Clustering Algorithms for Sensor Networks and Mobile Ad Hoc Networks to Improve Energy Efficiency

Dali Wei

This thesis is submitted in partial fulfillment of the academic
requirements for the degree of
Doctor of Philosophy in Electrical Engineering
in the Faculty of Engineering and The Built Environment
University of Cape Town

September 2007

To My Wife Jinjin Liu

Abstract

Many clustering algorithms have been proposed to improve energy efficiency of ad hoc networks as this is one primary challenge in ad hoc networks. The design of these clustering algorithms in sensor networks is different from that in mobile ad hoc networks in accordance with their specific characteristics and application purposes.

A typical sensor network, which consists of stationary sensor nodes, usually has a data sink because of the limitation on processing capability of sensor nodes. The data traffic of the entire network is directional towards the sink. This directional traffic burdens the nodes/clusters differently according to their distance to the sink. Most clustering algorithms assign a similar number of nodes to each cluster to balance the burden of the clusters without considering the directional data traffic. They thus fail to maximize network lifetime.

This dissertation proposes two clustering algorithms. These consider the directional data traffic in order to improve energy efficiency of homogeneous sensor networks with identical sensor nodes and uniform node distribution. One algorithm is for sensor networks with low to medium node density. The other is for sensor networks with high node density. Both algorithms organize the clusters in such a way that the cluster load is proportional to the cluster energy stored, thereby equalizing cluster lifetimes and preventing premature node/cluster death. Furthermore, in a homogeneous sensor network with low to medium node density, the clusterhead is maintained in the central area of the cluster through re-clustering without ripple effect to save more energy. The simulation results show that the proposed algorithms improve both the lifetime of the networks and performance of data being delivered to the sink.

A typical mobile ad hoc network, which usually consists of moveable nodes, does not have a data sink. Existing energy-efficient clustering algorithms maintain clusters by periodically broadcasting control messages. In a typical mobile ad hoc network, a greater speed of node usually needs more frequent broadcasting. To efficiently maintain the clusters, the frequency of this periodic broadcasting needs to meet the requirement of the potentially maximum speed of node. When the node speed is low, the unnecessary broadcasting may waste significant energy. Furthermore, some clustering algorithms limit the maximum cluster size to moderate the difference in cluster sizes. Unfortunately, the cluster sizes in these algorithms still experience significant difference. The larger clusters will have higher burdens. Some clustering algorithms restrict the cluster sizes between the maximum and minimum limits. The energy required to maintain these clusters within the maximum and minimum sizes is quite extensive, especially when the nodes are moving quickly. Thus, energy efficiency is not optimized.

This dissertation proposes a clustering algorithm to improve energy efficiency for mobile ad hoc networks by reducing unnecessary broadcasting and by adaptively balancing cluster sizes. Control messages are broadcast adaptively in accordance with the requirement of respective node speed. The unnecessary broadcasting is therefore reduced. Furthermore, the cluster sizes are maintained adaptively in accordance with the node speeds. When the speed is low, maintaining to a similar size is applicable because it better balances power consumption among clusters. However, when the speed is high, the difference in cluster sizes is more relaxed, freeing the network from frequent re-clustering. The simulation results and analysis show that the control messages are broadcast adaptively in accordance with the node speed, energy efficiency is improved and the first-node lifetime of the network is extended.

Acknowledgements

This thesis would never have been completed without the help of many people:

- My supervisor, Prof. H Anthony Chan, for his invaluable guidance, encouragement, and continuous support throughout my PhD study at University of Cape Town.
- My wife, Jinjin Liu, for her love, dedication and patience, whose support was fundamental to the success of my doctoral studies. My parents, brothers and my sisters-in-law, who are the most influential people in my life, for all the support, love and encouragement that they have given me throughout my life.
- Bo Wang, Dongping Li, Yuan Li, Yuechao Fang, Yue Wang, and all my friends in Cape Town, I can never sufficiently express my gratitude.
- Mr Neco Ventura, Yang Li, Shaun Kaplan, and all my colleagues and staff in the Department of Electrical Engineering, thank you for providing me with a very friendly research and life environment while doing my thesis.
- Dr Jane English and Sally Burt for teaching me writing.
- Financial support from the University of Cape Town, Telkom, Nokia Siemens Networks, Telescience, the National Research Council and the Chinese government for awarding me the oversea Chinese PhD student scholarship, which enabled me focus on my research.

Table of Contents

List of Figures

List of Tables

Glossary

50% availability lifetime for single-hop sensor networks: The period when the network starts to work up to the time only 50% of the sensors remain alive.

BPC: BPC is the abbreviation of the algorithm proposed in Chapter 5, which balances power consumption (BPC) throughout the homogeneous clustered sensor network with high node density.

BRPC: BRPC is the abbreviation of the algorithm proposed in Chapter 4, which balances and reduces power consumption (BRPC) in homogenous cluster sensor networks with low to medium node density.

