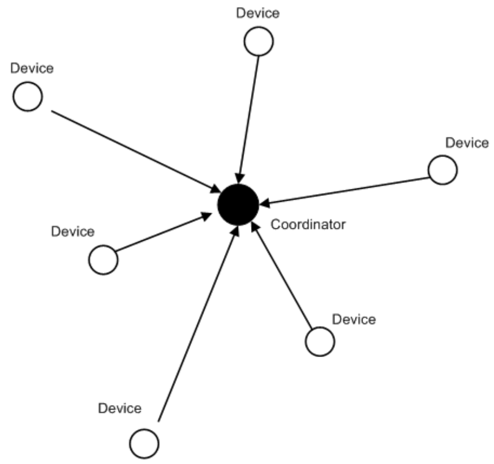


Figure 1.2: Star topology structure (adapted from [3])

munication in the cluster and is responsible to do all the broadcasting work, so it uses much more power than an ordinary device. Therefore, coordinators are either equipped with much powerful batteries or they are line powered. A device is a sensor which is responsible to sense data and transmits data to the cluster coordinator. There can be multiple devices in a cluster and the maximum number is 255. A device is usually equipped with ordinary batteries, such as a 2 AA Alkaline batteries. The communication protocol between an ordinary device and a coordinator is CSMA-CA [3]. A cluster following the star topology is called a piconet. A scatternet is a network structure where sensors are grouped into piconets and piconets are connected through intermediate nodes - bridges.

Bridge algorithms are not specified in IEEE 802.15.4 standard. However, the concept of bridges is specified in Bluetooth (IEEE 802.15.1 [2]). There are two types of bridges, Master/Slave bridge and Slave/Slave bridge (shown in Figure 1.3), defined in standard IEEE 802.15.1. A Master/Slave (M/S) bridge uses coordinators as the intermediate nodes. In the M/S bridge scheme, a coordinator collects data in its own piconet for one cycle and then switches to the next piconet and passes the data to the coordinator of the next piconet. In this process, the bridge switches between the roles of Coordinator (Master) and Device (Slave) and therefore it is called a M/S bridge. In a S/S bridge, an ordinary device is the bridge which switches between two piconets and downloads data from one piconet and uploads the data to the next piconet. In this process, the bridge acts as a Device (Slave) in both piconets, so it is called a S/S bridge. I adopted the conception of M/S bridge from Bluetooth and applied it to the IEEE 802.15.4 network. I designed the bridge scheduling algorithm and simulated the bridge communication between piconets. The simulation results show that a M/S bridge that uses the collision based CSMA-CA performs better than the TDM based Guaranteed Time Slot (GTS) bridge in low data rate while the GTS based bridge has higher capacity than the CSMA-CA based bridge. The bridge communication algorithm and the simulation results will be described and presented in chapter 3.

In LR-WPAN, all data have the same destination - sink. This determines the fact that piconets close to the sink will experience higher traffic load and therefore devices in those piconets consume more energy. If all the piconets have the same number of devices, then piconets close to the sink will be exhausted earlier than the others. We define a working network to be the network that all the poconets have alive sensors. Then, the lifetime of the network is the shortest lifetime among all the piconets. In order to avoid the situation that some piconets die much earlier than the others, the piconets close to the sink should have more redundant nodes. In chapter 4, I will describe a population compensation policy and use the simulation result to prove

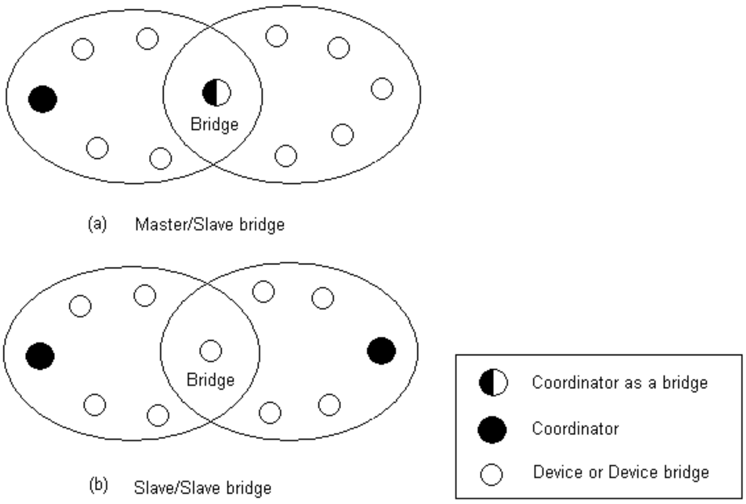


Figure 1.3: Illustration of M/S bridge and S/S bridge

that proper population compensation can extend the network lifetime effectively.

In a multihop sensor network, sensors need to decide which is the next hop to send the data. In a scatternet sensor network, ordinary sensors will only send data to their coordinstors while coordinators collect data and need to decide which is the next hop to send the data. It is noticable that improper routing will cause unbalanced traffic in the network and therefore will cause unbalanced lifetime among clusters. In chapter 5, I will propose a lifetime-balance routing algorithm in order to balance the lifetime of clusters effectively by adjusting the data flow to all the paths .

Contributions of this thesis are the following: First, I studied the bridge algorithms to communicate between clusters using a Master/Slave bridge. I found that the M/S bridge using acknowledged CSMA-CA protocol performs better than Guaranteed Time Slot (GTS) protocol when the data rate is very low. However, when the data rate is relatively high then GTS performs better. I also found the capacity of the M/S bridge in terms of the event sensing reliability and packet dropping rate. Second, I added sleeping function to all the devices so they can save energy and live longer. I found that the sleep technology can enable devices to control their utilization as low as possible to meet the minimum quality of service of the network. Therefore, the lifetime of each sensor is extended. Third, I studied the affection of cluster population to the lifetime of the cluster and found that the clusters close to the sink will die first and they need more population of sensors in order to extend their lifetime. Fourth, I found a sensors layout method and assigned channels to all the clusters to minimize the interference. I created a distributed routing scheme to balance the lifetime among clusters in the network. The purpose of this routing algorithm is to share workload evenly to all the piconets based on their power budgets and therefore balance the lifetime of all the piconets in the network. Finally, I integrated all the above solutions into a 7 clusters sensor network and present the performance and lifetime of such a network. My research results can provide useful information and methods when building a real sensor network.

The rest of the thesis is organized as following: Chapter 2 briefly discusses some other published works that are related to my thesis research. Chapter 3 describes the piconets inter-connection algorithms and scheduling. The performance of inter-connection using different protocols and parameters are evaluated. In chapter 4, I integrated duty cycle scheme into ordinary sensors and introduced nodes compensation to heavy-load clusters to equalize the lifetime among clusters. In chapter 5, I introduced a distributed routing algorithm to balance the lifetime among clusters and therefore extend the lifetime of the network further more. Finally, I conclude the

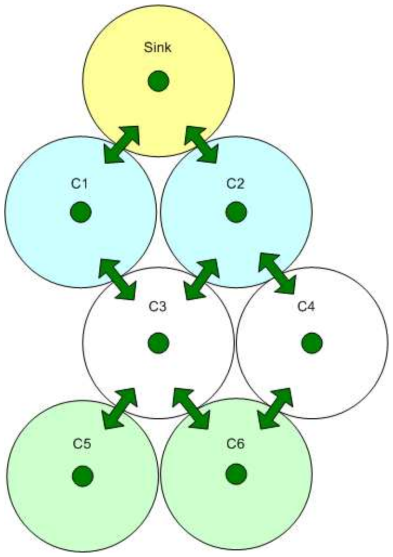


Figure 1.4: The topology graph of sensor network built on IEEE 802.15.4

whole thesis and indicate the future work that can be done.

Chapter 2

Related Work

In the last 5 years, there have been many papers published in the sensor networking field about sensor network routing and performance evaluation. Since energy consumption is the most important issue in LR-WPAN network and our network is a cluster-based network, I will discuss power-aware routing protocols and performance evaluation mainly for cluster based networks.

2.1 Power aware and energy efficient routing schemes in sensor networks

Routing algorithm in sensor network can be roughly divided into Power aware and Energy efficient routing. Power aware routing concerns the resident energy on nodes and tries to send data through the power-rich paths, while energy efficient routing tries to minimize the cost of sending data by finding lowest cost paths.

In 2004, Chang and Tassiulas [7] proposed "Maximum Lifetime Routing" for wireless sensor networks. Their scheme assumed that the energy consumption of packet transmission is a function of the distance to the next hop and, when some portion of the network is dead, the whole network is dead. They transform the lifetime problem into a linear programming problem and solve it to achieve optimal solutions. They also proposed a shortest-cost-path routing algorithm as a distributed implementation. The authors' simulation results showed that the lifetime of the shortest path routing algorithm is close to the optimal lifetime achieved by solving their linear programming problem. In their model, they assume the MAC layer can handle transmission perfectly. This means, packets are transmitted only once without error. In reality, there are always collisions in the MAC layer and it has to retransmit the packets when collisions happen. Therefore, their results are not applicable to LR-WPAN network.

Aslam et al. [5] proposed three power-aware routing algorithms for sensor networks. Their purpose was also to optimize the life time of the network. They assume that the energy consumption of a network node depends on which state it is in: transmitting, receiving or idling. Each state consumes a different level of energy. The first routing algorithm is called max-min $zPmin$ routing. The packet selects the path with the highest (maximum) minimum residual power node while the total power consumption is less than or equal to z times the minimum power consumption path (z is a constant that the user defines). This algorithm has a problem in that it is hard to get and maintain the power level on each node in a large network. Then the authors proposed a zone-based algorithm. They split the large network into smaller zones and applied the max-min algorithm in each zone. Thirdly, the distributed max-min algorithm was proposed, where sensors used a greedy algorithm to determine where to send the packets. In this way the excessive broadcasting cost was saved and the algorithm is easier to apply. However, all these three routing algorithms are unsuitable for LR-WPAN network since they need too much extra broadcasting to exchange power level information. This will lead the LR-WPAN network to waste energy in exchanging data and therefore have a shorter lifetime.

EAREC [8] by Eimon, et al. is a routing algorithm designed for clustered sensor networks where a duty cycle scheme is applied. Priority is introduced to balance the nodes' routing workload. Nodes with higher residual energy will have higher priority to route packets. In this way, the energy level balance of the network will be improved and this prolongs the network lifetime. However, EAREC doesn't account for MAC layer collisions and is still not a complete solution for LR-WPAN network.

Multipath Minimum Energy Routing [32] by Shouyi Yin and Xiaokang Lin proposes an optimal way to split data flow to multi-paths to reach the destination so that the energy consumption of data exchange can be minimized. However, their work is based on the assumption that energy consumption is a function of the data rate in a link. In my research, the energy consumption depends on the data rate reaching the cluster.

MESTER [31] by Yang Yang, et al. advocates developing a routing scheme based on throughput efficiency. The authors assume the transmission power is adjustable according to the communication distance and packets will be aggregated of each intersection. The sensors will pick the routing path with best energy efficiency. The network needs to change the routing structure periodically to evenly distribute the load in the network. However, this routing scheme is based on a TDMA MAC and the data aggregation is required. In my case, the MAC layer algorithm is contention based and data aggregation is optional.

There are many other energy aware/efficient routing algorithms for sensor networks. However, there is no routing algorithm for IEEE802.15.4 compliant clustered network. I will need to design a new routing algorithm for such a network.

Minimum Total Energy Routing (MTE) was proposed in [10] [28] [27]. The approach of these papers is to minimize the energy consumption for packets to reach the destination by choosing the shortest path or minimum hops or minimum cost path. These routing algorithms are mostly simple and easy to perform. However, when some of the paths are chosen, the battery of nodes along these paths can be exhausted quickly and will die out first. Therefore, pure MTE usually do not give the best solution.

Min-Max Battery Cost Routing (MMBCR) scheme [29] was proposed by Singh and Woo. Max-min considers the residual energy power for each node as the metric in order to enhance the lifetime of the nodes. The routing path is chosen by calculating the minimum residual energy values along each path and choosing the path which has the highest (Maximum) minimum residual energy. However, MMBCR does not consider the communication cost at all. In order to solve this, Toh proposed Conditional Max-Min Battery Capacity routing (CMMBCR) [30]. This routing combines the ideas of MMBCR and MTE routing. The minimum total transmission and reception energy path is chosen in the set of all paths whose minimum residual energy is above a given threshold. If the set if empty then max-min residual energy route is used.

MTE and Maxmin routing algorithms either look at the energy saving path or battery rich path, however, problem may occur that some nodes drains battery faster than the others for some reasons so we should protect these nodes from dying out fast. Kim et al. [15] used a new metric, the drain rate to find the best route. They defined a new cost function called $C_i = \frac{RBP_i}{DR_i}$ where RBP_i is the residual energy of node i and DR_i is the energy consumption rate for node i. C_i is the actually the estimated the time that node can survive. All the routing schemes above work on which path to go, however, this is not all we can do. Another category of routing algorithms proposed how to split the data flow to balance the network, such as Multi-path Minimum energy Routing [32] proposed by Yin et al. It proposes to find an optimal way to split data flow to multi-paths so that the energy consumption of data exchange can be minimized. This algorithm is based on the assumption that the energy consumption is a function of the data rate in a link. This is not true in our case.

2.2 Performance evaluation for cluster-based sensor networks

Many research works were published based on the design and evaluation of sensor networks after the birth of the IEEE 802.15.4 standard. Bougard et al. [6] evaluated the power consumption of communications in one cluster of dense micro sensors network. Their results show the optimized parameters to minimize the average power consumption. Raghavendra et al. [16] evaluated the tradeoffs between energy consumption and throughput. Misic et al [21] [22] [19] studied the tradeoffs between throughput and delay, the effect of duty cycle mode to energy consumption in the network, and the performance of cluster inter-connection using intermediate node-bridges.

Lu et al. [16] did a performance evaluation for IEEE 802.15.4 compliant networks in 2004. They built a simulation model and measured the network performance subject to different parameters. The parameters used were the beacon interval and the number of devices in a cluster. The outputs were the throughput and the latency. They found there is a trade off between throughput and delay in such networks. The maximum throughput was 70kbps while it should be 250kbps if the bandwidth is fully utilized. That was the first paper on performance evaluation in an IEEE 802.15.4 network. However, their study is limited to one cluster. There are also some other papers about the performance evaluation in single cluster IEEE 802.15.4 networks, including [21] and [6].