Clustered Ad Hoc Network: In a clustered ad hoc network, the nodes are grouped into clusters according to some special rules. Typically, each cluster has a clusterhead (CH) to act as a local controller or coordinator. Usually there are three types of nodes in a clustered ad hoc network: member node, gateway node and CH. Both CHs and gateway nodes form the backbone of the network with only CHs being essential.

Connectivity lifetime for multi-hop sensor networks: The period when the network starts to work up to the time connectivity between the sink and the network is lost.

Data aggregation ratio, λ: A data aggregation ratio of λ:1 means that, for every '1' unit of that must be sent to the sink when no data aggregation is performed, only 'λ' (which is smaller than 1) units of data must be sent to the sink after local data aggregation is performed.

First-node Lifetime: The period when the network starts to work up to the time any node dies in the network.

LEACH_C_F: LEACH is the abbreviation of low-energy adaptive clustering hierarchy. LEACH_C_F is the abbreviation of the algorithm that is used for comparison in Chapter 4. LEACH_C_F combines both LEACH-C (centralized LEACH) and LEACH-F (LEACH with fixed clusters).

MSPB: MSPB is the abbreviation of the algorithm that is used for comparison in Chapter 6. MSPB limits the maximum size of the cluster and broadcasts control messages periodically to maintain the clusters.

MSSAB: MSSAB is the abbreviation of the algorithm that is used for comparison in Chapter 6. MSSAB limits the maximum size of the cluster, maximizes the service time of the CHs and broadcasts control messages adaptively to maintain the clusters.

Network lifetime in sensor networks: Network lifetime is connectivity lifetime for multi-hop sensor networks, and is availability lifetime for single-hop sensor networks.

RBBPC: RBBPC is the abbreviation of the proposed algorithm in Chapter 6. RBBPC broadcasts control messages adaptively and balances cluster sizes adaptively.

Re-clustering with ripple effect: Re-clustering with ripple effect means that any re-clustering process caused by CH rotation or cluster maintenance alters the cluster topology over the whole network.

Round of data collection: A Round of data collection initiates the cluster set-up when the clusters are organized and ends when the sink receives all data from the CHs. During this procedure, some frames of data are delivered from the member nodes to the CH and on to the data sink.

SSCA: SSCA is the abbreviation of the algorithm that is used for comparison in Chapter 5. SSCA organizes clusters with similar sizes and allows only the CH to be active in each cluster.

SSMR: SSMR is the abbreviation of the algorithm that is used for comparison in Chapter 4. SSMR organizes clusters with similar sizes and requires the CHs to relay data to the sink over multi-hop routes.

Total delivered entire-network data: Data delivered to the sink from the entire network where all sensors are alive up to first-node lifetime.

Total delivered network data: Data delivered to the sink from the network up to the network lifetime.

Total delivered partial-network data: Data delivered to the sink from part of the network where sensors are still alive after first-node lifetime.

1 Introduction

1.1 Clustered ad hoc networks

1.1.1 Ad hoc networks

The past few years have witnessed the proliferation of devices such as intelligent sensors, laptops, palm-tops and mobile phones. The need for easy and spontaneous communications between such devices led to the development of a new class of wireless networks without infrastructure. These networks, known as "ad hoc networks," represent complex distributed systems comprising wireless nodes in arbitrary and temporary network topologies [1, 2]. Each device in an ad hoc network functions as a router to relay data from other devices by multi-hop routes.

According to the purposes of application and the features of the nodes forming the network, ad hoc networks can typically be divided into wireless sensor networks and mobile ad hoc networks [3, 4].

1.1.2 Wireless Sensor networks (WSNs) and mobile ad hoc networks (MANETs)

A typical WSN, as shown in Fig. 1-1, consists of small sensor nodes, which are usually immovable after being deployed in the network, with sensing, data processing and communication capabilities. These networks are usually deployed to accomplish particular tasks, such as target-field imaging, intrusion detection, weather monitoring, disaster management and ambient-condition detection [5]. The resource and capability of a typical sensor node is extremely limited. The data that the sensors generate are therefore not conclusive and a data sink is usually necessary in a typical WSN to collect data for further analysis. These unique characteristics present many challenges for the development and eventual application of ad hoc WSNs [6]. Furthermore, the extremely limited energy supply of sensor nodes makes energy efficiency one of the primary challenges [7, 8].

A typical MANET is shown in Fig. 1-2. Differing from the application purpose of a WSN, a MANET usually aims at enabling communication between the nodes in the network. The nodes in a MANET can usually move freely and arbitrarily [9]. As with sensor nodes, the wireless devices forming MANETs are also usually battery supply-based so that energy efficiency becomes one of the important challenges in MANETs. Unlike the relative stable topology in a WSN formed by stationary nodes, the topology in a MANET is more unstable because of the movement of devices. Therefore, the energy consumed by topology maintenance in a MANET is considerable [10].

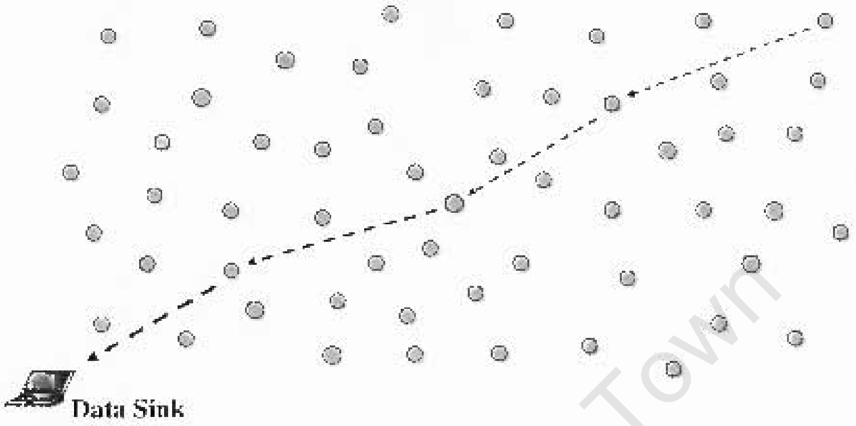


Figure 1-1. Sensor network with data sink.

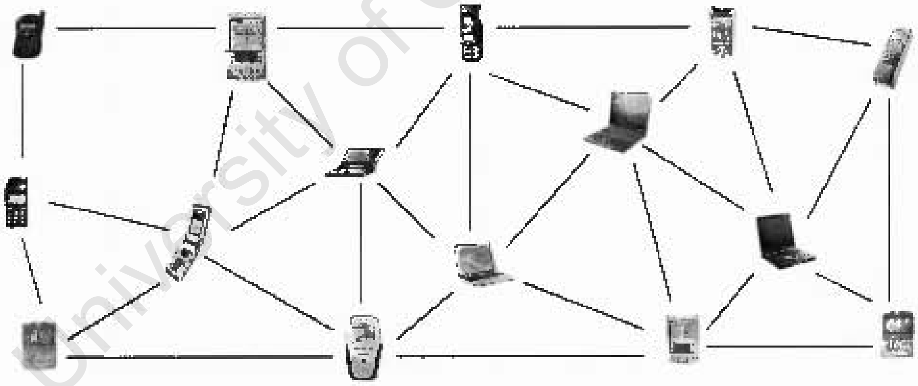
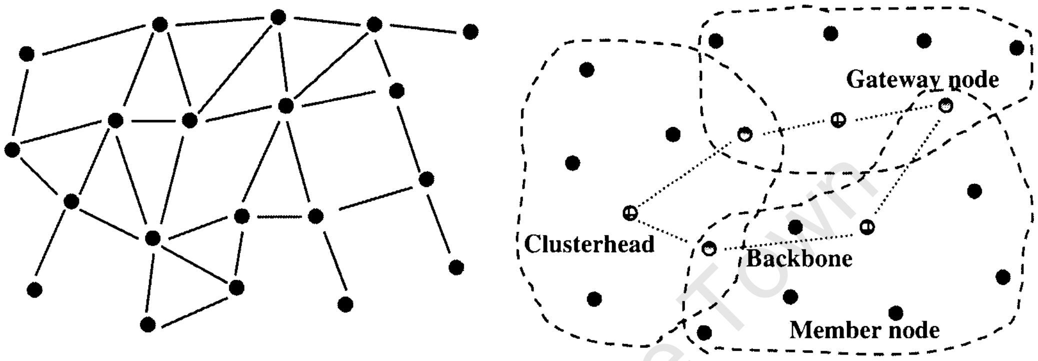


Figure 1-2. Mobile ad hoc network.

As explained above that a WSN usually is formed by stationary nodes and has a data sink/data sinks, whereas a MANET usually is formed by mobile nodes and has not any data sink, during the topology or routing protocol design, such questions as the direction of data traffic and the stability of the network topology needs to be taken into account.

1.1.3 Clustered ad hoc networks

The absence of infrastructure makes it convenient to build an ad hoc network [3]. This infrastructureless network, however, also brings many challenges to topology design. Much research has been conducted and many topology structures have been proposed. The existing structures can be typically classified into flat topology and hierarchical clustering topology, as shown in Fig. 1-3.



(a) Flat topology

(b) Hierarchical clustering topology

Figure 1-3. Topology structure of an ad hoc network.