Gupta and Younis [13] constructed a cluster-based sensor network and introduced a gateway, which is a node with much higher energy resources. The gateway is not only the coordinator for the cluster, but also the only link to the external clusters. The authors compared two clustering criteria: one called load-balanced clustering and the other called shortest distance clustering. In load-balance clustering, gateways cluster nodes close to them while the size of the clusters should be equal. The shortest distance clustering only considers the distance of sensors and gateways. The simulation results show that load-balanced clustering networks have longer lifetimes than shortest distance clustering networks. This is a good piece of work on performance evaluation in WSN. However, it is not based on IEEE 802.15.4 networks.

All the above work is related to my work. However, the routing algorithms can not be applied to my research they either ignore MAC layer collisions or they are designed for peer-to-peer networks. Some more research remains to be done to find the appropriate routing algorithm for scatternet-based IEEE 802.15.4 compliant network. There are many more papers about the performance evaluation in sensor networks. The performance evaluation in my work will be similar but I will have my own set of variable measurements of interest.

In the next section, I will discuss my strategy for how to build an energy efficient IEEE 802.15.4 compliant network.

Chapter 3

Inter-cluster Communication

3.1 Introduction

My first step to build a large-scale sensor network is to build a two piconet connection model and study the performance of Master/Slave bridge in IEEE 802.15.4 compliant network. In this chapter, I analyzed a Master/Slave bridge using a simple two-piconet IEEE 802.15.4 compliant network. One cluster is called sink cluster, the other cluster is called a source cluster (Figure 3.1). The bridging function is achieved by partitioning time in both clusters into superframes with active and inactive periods. The coordinator/bridge services the nodes in the source cluster during the active period in that cluster, and uses the inactive period to visit the sink cluster. The coordinator of the sink cluster acts as the network sink and collects data from both the ordinary nodes in the sink cluster and data from the source cluster via the bridge.

In order to inspect the bridge performance, I built a simulation model and eval-

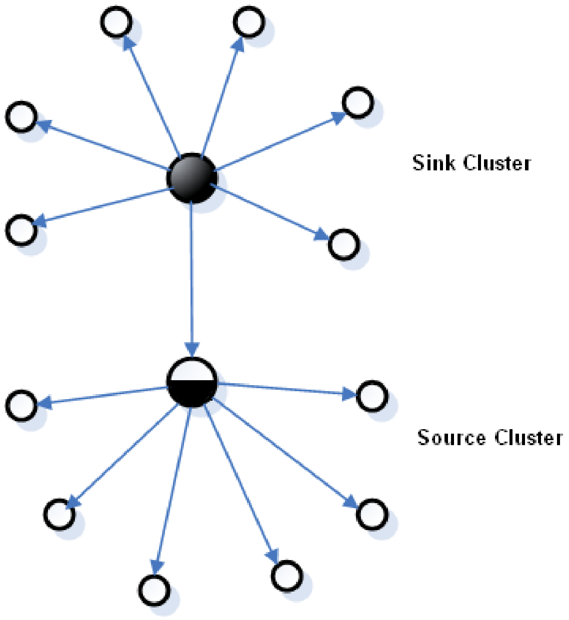


Figure 3.1: Two clusters source and sink model

uated the performance of the network. In this model, the communication protocol inside the source and sink piconet is using collision based CSMA-CA protocol [3]. The coordinator in source piconet is the bridge to connect the two piconets. The communication protocol that the bridge chooses to send data to the sink piconet coordinator can be CSMA-CA or Guaranteed Time Slot (GTS) protocol. A number of variable parameters are considered in this model, which includes the optional acknowledgments after each packet [3], the buffer size of devices and coordinators, the bit error rate, the number of devices in each cluster and the data arrival rate on each device. The impact of optional acknowledgement and the protocol of bridge are evaluated from low traffic load to high traffic load until the network is in a saturation mode. A network reaches saturation mode when the medium is high utilized and increasing the traffic load only degrade the performance of the network.

In my work, I simulated the bridge using CSMA-CA protocol and GTS using acknowledged exchange. I also simulated the situation that bridge uses acknowledged exchange and bridge uses unacknowledged exchange. The simulation results of the acknowledged exchange shows that when the network is in low traffic load, using CSMA-CA is superior to using GTS in packets' latency. While in higher traffic range, using GTS can has higher maximum throughput than using CSMA-CA since the latter reaches saturation mode earlier than using GTS. The simulation results of comparing acknowledged exchange and unacknowledged exchange shows that acknowledged exchange has higher throughput when the traffic load is low. With the increment of the traffic load, acknowledged exchange reaches saturation mode much earlier than unacknowledged exchange.

The chapter is organized as follows: In Section 1 I will describe the basic CSMA-CA and GTS protocol used in the MAC layer. In section 2 I will describes the bridging scheduling and algorithm. Section 3 I will present and discuss the simulation results, and in section 4 I will concludes this chapter.

3.2 Description of CSMA-CA protocol and GTS

In an IEEE 802.15.4-compliant wireless personal area network (WPAN), a PAN coordinator is the central controller device which builds a WPAN with other devices within a small physical area known as the personal operating area. Star topology and peer-to-peer topology are supported in LR-WPAN. In the star topology network, all communications, even those between the devices themselves, must go through the coordinator. In the peer-to-peer topology, the devices can communicate with one another directly - as long as they are within the physical range - but the PAN coordinator must be present nevertheless. The IEEE 802.15.4 standard also defines two channel access mechanisms, depending on whether a beacon frame (which is sent periodically by the coordinator) is used to synchronize communications or not. Beacon enabled networks use slotted carrier sense multiple access mechanism with collision avoidance (CSMA-CA), while the non-beacon enabled networks use simpler, unslotted CSMA-CA.

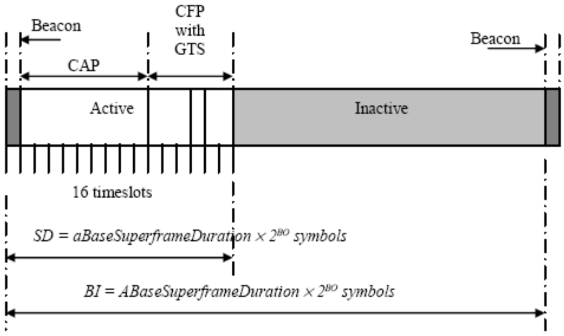


Figure 3.2: Superframe structure in beacon enabled networks [3]

In beacon enabled networks, channel time is divided into superframes (Figure 3.2) which are bounded by beacon transmissions from the coordinator. All communications in the cluster take place during the active portion of the superframe, which is divided into 16 equally sized slots. Beacon transmission commences at the beginning of slot 0, and the contention access period (CAP) of the active portion starts immediately after the beacon. Slots are further subdivided into backoff periods, the basic time units of the MAC protocol to which all transmissions must be synchronized. The actual duration of the backoff period depends on the frequency band in which the 802.15.4 WPAN is operating: 868 to 868.6MHz, 902 to 928MHz, or 2400 to 2483.5MHz [3]. The maximum data rates for these bands are 20kbps, 40kbps, and 250kbps, respectively.

3.2.1 CSMA-CA algorithm

During the CAP period, individual nodes access the channel using the CSMA-CA algorithm, the operation of which is schematically shown in Fig. 3.3. When a packet is arrives, The algorithm initializes NB to zero and CW to 2; the variable NB $= 0\ .\ .\ macMaxCSMABackoff - 1$ represents the index of the backoff attempt ($macMaxCSMABackoff$ is one of the parameters defined in [3]. The list of parameters are in table 3.1), while the variable CW = 0, 1, 2 represents the index of the Clear Channel Assessment (CCA) phase counter. If the device operates on battery power, as indicated by the attribute $macBattLifeExt$, the parameter BE (the backoff exponent which is used to calculate the number of backoff periods before the node device attempts to assess the channel) is set to 2 or to the constant $macMinBE$, whichever is less; otherwise, it is set to $macMinBE$ (the default value of which is 3). The algorithm then locates the boundary of the next backoff period; as mentioned above, all operations must be synchronized to backoff time units.

In step (2), the algorithm generates a random waiting time k in the range $0..2BE-1$ backoff periods. The value of k is then decremented at the boundary of each backoff period. Note that the counter will be frozen during the inactive portion of the beacon interval, and the countdown will resume when the next superframe begins. When this counter becomes zero, the device must make sure the medium is clear before attempting to transmit a frame. This is done by listening to the channel to make

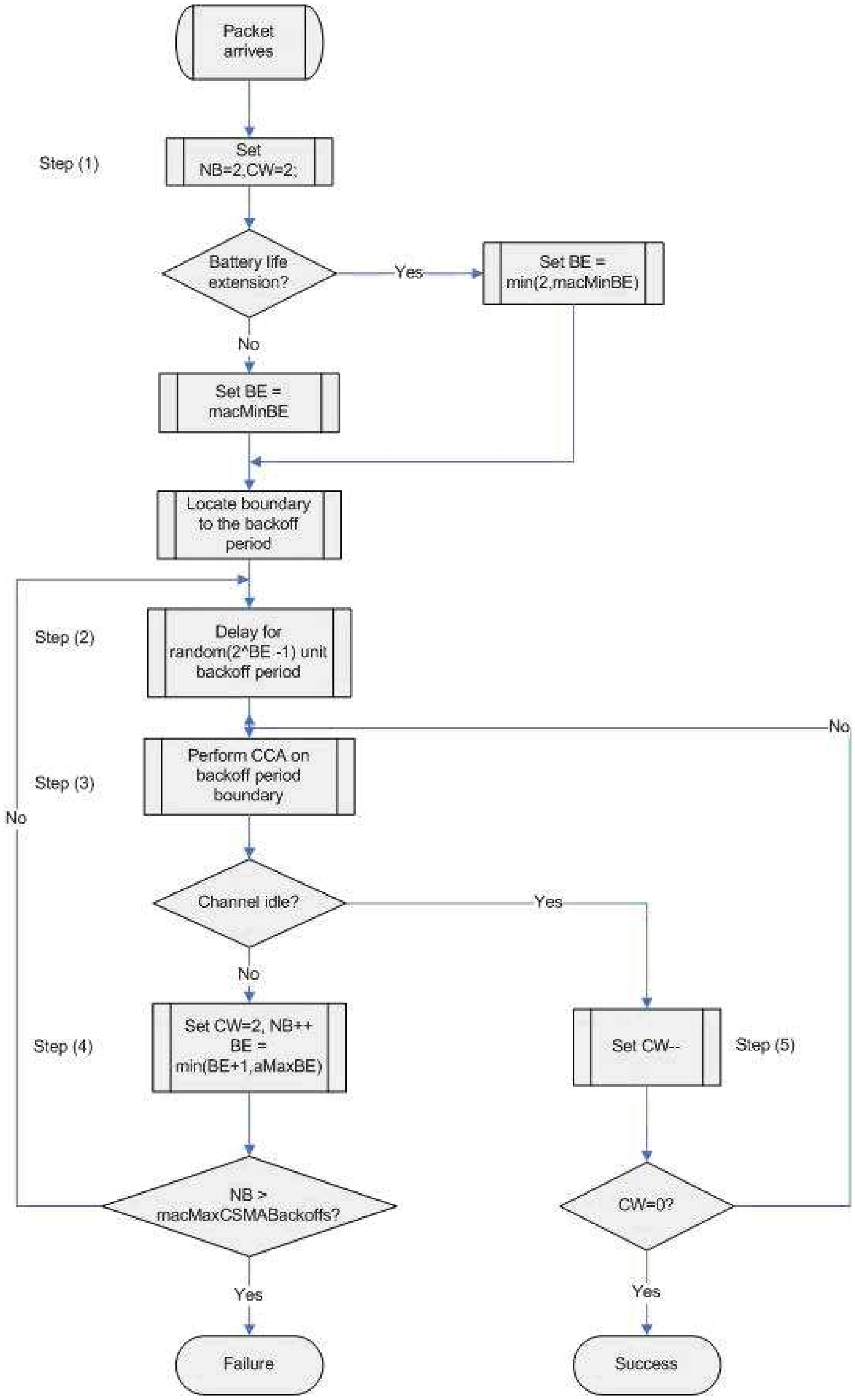


Figure 3.3: Beacon-enabled CSMA-CA algorithm in IEEE 802.15.4. (Adapted from [3])

Table 3.1: The list of constants used in this thesis

MAC constant	Description	Value
$aUnitBackoffPeriod$	The number of symbols comprise the basic time period used by the CSMA-CA algorithm	20
$aNumSuperframeSlots$	The number of slots contained in any superframe	16
$aBaseSlotDuration$	The number of symbols comprising a Superframe when the superframe order is equal to 0	60
$aBaseSuperframeDuration$	The number of symbols comprising a superframe When the superframe order is equal to 0	$aBaseSlotDuration$ * $aNumSuperframeSlots$=960
$aResponseWaitTime$	The maximum number of symbols a device shall Wait for a response command to be available Following a request command.	32 * $aBaseSuperframeDuration$
$aMaxBE$	The maximum number of the backoff exponent in the CSMA-CA algorithm.	5
$aGTSDescPersistenceTime$	The number of superframes that a GTS descriptor Exists in the beacon frame of a PAN coordinator.	4
$aMaxLostBeacons$	The number of consecutive lost beacons that will Cause the MAC sublayer of a receiving device to declare a loss of synchronization.	4

sure no device is currently transmitting. This procedure, referred to as Clear Channel Assessment (CCA), has to be done in two successive backoff periods, as shown by steps (3) and (5) in Fig. 3.3. If the channel is found busy at the second CCA, the algorithm simply repeats the two CCAs starting from step (3). However, if the channel is busy at the first CCA, the values of i and BE are increased by one, while c is reset to 2, and another random wait is initiated; this is step (4) in the flowchart. In this case, when the number of retries is less or equal to $macMaxCSMABackoffs$ (the default value of which is 5), the algorithm returns to step (2), otherwise it terminates with a channel access failure status. Failure will be reported to the higher protocol layers, which can then decide whether to re-attempt the transmission as a new packet or not. If both CCAs report that the channel is idle, packet transmission may begin. Before undertaking step (3), the algorithm checks whether the remaining time within the CAP area of the current superframe is sufficient to accommodate the CCAs, the data frame, the proper interframe spacing, and the acknowledgment. If this is the case, the algorithm proceeds with step (3); otherwise it will simply pause until the next superframe, and resume step (3) immediately after the beacon frame. Successful transmissions are optionally acknowledged by the recipient. The timing constraints dictated by the standard preclude the possibility of a collision between a regular packet and an acknowledgment.