An ad hoc network with a flat topology (Fig. 1-3(a)) is a peer-to-peer network. This structure is simple yet makes the topology complicated, especially in a large ad hoc network. Hierarchical clustering topology is therefore proposed. In a clustered ad hoc network (Fig. 1-3(b)), the nodes are organized into different virtual groups. The nodes geographically adjacent are allocated to the same cluster according to specified rules with different behaviors for nodes included in a cluster from those excluded from the cluster [11]. In a clustered ad hoc network, the nodes may be assigned a different status or function, such as clusterhead (CH), gateway node, or member node. A CH normally serves as a local controller or coordinator for its cluster, performing intra-cluster transmission arrangements, cluster maintenance, data aggregations and data forwarding. A gateway node is a non-CH node with inter-cluster links so it can access neighboring clusters and forward information between clusters. Both CHs and gateway nodes form the backbone of the network with only CHs being essential. A member node is a non-CH node without any inter-cluster links [11, 12].

A hierarchical clustering topology provides at least five advantages over a flat topology. First, it is more convenient especially in a WSN for a CH to aggregate data when it has collected all data from its member nodes. This consequently decreases power consumption by reducing the amount of data in inter-cluster communication [13, 14]. Secondly, in a clustered ad hoc network, only the CHs and gateway nodes form a virtual backbone for inter-cluster routing, thereby leading to a simpler backbone. The generation and spreading of routing information can then be restricted in this set of nodes. This consequently reduces the flooding to search for a backbone route [15]. Thirdly, a clustering topology makes an ad hoc network more stable, especially in MANETs [16]. When a mobile node changes its attaching cluster, only mobile nodes residing in the corresponding clusters need to update the link information [17]. Thus, local changes need not be seen and updated by the entire network, thereby greatly reducing information processed and stored by each node. Fourthly, a clustering topology results in a better scalability of the network [18]. This hierarchical topology structure makes it more convenient to set up the intra-cluster cooperation as most nodes in the same cluster are usually deployed in a closer area [18]. Finally, a clustering topology facilitates the spatial re-use of resources to increase the system capacity, thereby resulting in higher energy efficiency [19]. With the non-overlapping clustering topology, two clusters may deploy the same frequency or code set if they are not neighboring clusters. Also, a cluster can better coordinate its transmission events with the help of a CH residing in it. This can save many resources used for re-transmission because the transmission collision is reduced [19]. Clustering topology therefore represents a solution with great potential.

1.1.4 Routing protocols

Three typical routing protocol types for ad hoc networks have been proposed: the proactive, the reactive and the hybrid.

Proactive and reactive routing protocols are usually applied in flat topology [20]. When a node applies a proactive routing protocol to communicate with another node, such as in [21, 22], it maintains the routing information of the entire network. When it is ready to send data packets, the routing information to a destination node can be derived from its routing table. The network topology may be changed because of node failure or movement, which requires the routing information be updated by broadcasting control messages to acquire new links between the nodes. Proactive protocols are efficient for ad hoc networks with stable topology, such as WSNs formed by immovable nodes. However, to store the routing information of the entire network needs large memory size, thereby limiting its application in large sized WSNs. When a node applies a reactive protocol to communicate with another node, such as in [23, 24], it first explores its cache. If the cached routing information is not expired, it will send the data along this route. If there is no routing information to the destination or the cached routing information has expired, it needs to flood a route request to find an available route. Reactive protocols are more suitable for a network with an unstable topology, such as a MANET. Serious flooding, however, becomes a problem and challenge to control.

A hybrid routing protocol combines a proactive and a reactive routing protocol in a network. This hybrid protocol, usually applied in a clustered ad hoc network, features less routing table overhead than the proactive protocol and less flooding than the reactive protocol [25]. Typically, in a clustered ad hoc network, intra-cluster communication uses proactive routing protocol whereas inter-cluster communication applies reactive routing protocol [11].

1.1.5 Energy saving for WSNs using clustering algorithms

As the energy supply for sensor nodes is extremely limited, it is essential to improve energy efficiency or to optimize the lifetime for WSNs [26]. Clustering algorithms, as the potentially most energy efficient organization, witnessed wide application in the past few years [27] and numerous clustering algorithms have been proposed for energy saving. The most recent clustering algorithms that can be accessed are investigated and classified into six categories [12].

The first category aggregates the data that need to be sent to the sink. This aggregation can be realized over each hop or at a CH. The amount of data is greatly reduced after aggregation. Energy associated with data routing to the sink is therefore saved.

The second category balances the power consumption among the nodes by rotating the CH role. A CH needs to collect data from its member nodes, aggregate the collected data and then relay the aggregated data to the sink over a single-hop or a multi-hop route. A CH therefore will have a much higher burden than a member node. Rotating the CH role distributes this higher burden among the nodes and protects a CH from dying prematurely.

The third category balances the burden among clusters/CHs. The algorithms in this category assign (approximately) the same number of nodes to each cluster/CH to balance the burdens of the clusters/CHs, thereby preventing nodes from dying prematurely.

The fourth category reduces the total intra-cluster power consumption. Power consumption of sending data from each member node to the CH is relative to the hop distance in a single-hop cluster or number of hops in a multi-hop cluster. Selecting the node in the central area of a cluster to be CH reduces the total distance/hops from the member nodes to the CH, thereby saving energy of intra-cluster communication.

The fifth category optimizes the routes by combining different power-level hops. In a sparse WSN with an uneven node distribution, the nodes are organized into clusters with different power levels. Each node can find some routes towards the destination with different combinations of power-level hops. Selecting the most energy-efficient route reduces power consumption of routing data.

The sixth category reduces the power consumption of each communication link. Assigning the minimum needed transmission power to each link reduces the total power consumption of that route.

1.1.6 Energy saving for MANETs using clustering algorithms

Energy efficiency is a primary challenge in MANETs as well as in WSNs. To extend the network lifetime, a clustering algorithm for MANETs needs to consider the cost of maintaining the clusters. As with WSNs, the most recent clustering algorithms for MANETs are investigated and classified into four categories.

The first category distributes the higher burden of a CH among the nodes. Similar to the CH in a clustered WSN which plays the role of local controller, a CH in a MANET usually plays the role of coordinator. It needs to relay all data from its member nodes and maintain the cluster. A CH therefore has a higher burden than a member node. Rotating the CH role distributes this higher burden among the nodes.

The second category reduces the difference in burdens among clusters. The power consumption is also relative to the cluster size. The nodes in a larger sized cluster typically encounter a higher traffic burden, especially the CH. Limiting the maximum size or limiting both the maximum and the minimum sizes of the cluster moderates the difference in size, which reduces the difference in the traffic burden among the nodes/clusters and thus better balances the power consumption.

The third category reduces the power consumption of maintaining the cluster. The free and arbitrary node movement may frequently change the clusters. Reducing energy consumed by cluster maintenance results in higher energy efficiency.

The fourth category locates the CH in the central area of the cluster to reduce the power consumption of intra-cluster communication. This idea is similar to the one in WSNs. The node movement, however, makes realization in MANETs more difficult.

1.2 Weaknesses in existing energy-efficient clustering algorithms

1.2.1 Clustering algorithms for WSNs

In a WSN, the data generated by sensor nodes are usually not conclusive and need to be sent to the sink for further analysis, thereby making the data traffic directional towards the sink. This directional traffic burdens the clusters differently according to their distance to the sink. In a clustered WSN, in which the CHs send data to the sink over multi-hop routes, the clusters nearer the sink will usually have higher burdens because they need to relay more data from others. Most clustering algorithms assign (approximately) the same number of nodes to each cluster, which means (approximately) the same energy store in each cluster in a homogeneous WSN with identical nodes. Clusters with higher burdens will die prematurely. The areas in which these sensors are deployed will be out of coverage due to this premature node death, thereby degrading network performance. Furthermore, this premature cluster death will not only increase power consumption but also disconnect the sink and the network. When the clusters nearer the sink die prematurely, the clusters that are farther away from the sink need longer transmission range to reach the sink to avoid data loss, thereby increasing power consumption. The width of the area that exhausts energy keeps increasing. When this width exceeds the maximum transmission range of the sensors, the data generated by other nodes that still have residual energy can no longer be relayed to the sink. This residual energy is wasted. Therefore, the directional data traffic not only degrades the network performance, but also reduces the energy efficiency. Unfortunately, most existing energy-efficient clustering algorithms for WSNs do not take directional data traffic into account.

By contrast, locating the CH in the central area of the cluster minimizes total distances between the CH and the member nodes, thereby reducing the power consumption of intra-cluster communication. However, most clustering algorithms maintain the CH in the central area of the cluster by re-clustering the nodes with ripple effect. Re-clustering with ripple effect means that any re-clustering process caused by rotation of CH or cluster maintenance alters the cluster topology over the whole network. Therefore, ripple effect of re-clustering makes the total amount of clustering-related exchange of messages for re-clustering phase similar to that in the cluster formation phase [11]. The flooding caused by the nodes being frequently re-clustering with ripple effect consumes significant energy [14]. This fails to optimize energy efficiency.

1.2.2 Clustering algorithms for MANETs

The topology of a clustered MANET may be changed frequently due to node movement. The application of the proactive routing protocol in intra-cluster communication requires the routing information be updated. Most existing algorithms broadcast control messages periodically to update this information. Typically, a higher speed of node will change the topology more quickly, thereby requiring higher frequency of broadcasting. In order to maintain the clusters and the routing information efficiently, the control messages need to be broadcast periodically according to the requirement of the potentially maximum node speed. The node speed may change from time to time. When it is low, the unnecessary broadcasting wastes energy.

In a clustered MANET, it is difficult to maintain clusters with similar sizes. Limiting the maximum size of cluster moderates the difference in cluster sizes, thereby alleviating the difference in cluster burdens. However, the cluster may still experience significant difference in sizes and thus in burdens, thereby resulting in some nodes dying prematurely. Limiting the cluster size within a strict maximum and minimum range better balances burdens among the clusters. The power consumption of maintaining these cluster sizes, especially when the speed of node is high, however, will consume significant energy. This technique also fails to optimize energy efficiency.