3.2.2 GTS

Part of the active portion of the superframe (referred to as the contention-free period, or CFP) may be reserved for dedicated access by some devices. CFP may be divided to multiple slots, which is called Guaranteed Time Slots (GTS). A guaranteed time slot (GTS) is part of the superframe that is dedicated to a device and allows that device to use the time slots to communicate with the coordinator on the dedicated channel. GTS must be requested by a device and allocated by the coordinator before usage. If the GTS is not necessary any more, the device can request to cancel the GTS and the coordinator will de-allocate the GTS. The maximum number of GTS in one superframe is 7 and the minimum size of a GTS is one base slot.

If the device accesses the medium using guaranteed time slot (GTS), then this access scheme is a schedule-based access scheme. GTS can only be allocated by the PAN coordinator and the maximum number of GTS is 7, provided there is sufficient capacity in the superframe. When a device needs to request for a GTS to communicate with the coordinator, it can do so by sending request with GTS characteristics set according to the requirements of intended application. When the coordinator receives the request, it will check if it has sufficient bandwidth for the required GTS, noted that the coordinator shall preserve the minimum CAP length of $aMinCAPLength$. Once the GTS allocation is successful, the coordinator shall make this decision within $aGTSDescPersistenceTime$ superframes. According to the standard, $aGTSDescPersistenceTime$ is 4 superframes time. After receipt of the acknowledgment to the GTS request command, the device will track beacons and wait for at most 4 superframes time. If no GTS descriptor appears in this time period, then the allocation has failed. If there is sufficient bandwidth to allocating a GTS, then the coordinator allocates GTS in the superframe by First Come First Server (FCFS). Figure 3.4 shows the GTS allocating order, where the time order of allocation is GTS1, GTS2, and GTS3.

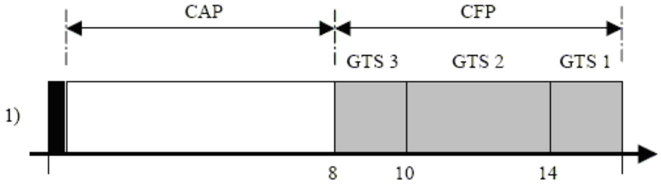


Figure 3.4: GTS allocation order in IEEE 802.15.4. [3]

3.2.3 Acknowledged and Non-acknowledged Data Exchange

The communication between sensors can be acknowledged or non-acknowledged communication. In acknowledged communication, an acknowledge will always follow the packet transmission within a gap of T_{ack}. MAC layer also needs sometime to process data received by the PHY layer. Therefore transmitted frames shall be followed by an inter-frame spacing period. The length of the inter-frame spacing period depends on the size of the frame that has just been transmitted. Frames of up to $aMaxSIFSFFrameSize$ in length shall be followed by a short inter-frame spacing (SIFS) period of a duration of at least $aMinSIFSPeriod$ symbols. Frames longer than $aMaxSIFSFFrameSize$ shall be followed by a long inter-frame spacing (LIFS) of a duration of at least $aMinLIFSPeriod$ symbols. Figure 3.5 illustrates the packets inter-frame spacing.

3.3 Bridge scheduling algorithm

The basic feature of a Master/Slave bridge is that the coordinator of a cluster (bridge) collect data from it own piconet and switch to a foreign piconet and act as

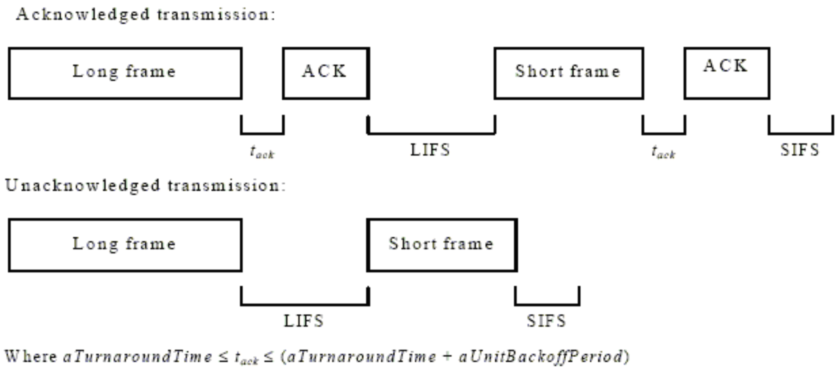


Figure 3.5: Inter-frame spacing in acknowledged and non-Acknowledged data exchange in IEEE 802.15.4. [3]

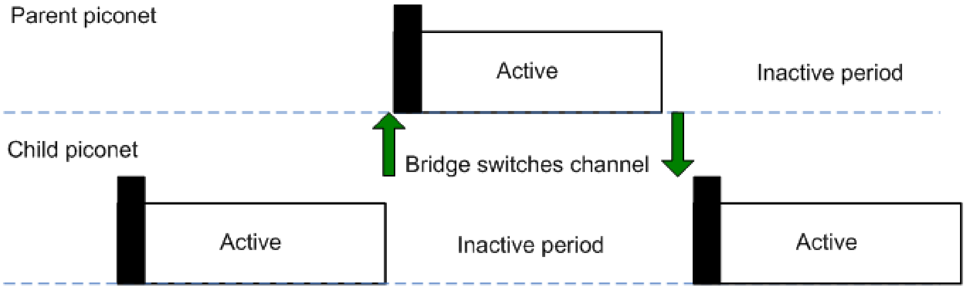


Figure 3.6: Bridge scheduling

a ordinary device to pass the collected data to the coordinator in the other cluster. This requires the bridge to share the working time between two piconets. The optional inactive period of the beacon cycle makes it easy for a coordinator to leave the home piconet and engages in another piconet to pass data to another piconet. Figure 3.6 illustrates the bridge scheduling method. The bridge in this example is the coordinator of the child cluster illustrated in Figure 3.1. Noted that the superframe period is the same in both parent and child piconet.

In this scheme, the coordinator/bridge has to buffer all the data it receives from the sensor nodes in the child cluster before it can upload it to the coordinator of the parent cluster (i.e., the network sink). Obviously, the size of that buffer is a critical parameter. Small buffers may lead to excessive packet blocking, which will in turn affect the quality of information received by the sensing application.

It should be noted that the bridge acts just like an ordinary node in the parent cluster. This means that a countdown that exceeds the active portion of the parent cluster superframe will be frozen during the inactive portion of the said superframe, and resumed in the next one. It also means that the bridge will compete for access to the medium with ordinary nodes in the parent cluster. As the bridge traffic originates from the entire child cluster, there is potentially significant contention in the parent cluster; a more detailed analysis of this contention is given below.

3.4 Simulation results and analysis

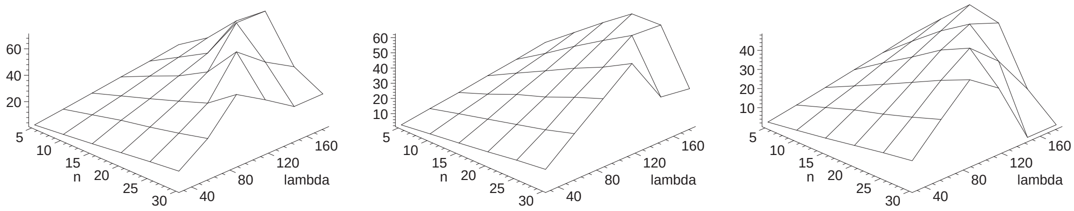
The following simulation results are based on my two piconets simulation model using Petri net [1] based platform Artifex 4.1 [12]. I developed the simulation model of acknowledged transmission with CSMA-CA protocol, acknowledged GTS and non-acknowledged CSMA-CA. My supervisor Dr. J. Misic created mathematical model for both CSMA-CA and GTS model. Our results match very well. Our publications for interconnection communication can be found in [19], [18] and [17].

To investigate the performance of the network, I assumed that the network operates with raw data rate of 250kbps in the 2.4GHz band. I also assume the child cluster has the same number of nodes with the parent clusters. The nodes number varies from 5 and 30. The superframe size in both clusters was controlled with SO = 0, BO = 1. Then the superframes with active and inactive portions take 48 backoff periods. The packet arrival rate to each node varies between 0.5 and 3 packets per second, and the data packet size was 3 backoff periods. The acknowledgment packet size is 2 backoff periods. The acknowledgment turn around time is 1 backoff period. Ordinary nodes have buffer size L = 2 packets while the coordinator/bridge has buffer size Lb = 6 packets. In all the simulations, I assume the bit error rate is 10^{-4}.

3.4.1 Bridge performance using acknowledged CSMA-CA

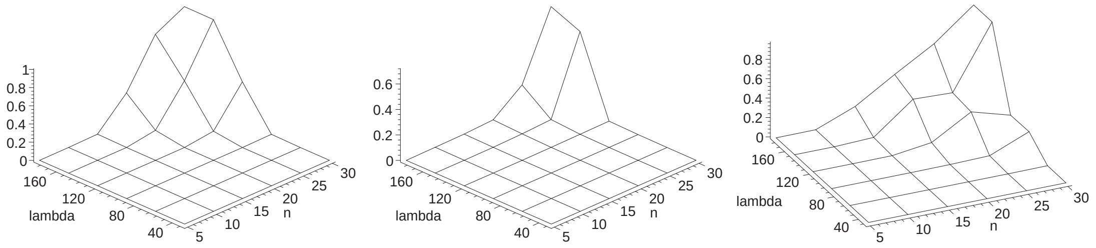
When bridge uses CSMA-CA protocol with acknowledgement to transmit packets, we can see that initially the throughput of source nodes to bridge, sink nodes to sink and source nodes to sink increase quickly when the number of nodes and packet arrival rate increase. When the traffic load is high enough, the throughput drops down dramatically. We call this situation as saturation mode. This happens because the media is almost fully occupied and the additional traffic will only collide with other packets and the number of non-collided packets will decrease. Comparing the throughput of source and sink cluster, we can notice that source cluster reaches saturation mode a little earlier than sink cluster. This is because the bridge has limited capacity and it has to drop some packets when its buffer is full. Nodes have to retransmit the dropped packets since this is acknowledged transmission.

When we compare the packets drop rate on nodes in source cluster, nodes in sink cluster and bridge, we can see that the source cluster drop rate increases faster than the sink cluster. The packets drop rate on bridge is the highest of all three. We can also notice that the bridge drop rate curves are not smooth. The increment of drop rate slows down in the middle and increases fast again. This is because bridge node



(a) Packets rate from souce cluster to the bridge (b) Packets rate from sink cluster to the sink coordinator (c) Packets rate from the source cluster to the sink coordinator

Figure 3.7: Event sensing reliabilities for source, sink and bridge when bridge uses acknowledged CSMA-CA protocol



(a) Drop-rate of an ordinary node in the source cluster (b) Drop-rate of an ordinary node in sink cluster (c) Drop-rate of bridge

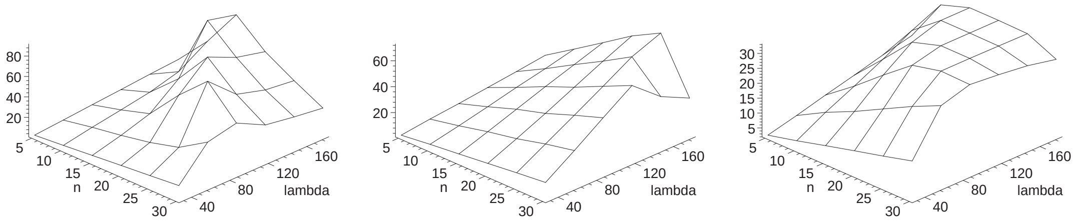
Figure 3.8: Packet drop-rate for source, sink and bridge when bridge uses acknowledged CSMA-CA protocol

can not deliver all the packets from source to the sink, then the repeating of packets transmission leads the source to reach saturation first. The non-collided packets reach the bridge drop down quickly with the increment of traffic while the packets pass bridge to the sink increase when the sink is not in saturation. Therefore the drop rate result has a bump in the middle. When the sink cluster reaches the saturation mode, the drop rate of bridge increases quickly again.

3.4.2 Bridge performance using acknowledged GTS

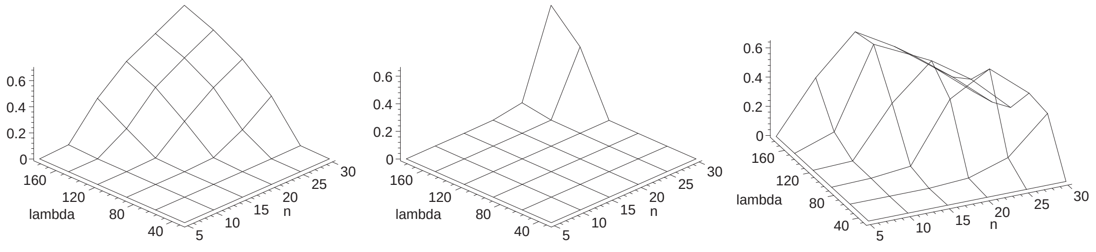
When bridge uses acknowledged GTS to transmit the packets to the sink, then all the packets from source cluster will go to the sink through reserved bandwidth and the traffic in source cluster will not affect the performance in sink cluster. We can expect that the maximum packets rate through bridge is determined by the width of the reserved bandwidth. Figure 3.9(a), Figure 3.9(b) and Figure 3.9(c) are the simulation results of data packet rate in source cluster, sink cluster and bridge where the reserved time slot for bridge is 2 base slot (6 backoffs). This reserved lane allows maximum of 32 packets to be passed to the sink. We can see that the source cluster reaches saturation mode quickly after the arrival traffic rate is over 32pkt/sec. The bridge packets rate increases with arrival traffic when it is less than 32pkt/sec. After that, the data rate through bridge will stay in 32pkt/sec until source cluster get into deep saturation when the offered packet rate to the bridge degrade to under 32pkt/sec. We can see that in Figure 3.9(c), data rate through bridge drop down when the number of nodes reaches 30 and arrival rate per node reaches 3pkt/sec.

The drop rate on source nodes increases dramatically when the arrival traffic is larger than 32pkt/sec. The drop rate on the bridge node increases dramatically before the source cluster is in saturation mode. After the source cluster is in saturation, the drop rate on the bridge node decreases since the offered data rate from the source cluster decreases. The sink cluster performance degrades much later than the source cluster and traffic mode in the source cluster will not affect the sink cluster.