1.3 Objectives

Energy efficiency is a primary challenge in ad hoc networks as is evident from the previous sections. The main objective of this research is, therefore, to improve energy efficiency of the entire network to maximize the network lifetime.

1.3.1 Clustered WSNs

As the directional data traffic towards the sink not only degrades network performance but also reduces the energy efficiency of clustered WSNs, the first objective is to equalize cluster lifetime of WSNs so as to actually balance the burden among the clusters. No node or cluster will die prematurely, thereby improving both network performance and energy efficiency.

Since maintaining the CH in the central area of the cluster by re-clustering with ripple effect fails to optimize energy efficiency, the second objective is to maintain the CH in the central area of the cluster as long as possible through re-clustering without ripple effect to further improve energy efficiency.

1.3.2 Clustered MANETs

Periodic broadcasting of control messages to update intra-cluster routing information and to maintain clusters causes unnecessary broadcasting in a MANET, which wastes energy. The first objective in MANETs is to adapt the broadcasting in accordance with the node speed to reduce unnecessary broadcasting.

The uneven cluster sizes burden the clusters differently, causing the clusters with higher burdens to die prematurely. The second objective is to maintain and balance cluster sizes adaptively in accordance with the node speed. The power consumption can then be better balanced among clusters throughout the network.

1.4 Scope of research

This research focuses on energy conservation in clustered ad hoc networks. Other issues like Quality of Service (QoS) and security are not covered in this project yet may be investigated in future. Considering that the key features of a WSN and a MANET are quite different, these two types of networks are treated separately.

In traditional wireless systems, node lifetime can be extended either by: increasing the capacity of the battery; minimizing power consumption of the electronics; using efficient coding and modulation methods; or by optimizing signal processing techniques or antenna design [28]. Differing from traditional wireless systems, ad hoc networks do not have associated infrastructure and each node, typically powered by batteries with limited energy supply, acts as a router to relay data for others. When a node exhausts its available energy, it ceases to function. This shutdown may potentially result in partitioning or disconnecting the entire network.

1.4.1 Clustered WSNs

The major scope of this research is to improve energy efficiency throughout a homogeneous WSN with identical nodes and uniform node distribution. During the design and performance evaluation, only the power consumption that is relative to data aggregation and communication is considered. Power consumption of other tasks, such as sensing data, is beyond the scope of this research.

Furthermore, though some research has been conducted in WSNs with mobile sensor nodes [29-31], this research only considers a WSN in which all sensors are immovable after they are deployed.

1.4.2 Clustered MANETs

The major scope is to improve energy efficiency of the homogeneous clustered MANETs with identical nodes. During the design and performance evaluation, only the power consumption that is relative to cluster maintenance and data communication is considered. Power consumption of other tasks, such as re-sending data due to link failure, is beyond this research.

Other issues in clustered MANETs like limited bandwidth, variation in link and node capabilities, low quality issues and network scalability [3, 32] are beyond the scope of this research.

1.5 Contributions

Different to most previous research which improves energy efficiency for some special links or routes between the source node and the destination node, this thesis mainly contributes to improving energy efficiency of the entire network. In WSNs, it contributes to both balancing power consumption throughout the network and improving energy efficiency; in MANETs, it contributes to both reducing unnecessary broadcasting and balancing power consumption throughout the network.

1.5.1 Clustered WSNs

First, a clustering algorithm, which requires the sink to participate in organizing the clusters during the cluster formation when the network starts to work, is proposed to equalize cluster lifetime and to maintain CH in the central area of the cluster. This algorithm takes the directional data traffic towards the sink into account. The relationship between the traffic, power consumption and energy store in a cluster is built. An effective algorithm to equalize cluster lifetime is then proposed. Furthermore, by building the relationship between the residual node energy in the central area and the cluster lifetime, the service time as a CH of the nodes in the central area is maximized. The simulation results show that the power consumption is well balanced throughout the network, the energy efficiency is increased and the performance of delivering data to the sink is improved with this proposed algorithm.

Secondly, a clustering algorithm which combines clustering and scheduling is proposed to balance power consumption throughout network for homogeneous WSNs with high node density. Simulation results show that this proposed algorithm improves both energy efficiency of the network and performance of delivering data to the sink.

1.5.2 Clustered MANETs

A clustering algorithm for MANETs is proposed to broadcast control messages and to maintain the cluster size adaptively. First, it proposes a method to build up the relationship between the node speed and the rate that this speed changes the cluster topology. It then proposes a method to determine the needed broadcasting frequency which can actually meet the requirement of the respective node speed. The control messages are then broadcast adaptively according to this requirement, thereby reducing the unnecessary broadcasting. Secondly, the cluster size is maintained and balanced adaptively in accordance with the node speed to balance power consumption through the network. Simulation results and mathematical analysis show that the control messages are really broadcast adaptively, energy efficiency is improved and the first-node lifetime of the network is extended.