(a) Packets rate from source cluster to the bridge (b) Packets rate from sink cluster to the sink coordinator (c) Packets rate from the source cluster to the sink coordinator

Figure 3.9: Event sensing reliabilities for source, sink and bridge when bridge uses acknowledged GTS protocol

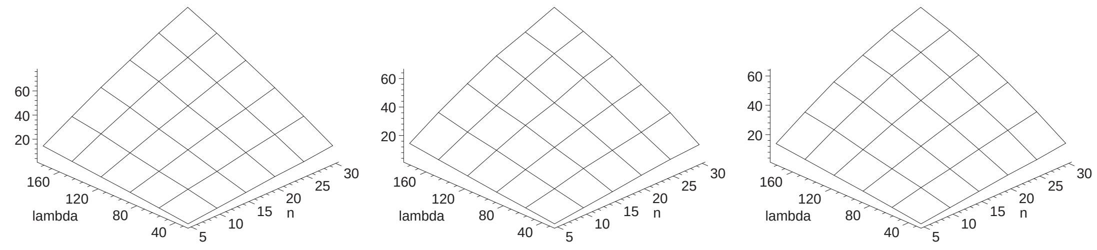


(a) Drop-rate of a ordinary node in souce cluster (b) Drop-rate of a ordinary node in sink cluster (c) Drop-rate of bridge

Figure 3.10: Packet drop-rate for source, sink and bridge when bridge uses acknowledged GTS protocol

3.4.3 Bridge performance using non-acknowledged CSMA-CA

In non-acknowledged transmission, packets can be lost due to noise, blocking or collision. However, this is acceptable in sensor network since the contents for different close-by nodes are correlated. We observed that with the non-acknowledged transfer with CSMA bridge, the packet drop rate on both clusters and bridge are

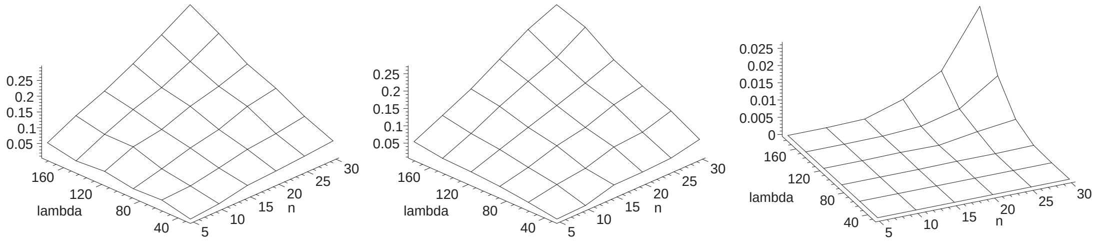


(a) Packets rate from souce cluster to the bridge

(b) Packets rate from sink cluster to the sink coordinator

(c) Packets rate from the source cluster to the sink coordinator

Figure 3.11: Event sensing reliabilities for source, sink and bridge when bridge uses non-acknowledged CSMA-CA protocol



(a) Drop-rate of a ordinary node in source cluster

(b) Drop-rate of a ordinary node in sink cluster

(c) Drop-rate of bridge

Figure 3.12: Packet drop-rate for source, sink and bridge when bridge uses non-acknowledged CSMA-CA protocol

lower compared to the acknowledged transfer with CSMA bridge, and the network can endure higher offered load. This is because in non-acknowledged transfer, only needs 3 backoff periods are needed to transfer one packet, which is smaller than 6 backoff period in acknowledged transfer.

3.5 Conclusion

In this chapter I have investigated and compared the performance of two interconnected clusters model using master-slave bridge in IEEE 802.15.4 beacon enabled network. I have considered the situation of bridge using acknowledged CSMA-CA protocol, non-acknowledged CSMA-CA protocol and acknowledged GTS. I compared the simulation results of the network performance in the above three situations.

First we compare the acknowledged CSMA-CA transfer with the GTS transfer. Results show that, under low packets arrival rate, CSMA-CA mode outperforms the one-lane GTS mode since more packets can be delivered to the sink in a single superframe. However, at moderate high load, the bridge under GTS mode performs much better than CSMA-CA mode. In GTS mode, the maximum data rate of source cluster is controlled by the width of the GTS slot. In CSMA-CA mode, it is controlled by sink traffic load, which is harder to predict compared to the GTS slot width.

Then we compare the performance of acknowledged CSMA-CA transfer and non-acknowledged CSMA-CA transfer. The results show that non-acknowledged CSMA-CA transfer can endure higher traffic load than acknowledged CSMA-CA transfer. This is because non-acknowledged CSMA-CA transfer takes less media bandwidth than acknowledged CSMA-CA transfer. If the sensor network can endure unreliable data communication then non-acknowledged CSMA-CA is the better choice.

The results of this chapter can provide useful information in choosing parameters when developing a sensor network application.

Chapter 4

Duty-Cycle and Node Population Compensation

4.1 Introduction

Many applications, especially those for environment surveillance purposes, require the sensor network to maintain reliable event sensing from the area. We can define ”event sensing reliability” as in terms of the minimum data rate that a cluster coordinator must receive from the sensors in the cluster. In a cluster-based network, a cluster represents a physical area. Each cluster should provide the minimum required event sensing reliability which can be determined by the physical area of the cluster. A network is considered to be in normal working situation if the event sensing reliability quality for each cluster can be achieved. If the one of the clusters is no longer able to provide the required event sensing reliability due to the lack of living nodes then we say the network is working in an impaired mode. Therefore, we define the lifetime of a cluster-based network to be the network working period that all the clusters can achieve their minimum event sensing reliability. Since nodes are battery powered in sensor networks, their power consumption should be minimized to achieve the longest lifetime of the network. A common way to do this is to put sensors in the sleeping mode periodically to save energy. In a cluster-based sensor network, we can place redundant nodes in clusters. When the number of sensors is higher than the minimum necessary to achieve the desired data rate, it allows individual sensors to switch between active mode and sleeping mode alternatively. The reduction of the duty cycle leads to longer network lifetime [14]. In order to lay the nodes economically, each cluster should have the right number of nodes that the lifetime of all the clusters are equalized. In the work of this chapter, we will investigate the problem of how many redundant nodes we should put in each cluster to equalize the cluster lifetime.

In a cluster-based multilevel sensor network, the power consumption of the nodes on different levels will differ - so will their lifetime. In a simple case of the two-level network model, if the devices at one level exhaust their batteries before those at the other, the network will die with some energy left. Ideally, we expect the two clusters die at the same time so that the leftover energy will be minimal.

In this chapter, I will address this problem in the context of a sensor network based on the IEEE 802.15.4 communication standard for low rate wireless personal area networks (LRWPANs) [3]. The network is composed of two clusters controlled by separate cluster (PAN) coordinators. The communication in both clusters are in beacon enabled, slotted CSMA-CA mode. Clusters are interconnected by a master-slave bridge [19]. The coordinator of one cluster (referred to as the child cluster) acting as the bridge. The coordinator/bridge in the child cluster visits the parent cluster during the inactive portion of the superframe and delivers its data to the coordinator, which acts as the network sink. The bridge uses CSMA-CA for communication in the parent cluster, which means that it has to compete with the ordinary nodes in that cluster.

I assume that both coordinators are main powered and individual sensor nodes are battery operated, and their battery capacity is expressed as a budget of b backoff cycles. I also assume that the power consumption is the same for transmission and reception. I made this assumption because this is a good approximation in activity-managed 802.15.4 sensor networks where the transmission mode dominates - the reception mode is used only to receive beacon and acknowledgement frames. I put redundant nodes in both clusters so it enables the reduction of individual sensor duty cycle through activity management [24]. Under the duty cycle management, each node spends most of its time in sleeping mode and wakes up only to transmit one packet. The goal is to maximize the network lifetime while the required data rate R is maintained at the network sink. This data rate is split evenly between the two clusters (R/2 for each cluster). I assume that the cluster dies when the number of alive nodes is not sufficient to maintain the required data rate. My supervisor Dr. J.Misic developed a mathematical model to calculate the number of redundant nodes under the assumptions that (a) each cluster is lightly loaded and (b) each sensor node autonomously determines its sleep time using the information about the required aggregate event sensing reliability and the number of live nodes obtained from the cluster coordinator. I developed a simulation model to observe and verify her analytical results.

The rest of this chapter is organized as follow. In Section 2 I will briefly discuss bridge operation on ordinary nodes to perform activity management. In Section 3 I will briefly go over my supervisor's mathematical model and results. In Section 4, I will present my simulation results and evaluate the results. Section 5 concludes the chapter.

4.2 Duty cycle techniques in each cluster

At first, we look at the case of the child and parent cluster contain are the same size with n child nodes and n parent nodes respectively. The packet arrival rate is Λ_i for ordinary sensor nodes. Each node has a finite buffer with the capacity of L packets, while the bridge buffer capacity is L_{bri}. The nodes sleep period follows Geometric distribution and the average sleeping time depends on requested data rate per cluster and total number of nodes in the cluster. The algorithm for nodes is the following: Every time when a node wakes up from sleeping mode, it will check if there is any packet in the buffer. If there is no packet then it goes back to sleep right away. If there is any packet then it will immediately turn on the receiver and wait for the next beacon from the coordinator. When beacon comes it will wait for a random backoff period (randomly chosen from 0 to 7 to avoid collision) and send out only one packet to the coordinator using CSMA-CA protocol. After finishing transmitting the packet and receiving the acknowledgement, the node will turn back into sleeping mode again. The duration of this time period is determined according to geometric distribution. Equation 4.1 is used to calculate the average sleeping time,

$$\frac{R}{n} = \frac{1}{T_{sleep} + T_{service}} \approx \frac{1}{T_{sleep}} \tag{4.1}$$

where T_{sleep} is the average node sleeping time and $T_{service}$ is the average packet service time. In a real sensor network, nodes should be put in sleeping mode at least 99.5% of the time in order to achieve the lifetime of one year or more. In this situation the node active time is very small compared to the node sleeping time and it is negligible.

In my simulation model, the R is chosen to be 20pkt/sec and n is chosen to be 100. In this case, the initial average sleeping time for each node is 5 seconds. However, the simulation result showed that the event sensing reliability is lower than expected. In order to make the model to control the event sensing reliability more accurate, I proposed to added in the following factor, G_c, which represents the probability that when nodes wake up from sleeping mode while there is no packet in the buffer (this is not negligible since the device buffer size is only 2). In my simulation model, the nodes will collect samples periodically and estimate the value of G_c using the exponential moving average method, i.e., $G_{ck} = \alpha Sample_k + (1-\alpha)G_{ck-1}$ and $Sample_k = N_0/N$, where N_0 is the number of times that buffer is empty when node wakes up and N is the total times nodes wake up. The parameters are chosen as follow: sample period is 100 beacon cycles and α is chosen to be 0.125. Then the nodes sleeping time equation is modified to be as follows (equation 4.2):

$$\frac{R}{n} = \frac{1 - G_c}{T_{sleep}} \tag{4.2}$$

The cluster coordinator periodically broadcasts the required event sensing reliability per cluster R and the number of associated alive nodes n_{chi} or n_{par}. Nodes use this information to calculate the mean duration of period between node transmissions using equation 4.2, respectively.

4.3 Analytical model to calculate the number of node in clusters

This section is subtracted from [20].

In order to reduce the computation complexity of the analytical model, Dr. J.Misic proposed a computationally lightweight technique to compute the number of nodes to equalized the utilization of nodes on both clusters and therefore equalize the lifetime of the two clusters. The first assumption of this model is, the required event sensing reliability per cluster R is much lower than the capacity of the cluster, which means that at any given time only a small fraction of nodes is active. The mean number of packets sent by the bridge in a single superframe can be approximated with $N = R \cdot BI \cdot t_{boff}$, where R is the desired event sensing reliability(expressed in packets per second), the beacon interval BI includes both active and inactive portion of the superframe, and t_{boff} denotes the backoff period that lasts for $0.32ms$. Assuming the event sensing reliability of R = 20 packets per second per cluster, the mean number of packets sent by the bridge per superframe is $N = 20 \cdot 96 \cdot 0.00032 \approx 0.6144$; the same amount of traffic is sent by all the nodes in the parent cluster.

Mean duration of period between node transmissions in backoffs, as determined by each node, as $B_i = n/(t_{boff}R)$, where n is the number of nodes in a particular cluster and R is the required event sensing reliability (appropriate subscript should be used to indicate the cluster in question). Let Q_c be the probability that a node wakes up with empty buffer and let f_k stand for the probability that k packets will arrive to the node buffer during one sleep period. Then we have $Q_c = \frac{f_0}{1-f_0}$. The probability f_0 of zero packet arrivals during the sleep period can be found as follows. First, the moment generating function for the sleep period is,

$$V^*(s) = \sum_{k=1}^{\infty}(1-P_{sleep})P_{sleep}^{k-1}e^{-sk} = \frac{(1-P_{sleep})e^{-s}}{1-e^{-s}P_{sleep}} \tag{4.3}$$

The Probability Generating Function (PGF) for the number of packet arrivals to the buffer during the sleep period can be found as $F(z) = V^*(\lambda_i - z\lambda_i)$ and $f_0 = V^*(\lambda_i) = (1-P_{sleep})e^{-\lambda_i}/(1-P_{sleep}e^{-\lambda_i})$. The probability distribution for the total inactive time of the node has a geometric distribution with the parameter Q_c, applied at the moments when the node returns from sleep. The corresponding moment generating function is,

$$I^*(s) = \sum_{k=1}^{\infty}(1-Q_c)Q_c^{k-1}V^*(s)^k = \frac{(1-Q_c)V^*(s)}{1-V^*(s)Q_c} \tag{4.4}$$

and the mean value is $\overline{I} = 1/((1-Q_c)(1-P_{sleep}))$.

Finally, by equating the average period between the transmissions with the average inactive time, we obtain

$$B_i = \overline{I} \tag{4.5}$$

which can be solved for P_{sleep} using the number of nodes in the cluster, the required sensing reliability, and packet arrival rate per node as independent variables.

A node that wakes up has to wait for the beacon for synchronization. As there may be more than one node in this mode, increased collisions may result for the packets sent immediately after the beacon. To avoid this, we introduce an additional waiting time, the duration of which is uniformly distributed in the range 0..7 backoff periods [23]. The PGFs for these two synchronization functions are $S_1(z) = (1/BI)\sum_{i=0}^{BI} z^i$ and $S_2(z) = \frac{1}{8}\sum_{i=0}^{7} z^i$, respectively.

Given the battery budget of b backoff periods, the average number of transmission/sleep cycles in a cluster can be found as $\left\lceil \frac{b}{\overline{S_1} + \overline{S_2} + \overline{T_x}} \right\rceil$, where the proper cluster label should be substituted for x.

Given the law of large numbers [11], the PGF for total lifetime of the node in cluster x becomes $L_x(z) = (S_1(z)S_2(z)T_x(z)I_x(z))^{n_{c,x}}$. By differentiating the respective PGFs we can obtain the standard deviation of the node lifetime as well as the coefficient of skewness, μ, which measures the deviation of a distribution from symmetry: values that are clearly different from 0 indicate that the distribution is asymmetrical around the mean.

4.4 Simulation Results

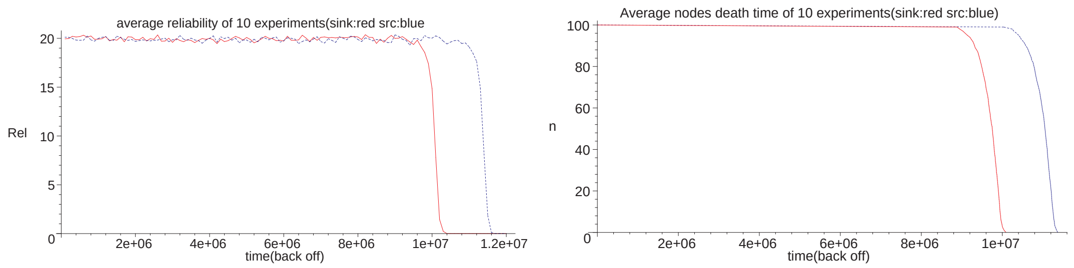
In order to verify the analytical model for distributed calculation of the sleep interval, for finding the parent population, and for determining the lifetime of the network, I have developed a simulation model of a two-cluster 802.15.4 network using the object-oriented Petri net simulation engine Artifex by RSoft design, Inc. [12]. In this model, two clusters are interconnected by the master-slave bridge. The coordinator of the child cluster switches to parent cluster periodically to send data to the sink. The sensors in both clusters perform duty cycles and turn in to sleep mode periodically. The network is assumed to operate in the ISM band at 2.4 GHz, with the maximum raw data rate of 250kbps. The child cluster had one coordinator and 100 ordinary sensor nodes. I assume the packet arrivals to each node follow the Poisson distribution and the average data rate is 1 packet per second. I assume the packet size is fixed at 30 bytes, including all PHY and MAC layer headers. The buffer size for an Ordinary node is L = 1 packet. The buffer size for the bridge is $L_{bri} = 6$ packets. The prescribed event sensing reliability in each cluster was 20 packets per second; in this case, the traffic load in the parent cluster will be 40 packets per second. The battery budget for each ordinary node was set to be 50, 000 backoff periods. When the radio subsystem was turned on, the remaining backoff periods will count down for each backoff period.

Table 4.1: Initial network parameters, taken from [20]

parameter	child cluster	parent cluster
number of nodes	100	100
inactive period	15624.98	15624.98
utilization	0.00427	0.00466
lifetime	1.1886E7	1.0763E7
std. deviation	4.260%	4.35%
skewness μ	0.00018	0.00018

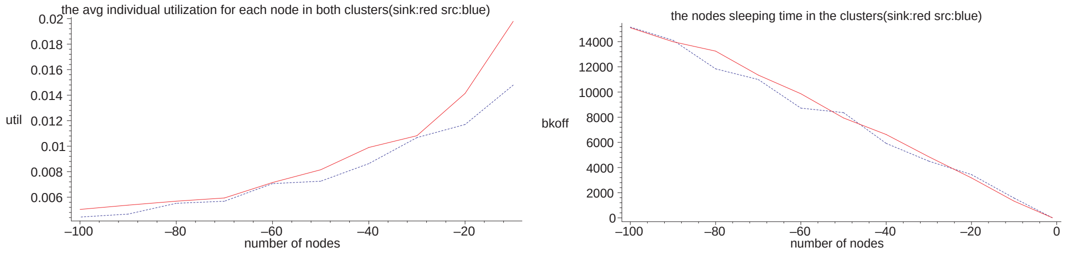
4.4.1 Maintaining the required event sensing reliability

In my first experiment, I investigated the performance of the duty cycle management algorithm only. I put 100 nodes in both parent and child cluster, and calculated the initial network parameters; the calculated values that describe the state of the



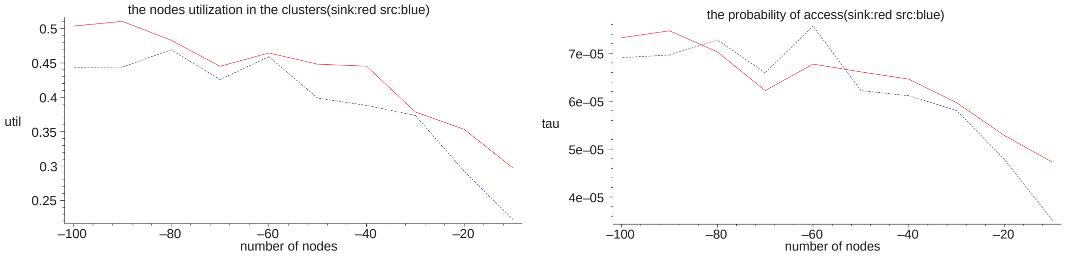
(a) Event sensing reliability (data rate) per cluster.

(b) Number of live nodes.



(c) Individual node utilization.

(d) Mean duration of sleep period.



(e) Aggregate(Per-cluster) node utilization.

(f) Access probability.

Figure 4.1: Network performance with initial population of 100 nodes in both clusters. Red, blue denotes sink and source cluster, respectively.

network are given in Table 4.1. As can be seen, the standard deviation is small, only about 4% of the mean, and the skewness is close to zero; this means that all nodes in a cluster will die within a short interval centered on the mean lifetime value for that cluster.

My simulation results are shown in Fig. 4.1. Figure 4.1(a) shows the event sensing reliability of child cluster and parent cluster. The X-axis corresponds to the time in backoff periods and the Y-axis corresponds to the average event sensing reliability from each cluster in the sampling periods. We can see that the sleep management algorithm manages to maintain the required event sensing reliability for quite a long time; this holds for both clusters. Figure 4.1(b) shows the nodes death time of both clusters. The X-axis corresponds to the time in backoff periods and the Y-axis corresponds to the number of living nodes in the cluster. We can see that when two clusters have the same number of nodes, the parent cluster will be exhausted first. We should put more nodes in the parent cluster. We can also see that the death of all the nodes in a cluster happens around the same time. This confirms the analytical results.

When sensor nodes start to die, the rest of the nodes will need to work more frequently in order to maintain the prescribed event sensing reliability. Therefore the mean utilization of the ordinary nodes will increase while the mean sleep time will decrease. When the number of living nodes decreases, the cluster's aggregated utilization decreases slowly, which means the system is trying to keep the total working afford of the sensors in the cluster. Note that the utilization is somewhat higher in the parent cluster since the traffic load is higher there. This is because the bridge competes with other nodes in the parent cluster, and the collision probability is higher. Consequently, the success probability for ordinary nodes in parent cluster is lower and the average number of transmission attempts per packet is higher. This leads to higher utilization and higher mean number of active nodes in the parent cluster.

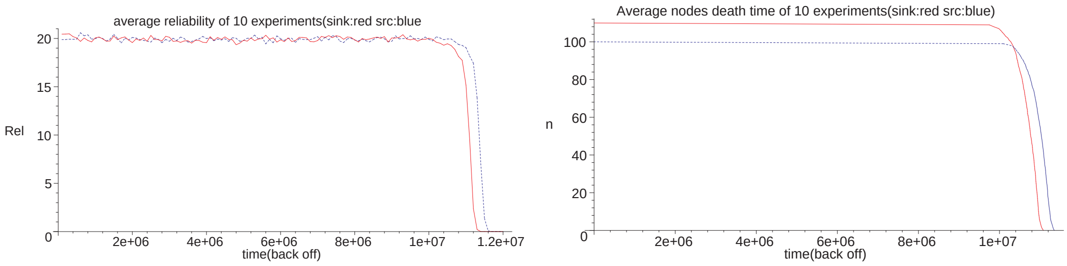
When there are only about twenty nodes (or less) left, the aggregated utilization starts to decline steeply. Due to the reduced number of nodes and short sleep times, the access probabilities decrease as well; this means more collisions leads the nodes to use up their battery faster. In this situation, the parent cluster is incapable of sustaining the required event sensing reliability and the network ceases to operate when parent cluster is exhausted; on account of its higher traffic load. The lifetime of the parent cluster is shorter than the lifetime of the child cluster by about 15%. The solution of this is to increase the redundant nodes in parent cluster.

4.4.2 Equalizing lifetime among the clusters

Table 4.2: Initial network parameters revised, taken from [20]

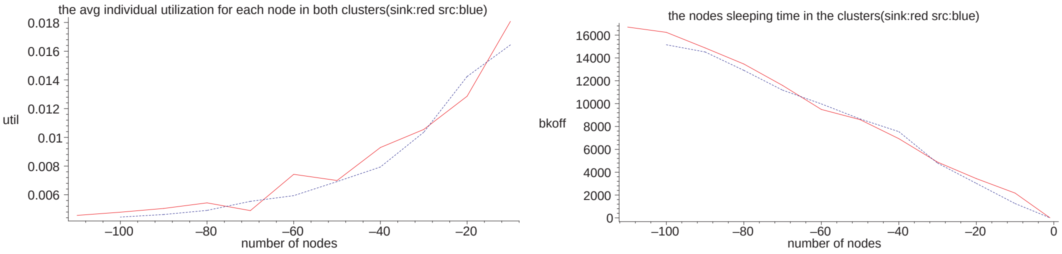
parameter	child cluster	parent cluster
number of nodes	100	110
inactive period	15624.98	17031.31
utilization	0.00427	0.00427
lifetime	1.1886E7	1.1886E7
std. deviation	4.26%	3.51%
skewness μ	0.00018	0.00042

In my second experiment, I have adjusted the number of nodes in the parent cluster according to the analytical result. I put 100 nodes in the child cluster, and then I put 110 nodes in the parent cluster (the analytical result is 109.81 and I took the closest interger) in order to prolong the lifetime of parent cluster to be the same as the child cluster. The calculated network parameters for both clusters as shown in Table 4.2. Figure 4.2(a) shows the maintained event sensing reliabilities in both clusters. After nodes adjusting, the event sensing reliabilities still satisfies the



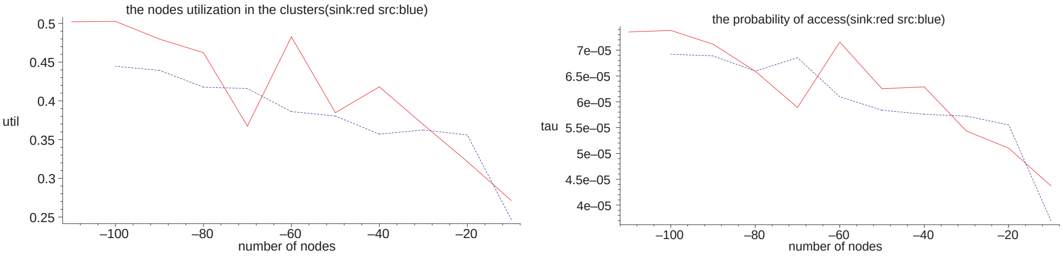
(a) Event sensing reliability (data rate) per cluster.

(b) Number of live nodes.



(c) Individual node utilization.

(d) Mean duration of sleep period.



(e) Aggregate(Per-cluster) node utilization.

(f) Access probability.

Figure 4.2: Cluster performance with initial population of 100 in source cluster and 110 in sink cluster. Red, blue denotes sink and source cluster, respectively.

prescribed requirement. Figure 4.2(b) shows the nodes death time of both clusters after population adjustment. We can see that after the population adjustment, the lifetime of the two clusters is much closer than the non-adjustment case. This is because the parent cluster has more redundant nodes and the average utilization of the nodes decreases and therefore the network lifetime is extended.

We can see that, the number of live nodes is initially larger in the parent cluster; once parent cluster nodes number begins to drop, the child cluster follows soon. Individual node utilization is again higher in the parent cluster, due to the higher competition in the parent cluster. Individual node utilization in the parent cluster is lower than in the case where the parent cluster has 100 nodes. However, aggregate node utilization is about the same. This is because the required event sensing reliability is to be contributed by the entire cluster, regardless of the number of nodes.

The large variations of the aggregate node utilization in Figure 4.1(c) are caused by the short intervals between individual node deaths. In this case, when the number of live nodes is still high (between hundred and forty), several nodes happen to die in a short period while most of the others are still sleeping. As a result, the aggregate utilization exhibits a sharp drop. When the sleeping nodes awake, they adjust their sleep periods to compensate for the reduction in event sensing reliability, and the aggregate utilization rise up.

4.5 Conclusion

In a multi-cluster network, clusters close to the sink experience higher traffic load than the others. This causes the fact that the nodes in those clusters use up their power first and the lifetime of the clusters varies. In this chapter, I have introduced a cluster population compensation method to balance the lifetime of the clusters in the network. Simplified analytical expressions for the modeling of interconnected 802.15.4 clusters were developed by my supervisor and were briefly presented here. In order to verify the correctness of the analytical model, I developed a simulation model which uses activity management to control the event sensing reliability and observe the lifetime and other interested outputs of network. The simulation results confirm the analytical results and the clusters lifetime are equalized.

Chapter 5

Routing Algorithm for Balancing Network Lifetime

5.1 Introduction

To optimize the performance of a large-scale sensor network, a critical problem is to find a good routing algorithm. A good sensor network routing algorithm should allow the network to survive for longer time with pre-defined power budget. In a scatternet-based sensor network, the lifetime of the network can be defined as the minimum lifetime of all the clusters. In this chapter, I will propose a routing algorithm for cluster-based sensor network to balance the lifetime of all the clusters in the same layer. Simulation results show that this routing algorithm can effectively extend the lifetime of the sensor network. Finally I will present the routing algorithm in a large-scale 7 clusters sensor network and investigate the performance of such a sensor network.

In a cluster-based sensor scatternet, clusters are connected by intermediate nodes - bridges. A bridge collects data from the local sensors and periodically visits the clusters which are closer in hops to the sink, and send the data out using CSMA-CA protocol. In this situation, if there are multiple clusters to choose from, a coordinator needs to decide which cluster to visit. The criterion to decide the routing paths varies, such as minimum hops, minimum power consumption and minimum drain rate, etc. However, the goals are the same - extending the lifetime of the network as much as possible.