1.6 Thesis outline

This thesis is organized into seven chapters. Energy-efficient clustering algorithms for WSNs and for MANETs are investigated in Chapter 2 and Chapter 3, respectively. The organization of both chapters is similar. Following the classification of the algorithms, the major features of each category are summarized and some representative algorithms are described in detail in each category. The strengths and weaknesses of each category are then analyzed through detailed examples.

The outline of Chapters 4, 5 and 6 is shown in Fig. 1-4. Chapter 4 proposes a clustering algorithm to balance power consumption throughout the network and to improve energy efficiency of WSNs with low to medium node density. The strategy of the algorithm is first presented followed by the detailed design, in which the methods to equalize cluster lifetime and to maintain the CH in the central area of the cluster are analyzed. To evaluate the performance of the proposed algorithm, new parameters are defined and simulations are conducted. Chapter 5 proposes an algorithm to balance power consumption throughout the network with high node density by combining clustering and scheduling. The organization of this Chapter is similar to Chapter 4. Chapter 6 proposes a clustering algorithm for MANETs to reduce unnecessary broadcasting and to balance power consumption throughout the network. The strategy of the design is first presented followed by the detailed design, in which the methods to broadcast control messages and to balance cluster sizes adaptively are analyzed. To evaluate the performance, a new energy model is built and simulations and mathematical analysis are conducted.

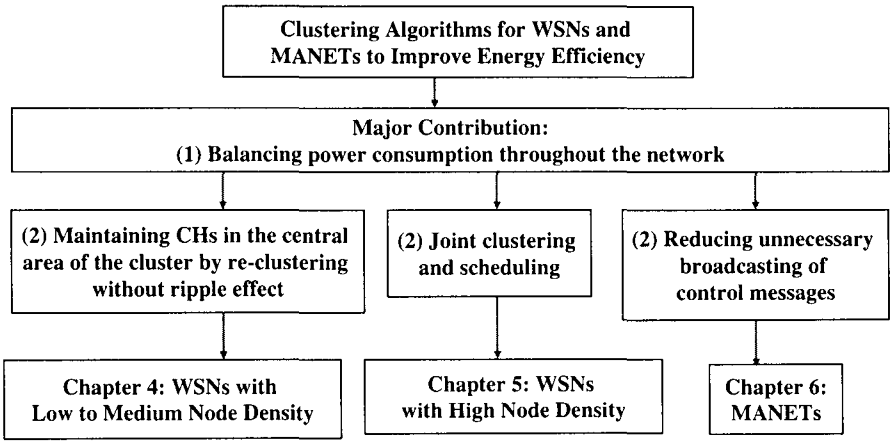


Figure 1-4. Outline of Chapters 4, 5 and 6.

Chapter 7 summarizes the contribution of the thesis, and discusses the foreseen challenges of the algorithms.

2 Review of energy-efficient clustering algorithms for WSNs

This chapter investigates existing energy-efficient clustering algorithms for WSNs. Section 2.1 classifies the algorithms and describes some representatives for each category. Section 2.2 analyzes these algorithms and addresses their weaknesses.

2.1 Energy-efficient clustering algorithms for WSNs

Energy efficiency is one of the primary challenges in WSNs. Clustering schemes strive to reduce power consumption in order to prolong the network lifespan. This chapter classifies the existing clustering algorithms into six categories [12], as shown in Table 2-1.

Table 2-1. Classification of energy-efficient clustering algorithms for WSNs.

Categories	**Objectives**
Aggregating data	To reduce the amount of data that need to be sent to the sink
Rotating the role of CH	To distribute the higher burden of a CH among the nodes
Equalizing cluster size	To balance the burden among the clusters
Locating CH in the central area of the cluster	To reduce the total power consumption of intra-cluster communication
Assigning the lowest needed power to the nodes	To minimize the power consumption of each hop
Optimizing route using different power level clusters	To select the route with the lowest energy to relay the data

As most existing energy-efficient clustering algorithms share the same energy model, this chapter describes this model for further explanation. The energy dissipated in sending an a-bit data message from a source node to a destination node can be typically expressed by (2.1) [13, 33-37], with the definition of each parameter in Table 2-2.

$$E = E_T + E_R = a \times \left(e_t + e_{amp} \times d^n\right) + a \times e_r \qquad (2.1)$$

Table 2-2. Definitions of the parameters of the energy model.