A cluster-based sensor network reduces the complexity of the network much since ordinary nodes do not need to maintain a big routing table. A cluster-based sensor network also allows sensors to be free of routing duties so they can put themselves into the sleep mode periodically. In addition, data integration is made much easier in cluster-based network. Howerver, existing sensor network routing algorithms either ignore the collisions in MAC layer or are designed for peer-to-peer networks. In my case, I need to find a routing algorithm that is suitable for cluster-based network and the MAC collisions are considered. Therefore, I propose a cluster-based power aware lifetime balance routing algorithm. In this algorithm, ordinary sensors will sleep periodically to save energy and send data to the coordinators when they wake up. Coordinators will collect data from ordinary sensors and send data to the next hop which is closer to the sink. If there are multiple choices of the next hop, data rate to each path will be justified so that the lifetimes of all the paths are balanced as much as possible. The lifetime of a path is defined to be the minimum cluster lifetime on the path. The cluster lifetime is calculated by $\frac{RBL_i}{DR_i}$, where RBL_i is the residual battery lifetime of the ith cluster and DR_i is the statistical battery drain rate of the cluster. The lifetime of a path is calculated by taking the minimum lifetime of all the clusters on the path. Each coordinator compares its own lifetime with the path lifetime from its parents and broadcasts the minimum value to its children. Each cluster is aware of the lifetime bottle neck of all the paths and it will use the flow adjusting algorithm I describe next to adjust the flows to all the paths so that the lifetime of all the paths be balanced as much as possible. I simulated the above routing algorithm on a star topology based IEEE 802.15.4 compliant network. The simulation results show that when using this algorithm, the data flows in the network will converge to balanced values and the lifetime of each cluster will be extended.

5.2 Network design and routing algorithm

5.2.1 Network model structure

In this system, sensors are grouped into clusters using the star topology [3]. The lifetime of the network is determined by the sensors traffic load and the sensor power consumption rate. The power consumption for devices is determined by the workload in the cluster.

The interconnection of clusters is implemented using Master/Slave bridge [19], where a coordinator will act as an intermediate node and passes data from the local cluster to the neighbour cluster. The communication protocol inside a cluster and between clusters is the collision-based CSMA-CA, which is specified in IEEE 802.15.4 [3].

The next two sections will introduce a sensors layout method and a communication channel assignment policy.

5.2.2 Layout model for sensors

For an environmental monitory sensor network, events may be sampled evenly in the field and be sent to the sink. In IEEE 802.15.4, the communication range of a sensor is $R = 10m$ or above. Therefore, I assume the transmission range of an ordinary sensor is tuned to be 10 meters and any sensor in this range can receive signals effectively. I also assume coordinators can have longer transmission range. Therefore, I can layout coordinators so that all the sensors in the feild can find at least one coordinator in its communication range. The simplest way is to divide the filed into cells and let coordinators lie in the center of the cells. Ordinary sensors are deployed in the field around coordinators. I chose to divide the field into equal sized hexagon cells with 10 meter to be the furthest distance from the hexagon center to any point in the cell. A coordinator should lie in the center of each cell. Hexagon cells were proved to be the better choice compared to squares and equilateral triangles [26]. All the sensors in one hexagon cell is in the communication range of the coordinator which lies in the centre of the hexagon. All the sensors in one cell will form one cluster. Coordinators collect data in the cluster and send the packets out to the neighbour clusters which are closer in hops to the sink.

Now all the sensors have their closest coordinator in their communication range. The cluster formation is based on the closest distance method. However, the coordinator in the center of the hexagon should be able to communicate with the coordinators

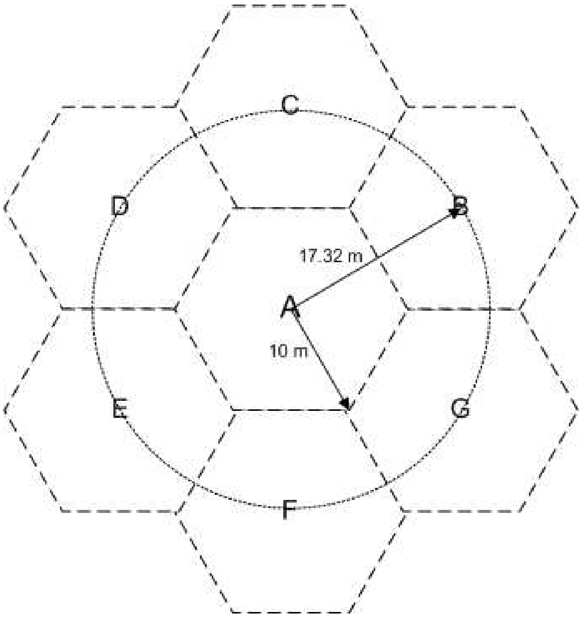


Figure 5.1: Communication range of normal sensors and coordinators

in the neighbour clusters, so its transmission range should be doubled to be higher. A example is shown in figure 5.1. In cell A, 10 meter is the distance from the coordinator to the furthest sensors in the cell. That is, all the sensors in this cell can reach the coordinator in their transmission range. The transmission range of a coordinator is $R_{bri} = 17.32m$. This is far enough to reach the coordinators in the neighbour clusters coordinators.

5.2.3 Channel assignment policy

In order to let the network work effectively, contagious clusters should use different channels to reduce the interference as much as possible. This is very similar to the situation in cellular networks. Therefore, I borrowed the idea of the spectrum allocation in cellular network to assign the channels in the network. In the standard,

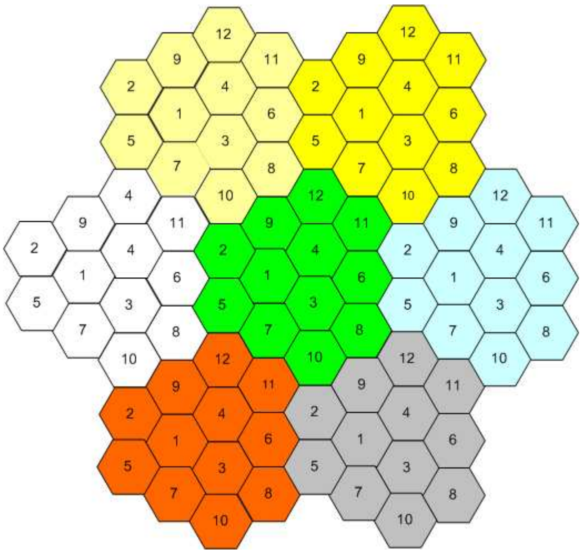


Figure 5.2: Field is divided into hexagons and channels are assigned, channel reuse rate is 12

there are 16 channels used in 2.4G band. Using the methodology introduced in [26] for channel assignment under specified channel reuse rate, I assigned channels to all clusters to make sure the independence distance is far enough so that the interference is kept under certain threshold. One of the channel allocation design is shown in 5.2. The letter in each hexagon represents the channel number used in that region. The independence distance in this design is 60 meter. In a free space transmission, the power of signal decades with the square of the distance. Then the signal to Noise Ratio (SNR) for a sensor in the bound of the cell can be calculated as follow:

$$SNR = 20 * \log_{10} \frac{d_{noise}}{d_{signal}} = 13.9 \tag{5.1}$$

Using the experimental results in the standard page 644, we can find that the bit error rate caused from the same channel interference is less than 10^-8, which means the interference is negligible. In this design, the total number of channels needed is 12, which is less than the 16 available channels in 2.4G band. Given a field, we can

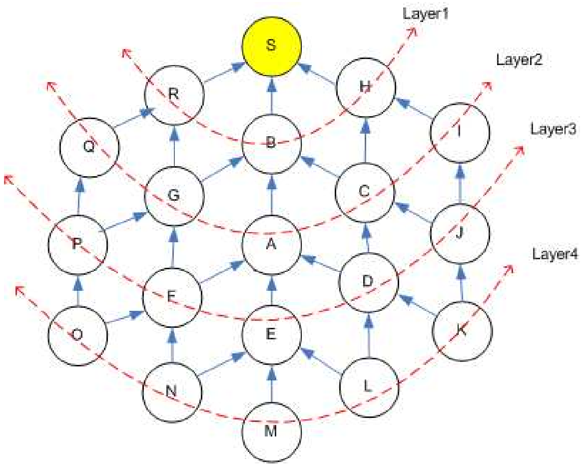


Figure 5.3: Routing topology for a sample network

always cover the whole field by hexagon regions and out one coordinator in the center of each hexagon.

After the channel assignment, I will introduce how to schedule the beacon phase of all the clusters.

5.2.4 Beacon scheduling

First I assign a cluster to be the sink cluster and set all the data destination to be the sink cluster. Before applying any routing algorithms, I divide all the clusters into layers around the sink cluster. Figure 5.3 shows the sink cluster and the layers around the sink. In the graph, a vertex represents a cluster in the corresponding hexagon cell and an an edge represents a possible routing path for the bridge to pass a packet. It is possible for a bridge to send packets to multiple parents and the data flow to each parent is adjustable. For example, node C passes packets to B with probability 0.3 and to H with probability 0.7.

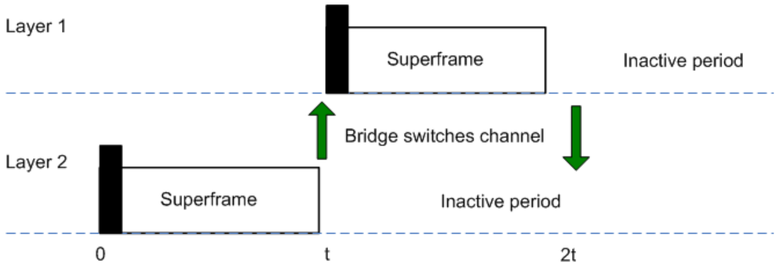


Figure 5.4: Beacon phrase scheduling between contagious layers

In figure 5.3, I assign clusters into layers. Sink S is in Layer 0, R, B and H are in Layer 1, A, C, I, G, Q are in layer 2, etc. Here I introduce the terms of ”parent cluster” and ”child cluster”. A parent cluster is the cluster which is next to the child cluster and is one hope closer to the sink cluster. A child cluster should have its beacon phase one superframe earlier than the parent layer. In this way, as soon as the communication in the child cluster finishes, the communication in the cluster starts. This can minimize the beacon waiting delay for the packets. The beacon time of all the clusters in the same layer is scheduled at the same time. Figure 5.4 illustrates the beacon phrase scheduling between layer1 and layer2. At time 0, layer 2 coordinators broadcast beacons and collect data in their own clusters. At the time t, when the superframes for layer 2 finish, coordinators in layer2 switch their channels into their neighbours' channels in layer1 and pass the data in their buffers to their parent clusters. At time 2t, when the data passing finishes then the bridges switch back to their home clusters. In order to simplify the routing problem, I let bridges always pass packets to the higher layer, which is the shortest path to the destination. In this way I can avoid the problem of routing loop automatically.

5.2.5 Lifetime balanced distributed routing

In a cluster-based IEEE 802.15.4 compliant sensor network, coordinators are responsible for maintaining the information of their clusters and they need to work continuously. Ordinary sensors put themselves in sleep mode in most of the time and when they wake up, they send data in their buffers to their PAN coordinators using the collision based CSMA-CA algorithm. I define the data flow from local sensors to the coordinator as local flow. Meanwhile, bridges from lower layer neighbour clusters also send data to this coordinator. I define the flow from other clusters via bridges to the coordinator as external flow. Local flow can be fixed while external flow depends on the current routing status. A local sensor needs to compete the media resource with the other sensors including bridges using the collision based CSMA-CA algorithm. Higher traffic load in the cluster leads to higher competition and the sensors spend more time to send out data and therefore consume more energy in average.

In this lifetime-balance routing algorithm, I use the criterion of estimated lifetime to make routing decision. A bridge always chooses to send more data to the path with the higher estimated lifetime. To calculate the estimated lifetime of a cluster, the cluster coordinator needs to know the current battery power in the cluster and the energy consumption rate in the cluster. To get the current battery power and energy consumption rate, sensors should report the estimated energy consumption and battery power left periodically and sends this information to the coordinator by piggy-backing this additional information after regular data packet. I assume the additional data takes 4 bytes and the regular data is 30 bytes, then the piggy-backing data payload would be 34 bytes. Coordinators collect the energy consumption information and take the average energy consumption per packet over the sampling period t.

In order to save energy to send routing information, sensors will only update its battery lifetime when it is changed by a certain amount. For example, if I set the level step to be 1000 unit, then sensors only send information when their accumulated energy consumption reach 1000, 2000, etc. When a coordinator receives the update value of the battery residential energy of a sensor, it will update the corresponding table which keeps tracking of the current battery power of all the nodes in the cluster.

Equation 5.2 is used to calculate the energy consumption rate per packet in a cluster.

$$E_i = (\sum_{j=1}^{N_i} E_{ij})/NP_i \tag{5.2}$$

where, E_i is the energy consumption rate in the ith cluster; N_i is the number of sensors in cluster i; E_{ij} is the cumulative energy consumption of the j_{th} sensor in cluster i during the sampling period; NP_i is the total number of packets of cluster i from the local data flow in the sampling period t.

In the end of sampling period t, a cluster will calculate the estimated lifetime of the cluster using drain rate method, that is, a cluster lifetime is the cluster remaining energy divided by the cluster energy consumption rate. As shown in formula 5.3, the remaining cluster battery power is the sum of all the remaining powers for all the sensor nodes.

$$LT_i = (\sum_{j=1}^{N_i} RBP_{ij})/(E_i R_i) \tag{5.3}$$

where, LT_i is the estimated lifetime of cluster i; N_i is the number of sensors in cluster i; RBP_{ij} is the estimated remaining battery power for sensor j in cluster i; R_i is the

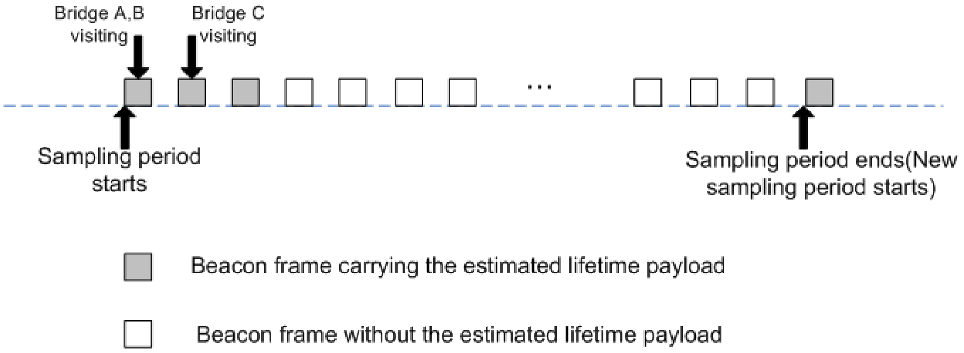


Figure 5.5: Bridges retrieves estimated lifetime of parent clusters in the beginning of the sampling period

data rate of the cluster i.

At the end of the first sampling period, a second sampling period begins and a cluster coordinator will broadcast the updated estimated cluster lifetimes using the beacon data payload for a few beacon cycles and all the bridges are scheduled to visit the cluster and retrieve the estimated lifetime of this cluster. Figure 5.5 illustrates the lifetime information exchange process. When a sampling period starts, coordinator's beacon will carry the estimated lifetime of this cluster and bridges connected to this cluster will be scheduled to visit and retrieve the lifetime information.

A bridge needs to have the estimated lifetime of all the parent clusters before making a routing decision. It is scheduled to visit all the parent clusters one by one and retrieve their estimated lifetimes. After getting the estimated lifetimes of the parent clusters, the bridge will use the flow adjusting based lifetime-balance routing algorithm (will be described later) to route less data to the shortest lifetime cluster and route more data to longer lifetime clusters. The effort of this lifetime-balance routing algorithm is try to alleviate the bottleneck of the network and reach the workload balance of all the parent clusters eventually.

In a cluster-based sensor network, if one cluster drains out, then the whole network is considered to be impaired. Therefore, I assume that the lifetime of the cluster-based network is the shortest lifetime of all the clusters. That is, the lifetime of the network is the time that at least one of the clusters drains out completely. On the other hand, we define the lifetime of a cluster to be the time that the last sensor in that cluster drains out.

The goal of this lifetime balance routing algorithm is to manage the data flow in the network so that all the clusters are fair loaded and it can prevent the situation that one cluster dies much earlier than the others.

The lifetime-balance routing decision algorithm is described in a flow chart shown in figure [5.6].

The algorithm uses an iterative approach to adjust the data flow to all the parents. The routing decision table is updated periodically according to the current status of the network. The algorithm is described as follow.

Denote the probability to send the next packet to node V_i is P_i. If the lifetime of a path is shorter than all the others then decrease the flow to this path.

1) In the beginning, initialize the parameters for the current adjusting step and the upper-bound and lower-bound of the adjusting step. I choose 0.1 as the initial adjusting step and 0.01, 0.2 to be the lower-bound and upper-bound of the adjusting step.

The following actions are triggered periodically to update the routing decision table.

2) The coordinator switches to all the parents in order and retrieve their current

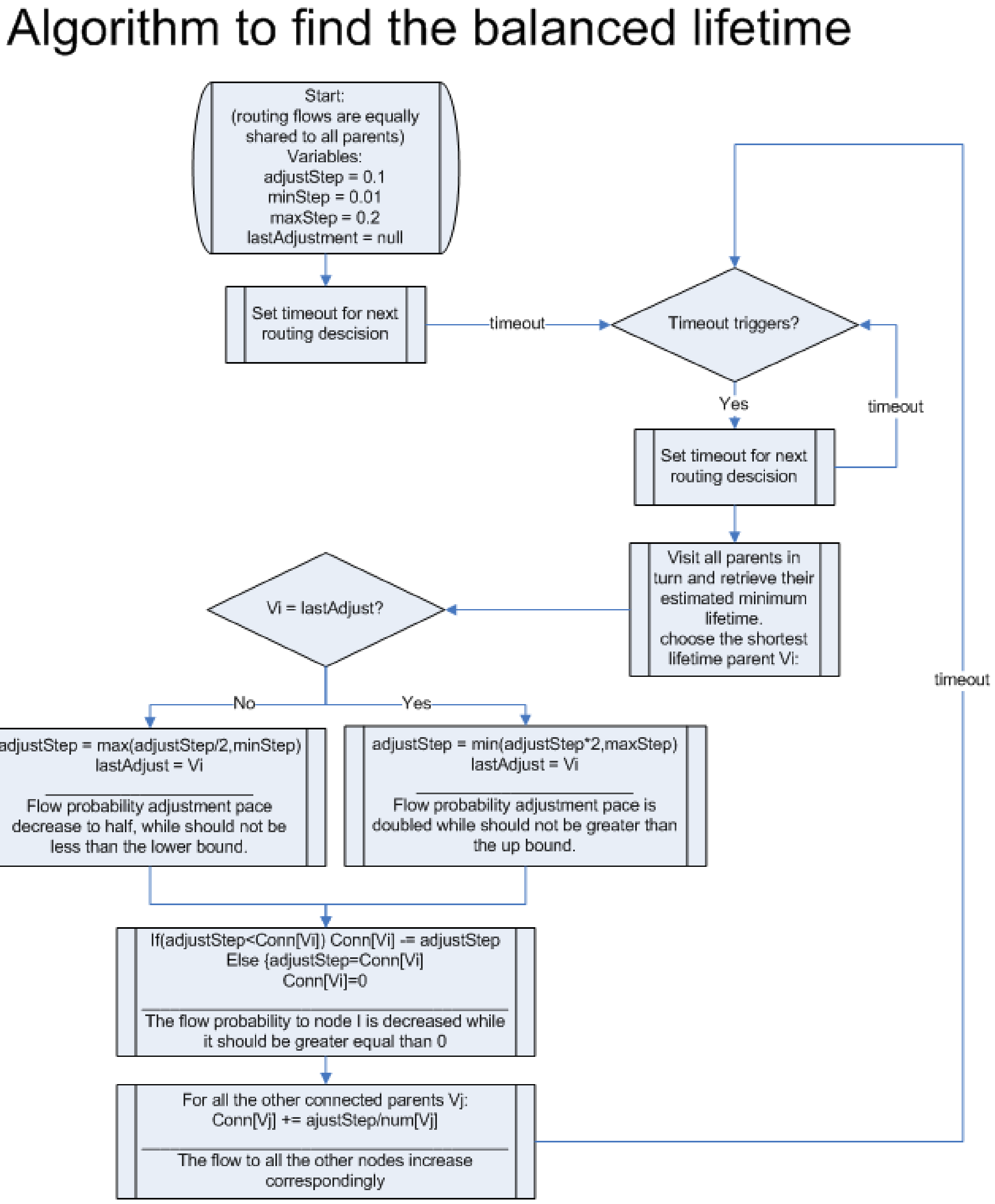


Figure 5.6: Algorithm flow chart of routing decision

estimated lifetime of the cluster. Find the parent node with the shortest lifetime, denoted V_i.

3) If V_i is different from the last V_i node, then the flow adjusting step is set to be a half of the previous value or the lower-bound, whichever is larger. Otherwise it is set to be double of the previous value or the upper-bound, whichever is smaller.

4) The flow probability to V_i is decreased by the adjusting step, while it should always be greater equal to 0. $P_i = max(P_i - adjustment, 0)$

5) The flows to all the other paths are increased evenly to balance the total flow.

In this algorithm, coordinators do not need to know the whole picture of the network. The calculation is distributed to each bridge. Sensor nodes update their status information only when it is different from the previous value. The increment of the network size will not increase the computation amount on each bridge. So this algorithm is scalable and suitable for large scale sensor networks.

5.3 Simulation Results and Analysis

Our simulation models were built on Petri Net based simulation tool Artisoft Artifex[12]. The first simulation model is a three clusters network structure as shown in figure 5.7. The purpose of choosing this model is that we this is a simple structure that can validate the routing algorithm. In this network model, three clusters C0, C1, C2 are equal sized and each has 18 nodes per cluster. All the clusters use duty cycle management [22] to save energy. The packet size exchanged is 30 bytes. Child cluster C2 has two parent clusters and data go out of C2 will go to both C0 and C1 based on the routing decision.

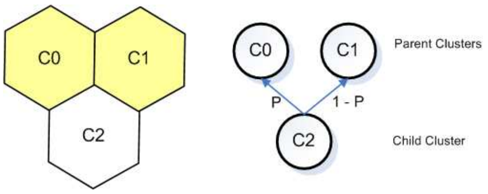


Figure 5.7: A network structure to be implemented in simulation

Table 5.1: Parameters in expeiment 1

Parameter	Value
Packet size	30 bytes
info data	4 bytes
event sensing reliability	20 pkt/s
Initial battery budget	4.32 minute continuous working time
sampling interval	128 second
sensor info update gap	2500 backoff lifetime
initial flow to C0	0
population in C0	18
population in C1	18
population in C2	18

Table 5.1 gives the parameter list for experiment 1.

Figure 5.8 shows the event sensing reliability of 3 clusters vary with time. Figure 5.9 shows the nodes death time of 3 clusters. Figure 5.10 shows the routing decision of

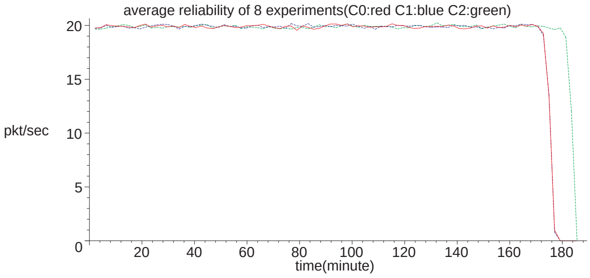


Figure 5.8: Simulation results of local event sensing reliabilities changing with time.

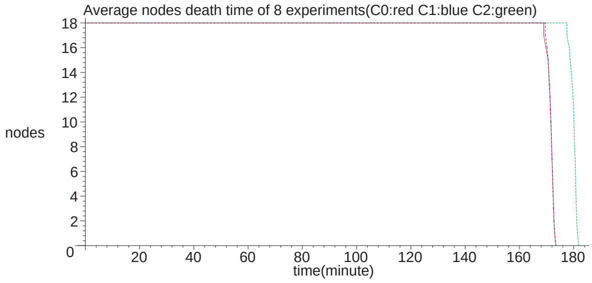


Figure 5.9: Simulation results of nodes death time of clusters.

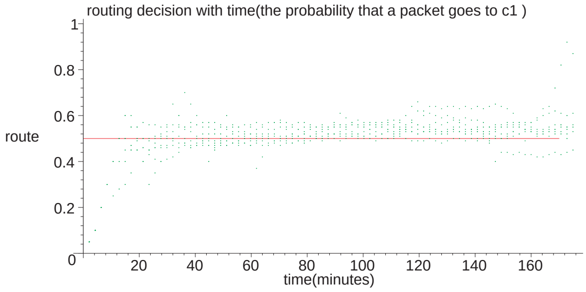


Figure 5.10: Simulation results of routing decision that flow from C2 to C0.

the flow from C2 to C0 changing with time. From the simulation results, I notice that the lifetime-balance routing algorithm efficiently balanced the lifetime of the clusters. The routing decision split the data flow around half and half to both parents. This makes both parents equally loaded and they have very close cluster lifetime eventually.

In the second experiment, I let the parameter sensor energy consumption "update gap" increase from 1 backoffs to 10000 backoffs by step 2500. The 3-cluster network lifetime result with different parameters is shown in figure 5.11. I noticed that when parameter "update gap" is greater than 2500, the lifetime of the network is close to be a stable value.

In the third experiment, I build a large-scale network model with 7 clusters and each cluster has at least 50 sensors in it (figure5.12). Duty cycle management is

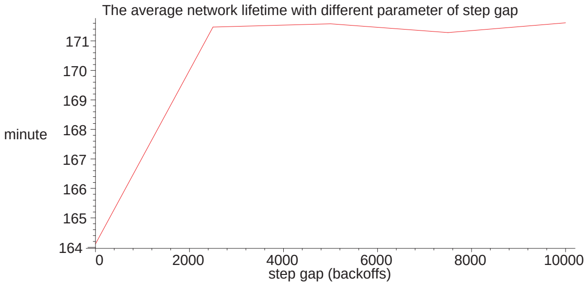


Figure 5.11: Simulation results of network lifetime with different parameter "update gap".

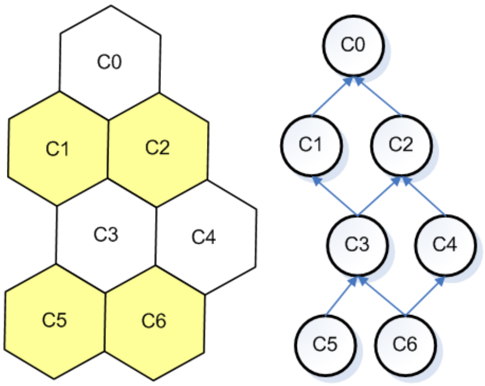


Figure 5.12: A network structure to be implemented in simulation

implemented on each node. Sensors nodes are put into sleep mode periodically in order to save energy. The system requires every cluster sending 5 data packets to the sink per second (150 Byte/Second). As I indicated before, the cluster with high workload will have high energy drain rate. Therefore, if all the clusters have the same number of nodes, then sink cluster will be the first cluster to be drained out. However, using the nodes population techniques [20], I can populate clusters according to their traffic load. In this case, the balanced workloads for clusters 0 to 6 are: 35, 15, 15, 10, 10, 5, 5. The nodes population are: 58, 52, 52, 51, 51, 50, 50.

Table 5.2 gives the parameter list for experiment 3.

Table 5.2: Parameters in expeiment 3

Parameter	Value
Packet size	30 bytes
info data	4 bytes
event sensing reliability	5 pkt/s
Initial battery budget	4.32 minute continuous working time
sampling interval	17 minutes
sensor info update gap	2500 backoff lifetime
initial flow to C0	0.5
population in C0	58
population in C1,C2	52
population in C3,C4	51
population in C5,C6	50

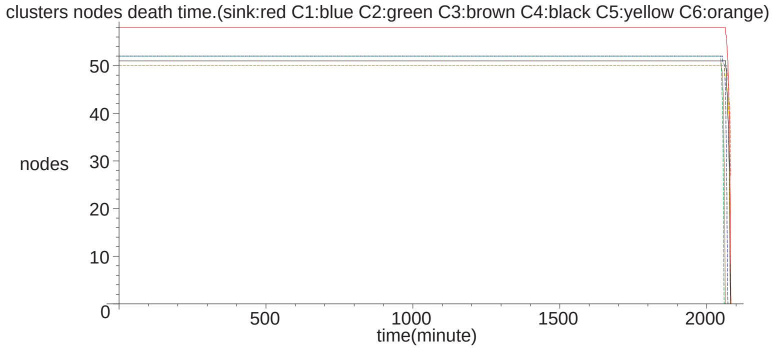


Figure 5.13: Nodes death time of network of 7 clusters model

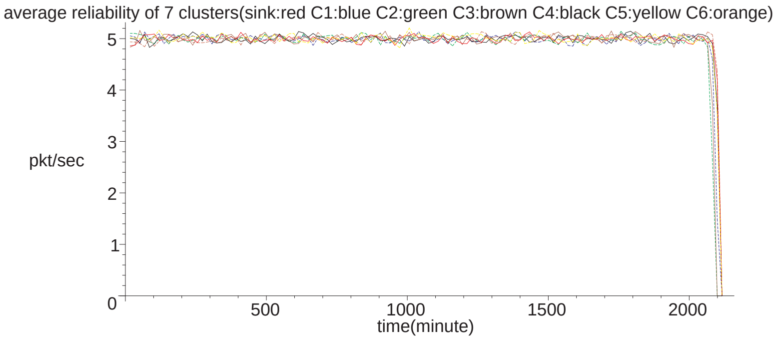


Figure 5.14: Event sensing reliabilities of network of 7 clusters model

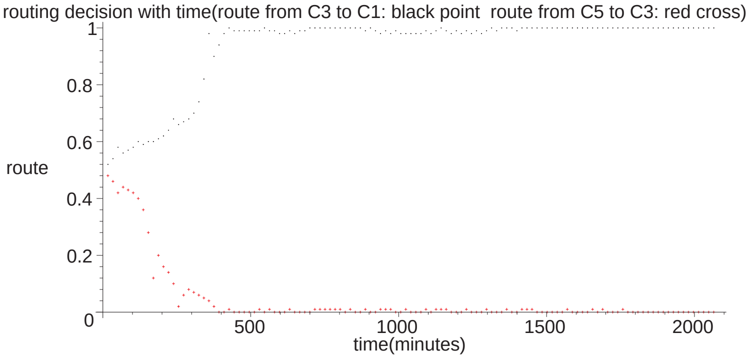


Figure 5.15: Routing decision of 7 clusters model

Figure 5.14 shows the event sensing reliability of all the clusters in the network. Figure 5.13 shows the nodes death time of all the clusters. I notice that after the repopulation, all the clusters have very close death time. The lifetime of all the clusters only have difference of less than 1 percent. The network is utilized very well.

The theoretical routing for the network to reach balanced workload is probability 1 from C3 to C1 and probability 0 from C6 to C3. Figure 5.15 shows the routing probability from C3 to C1 and the routing probability from C6 to C3. Both routing decisions converge to their theoretical optimal results after 5 hours network time. This transition period is negligible compared with the network lifetime.

Figure 5.16 shows the estimated lifetime for each clusters which is useful when doing routing decision.

Figure 5.17 shows the average service time for sensors to send out a packet. Blue points are service time in the sink, which is obvious higher than the service time in other clusters. This is because the sink cluster has higher traffic load and the media is busier than the other clusters. Therefore, sensors in the sink need to wait longer in average to get the service.

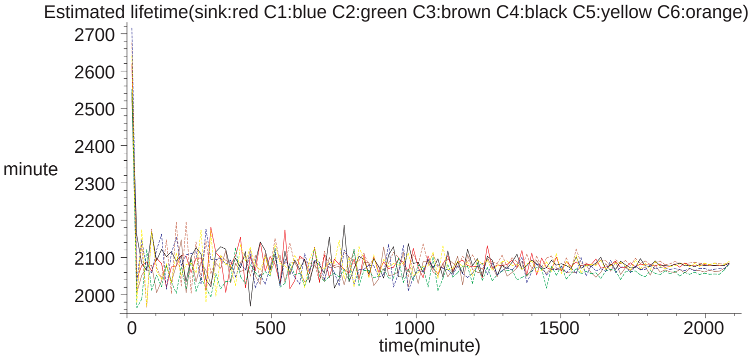


Figure 5.16: Estimated lifetime of 7 cluster model

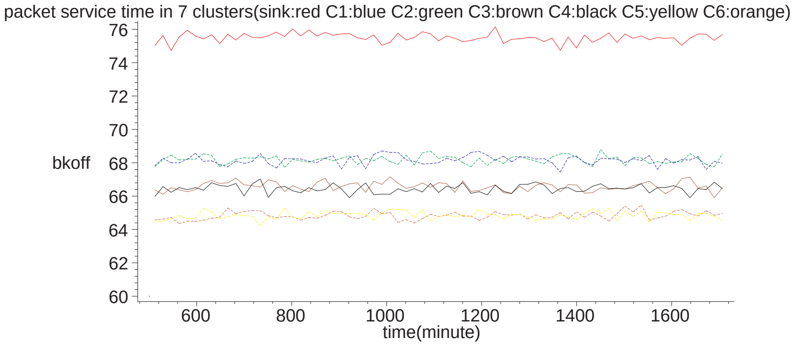


Figure 5.17: Average service time of 7 cluster model

Figure 5.18 shows the average nodes sleep time after sending out a packet in each cluster. The experiment result shows that the average sleep time for sensors are around 12 seconds for C0 and between 10 and 11 for the other clusters. Sink cluster C0 has the highest sleep time. This is because sink cluster has more nodes and each node takes longer turn in average to get a chance to transmit a packet.

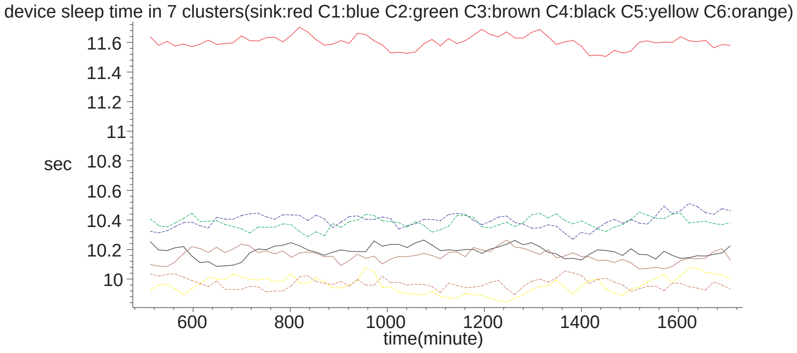


Figure 5.18: Average node sleep time of 7 cluster model

5.4 Conclusion

In this chapter, I propose a simple method to build a large-scale LR-WPAN sensor scatternet. I build clusters by dividing physical area into equal sized hexagons and lay coordinators in the center of hexagon cells. I propose a channel assignment method and a beacon scheduling method. At last, I propose a lifetime-balance routing algorithm using data flow tuning method to balance the workload of clusters based on their residential remaining power. I simulated this routing algorithm in both small scale and large-scale network. I found that this distributed routing algorithm can effectively balance the lifetime of the clusters and routing decision converges to the theoretical optimal routing. Since each sensor makes routing decision locally and the increment of network size will not increase the computation amount, then this routing algorithm is a scalable routing algorithm.

Chapter 6

Concluding Remarks and Future Research

In this thesis work, I started to study the performance of cluster-based sensor network by constructing a two-cluster model interconnected by a Master/Slave bridge. I have studied the bridge performance under CSMA-CA mode and GTS mode with acknowledged and non-acknowledged transfer respectively. Then I simulated the large cluster with redundant nodes and nodes activities are controlled by duty cycle management. In order to equalize the clusters' lifetime in the network, this thesis proposed a nodes compensation method to extend the heavy duty cluster lifetime by placing more redundant nodes inside. My simulation results verify the results of the analytical model which is developed by my supervisor. At last, I designed a scalable, energy-efficient and lifetime-balanced routing algorithm to balance and extend the network lifetime as much as possible.

All the above work is based on the assumption that no data aggregation is performed in the network. The future work of this thesis can be the case that data aggregation is considered in the network and the improvement and trade off of introducing data aggregation is waiting to be found.

Bibliography

[1] Petri net. Website. `http://en.wikipedia.org/wiki/Petri_net`.

[2] *Standard for part 15.1: Wireless medium access control (mac) and physical layer (phy) specification for wireless personal area networks (WPAN), IEEE Standard 802.15.1, IEEE.* New York, NY, 2002.

[3] *Standard for part 15.4: Wireless medium access control (mac) and physical layer (phy) specification for low rate wireless personal area networks (WPAN), IEEE Standard 802.15.4, IEEE.* New York, NY, 2003.

[4] I. F. Akyildiz, Y. Sankarasubramaniam W. Su, and E. Cayirci. Wireless Sensor Networks:a Survey. *Communications Magazine, IEEE*, 40(8):102–114, 2002.

[5] Javed Aslam, Qun Li, and Daniela Rus. Three Power-Aware Routing Algorithms for Sensor Networks. *Wireless Communication and Mobile Computing*, 3:187–208, 2003.

[6] B. Bougard, F. Catthoor, D.C. Daly, A. Chandrakasan, and W. Dehaene. Energy Efficiency of the IEEE 802.15.4 Standard in Dense Wireless Microsensor Networks: Modeling and Improvement Perspectives. In *Design, Automation and*

Test in Europe, 2005. Proceedings, volume 1, pages 196–201, Munich, Germany, March 2005.

[7] Jae-Hwan Chang and Leandros Tassiulas. Maximum Lifetime Routing in Wireless Sensor Networks. *IEEE/ACM Transactions on Networking (TON)*, 12(4):609–619, 2004.

[8] A.R.A. Eirmon, Choonq Seon Hong, and T. Suda. EAREC: Energy Aware Routing with Efficient Clustering for Sensor Networks. In *Consumer Communications and Networking Conference, 2006. CCNC 2006.*, volume 1, pages 330–335, Las Vegas, Nevada, January 2006.

[9] C. Evans-Pughe. Bzzzz zzz[Zigbee Wireless Standard]. *IEE Review*, 49(3):12628–31, 2003.

[10] R. G. Gallager, P. A. Humblet, and P. M. Spira. A distributed algorithm for minimum weight spanning trees. *Lab. Inform. Decision Syst., Massachusetts Inst. of Technol., Cambridge, MA, Tech. Rep. LIDS-P-906-A,*, 1979.

[11] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes.* Oxford University Press, 1992.

[12] Rsoft Design Group. Artifex v.4.4.2, 2003.

[13] G. Gupta and M. Younis. Performance Evaluation of Load-balanced Clustering of Wireless Sensor Networks. In *Telecommunications, 2003. ICT 2003. 10th International Conference on*, volume 2, pages 1577–1583, Papeete, French Polynesia, February 2003.

[14] R. Iyer and L. Kleinrock. Qos control for sensor networks. In *In Proc. ICC03*, pages 517–521, 2003.

[15] Dongkyun Kim, Garcia-Luna-Aceves J.J., K. Obraczka, J.-C. Cano, and P. Manzoni. Routing mechanisms for mobile ad hoc networks based on the energy drain rate. *IEEE Transactions on Mobile Computing*, pages 161–173, 2003.

[16] Gang Lu, Bhaskar Krishnamachari, and Cauligi S. Raghavendra. Performance Evaluation of the IEEE 802.15.4 MAC for Low-rate Low-power Wireless Networks. In *Performance, Computing, and Communications, 2004 IEEE International Conference on 2004*, pages 701–706, Phoenix, Arizona, April 2004.

[17] J. Misic and C. J. Fung. The Impact of Master-Slave Bridge Access Mode on the Performance of Multi-Cluster 802.15.4 Network. *Elsevier's Computer Networks*, 2007.

[18] J. Misic, C. J. Fung, and V. B. Misic. Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis. In *Eighth International Symposium of Parallel Architectures, Algorithms and Networks (I-SPAN 2005)*, Las Vegas, NV, December 2005.

[19] J. Misic, C. J. Fung, and V. B. Misic. Interconnecting 802.15.4 Clusters in Slotted CSMA-CA Mode. In *2006 IEEE International Conference on Communications ICC 2006*, Istanbul, Turkey, June 2006.

[20] J. Misic, C. J. Fung, and V. B. Misic. Network lifetime equalization in interconnected 802.15.4 clusters. In *The Third International Conference on Quality*

of Service in Heterogeneous Wired/Wireless Networks(QShine 2006), Waterloo, Ontario, August 2006.

[21] J. Misic and V. B. Misic. Access Delay and Throughput for Uplink Transmissions in IEEE 802.15.4 PAN. *Elsevier Computer Communications Journal*, 28(10):1152–1166, 2005.

[22] J. Misic and V. B. Misic. Duty Cycle Management in Sensor Networks Based on 802.15.4 Beacon Enabled MAC. *Ad Hoc and Sensor Wireless Networks Journal*, 1(3):207–233, 2005.

[23] J. Misic, S. Shafi, and V. B. Misic. Avoiding the bottlenecks in the MAC layer in 802.15.4 low rate wpan. In *In Proc. First International Workshop on Heterogeneous Wireless Sensor Networks HWISE2005*, volume 2, pages 363–367, July 2005.

[24] J. Misic, S. Shafi, and V. B. Misic. Maintaining Reliability through Activity Management in an 802.15.4 Sensor Cluster. In *IEEE Transactions on Vehicular Technology*, volume 55, pages 779–788, May 2006.

[25] N.Vlajic and D. Xia. Sensor networks: To cluster or not to cluster. In *Proc of the 2006 International Symposium on World of Wireless, Mobile and Multimedia Networks*, pages 258–268, 2006.

[26] Theodore S. Rappaport. *Wireless Communications Principles and Practice.* Prentice -Hall, Inc., 1996.

[27] V. Rodoplu and T. H. Meng. Minimum energy mobile wireless networks. In

Proc. IEEE Int. Conf. Communications (ICC98), volume 3, pages 1633–1639, Atlanta,GA, June 1998.

[28] K. Scott and N. Bambos. Routing and channel assignment for low power transmission in pcs. In *ICUPC'96*, Cambridge, MA, 1996.

[29] S. Singh, M. Woo, and C.S. Raghwendra. Power aware routing mobile in mobile ad hoc networks. In *Mobicom 1998*, Dallas TX, 1998.

[30] C. K. Toh. Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks. *IEEE Communication Magazine*, pages 138–147, 2001.

[31] Yang Yang, Huihai wu, and Weihua Zhuang. MESTER: Minimum Energy Spanning Tree for Efficient Routing in Wireless Sensor Networks. In *The Third International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine 2006)*, volume 1, pages 330–335, Waterloo, Ontario, August 2006.

[32] Shouyi Yin and Xiaokang Lin. Multipath Minimum Energy Routing in Ad Hoc Network. In *IEEE International Conference 2005*, volume 5, pages 3182–3186, May 2005.