Parameters	Definitions
E	Total energy dissipated during the communication
E_T	Energy dissipated in the transmitter of the source node
E_R	Energy dissipated in the receiver of the destination node
e_t	Radio dissipation of the transmitter circuit
e_r	Radio dissipation of the receiver circuit
e_{amp}	Parameter of transmitter amplifier
d	Transmission range of the hop
n	Power attenuation ($2 \leq n \leq 4$)
a	Data amount

2.1.1 Aggregating data

The basic operation in a WSN is the systematic collection of sensed data to be eventually transmitted to a data sink for processing [7, 13]. The key challenge in such data gathering is to conserve the sensor energy to maximize sensor lifetime and thus network lifetime [38]. Data aggregation has emerged as a basic tenet in WSNs. The key idea is to aggregate data from different sensors to eliminate redundant transmissions [39]. In a clustered WSN, two typical methods to aggregate data are proposed: one is to require a CH to aggregate data when it has collected all data from the member nodes before the inter-cluster communication occurs [13, 34, 40, 41]; the other is to aggregate data over each passing hop [35, 38].

Low-Energy Adaptive Clustering Hierarchy (LEACH) applies local data aggregation to reduce global communication. This algorithm requires each CH to aggregate data collected from member nodes before it sends them to the sink. During the data communication process, a CH keeps its receiver ‘on’ to receive all the data from the member nodes in the cluster. When all the data have been received, a CH performs signal processing functions to aggregate the data into a single signal. For example, if the data are audio or seismic signals, a CH can aggregate the individual signals to generate a composite signal [13].

Unlike LEACH [13], Chen [35] proposes an algorithm to aggregate data over each hop. This algorithm assumes that N nodes are randomly deployed in the network. Chen proposes the concept of entropy to evaluate the relation of the data generated by one sensor and the other sensors. A higher value of entropy means more correlative data. The information that is provided only by the sensor S_i can be expressed by entropy (2.2):

$$H(S) - H(S \cap \overline{S_i}) \tag{2.2}$$

where $H(S)$ is the entropy of the total WSN and $H(S \cap \overline{S_i})$ is the entropy of the total network excluding sensor S_i. The coefficient c_i is defined as the fraction of unique information of sensor S_i compared to $H(S_i)$ (2.3):

$$c_i = \frac{H(S) - H(S \cap \overline{S_i})}{H(S_i)}, \ 0 < c_i \leq 1 \tag{2.3}$$

This coefficient c_i expresses the degree of correlation between a sensor and its neighboring sensors. The data can then be aggregated in accordance with this coefficient as they are relayed hop by hop. The data from two sensors have almost no correlation when they are located far away from each other.

2.1.2 Rotating the role of CH

In a clustered WSN, a CH has a higher burden than that of member nodes. This higher burden will consequently make it drain energy much more quickly than member nodes. Rotating the CH role distributes this higher burden among the nodes, thereby preventing the CH from dying prematurely [13, 34, 36, 40, 42-46].

LEACH employs the technique of randomly rotating CH role among all the nodes [13]. The operation of LEACH is organized in rounds where each round consists of a set-up phase and a transmission phase. During the set-up phase, the nodes organize themselves into clusters with one node serving as CH in each cluster. The decision to become a CH is made locally within each node and a predetermined percentage of the nodes serving as CH in each round. During the transmission phase, the CHs collect data from member nodes within their respective clusters and apply data aggregation before forwarding them to the data sink over a single-hop route. At the end of a given round, a new set of nodes with higher residual energy become CHs for the subsequent round. The higher burden of a CH is therefore distributed among the nodes. It has been shown that LEACH provides significant energy savings and prolongs network lifetime over clustering schemes with fixed CHs.

Compared to LEACH [13], the algorithm by Liu [36] better balances power consumption among the cluster by balancing the cluster sizes. The nodes in a WSN with identical nodes have the capability of adjusting their transmission power. They therefore can determine the number of neighboring nodes by adjusting their transmission range. If a node has too many neighboring nodes, it will reduce its power to reduce its neighbors. Alternatively, if neighboring nodes are lacking, it will increase its power to attract more. After the nodes are connected with their neighbors, they will be grouped into clusters. The node with the most residual energy in each cluster is selected to be CH of that cluster and to form the backbone of the network. Because a CH drains energy quickly, its role is rotated. When a CH period expires, the CH will become a member node and the node with maximum residual energy in that cluster becomes CH for the consequent CH period. Figure 2-1 is used to explain this algorithm. The neighboring relationship is first established according to one hop distance range and the number of the neighboring nodes is controlled by adjusting the transmission power of the node (Fig. 2-1(a)). Next, the nodes in the network are partitioned into clusters and the node with the highest residual energy is selected as the CH for that cluster (Fig. 2-1(b)). During each CH period, the node with the highest residual energy in each cluster is selected as the new CH (Fig. 2-1(c)). The higher burden of CH is therefore distributed among the nodes in the cluster.

Muruganathan [34] proposes a centralized clustering algorithm to better organize clusters. Similar to LEACH, each round of data collection in this centralized algorithm also consists of cluster set-up phase and data communication phase. During each set-up phase, the nodes send their residual energy information to the data sink. On reception of the energy information, the data sink calculates the average residual energy of the nodes and chooses a set of nodes whose residual energy is above the average residual energy. After this set is determined, the network is split into clusters, the number of which is predetermined. The higher burden of CH is thus distributed among the nodes.

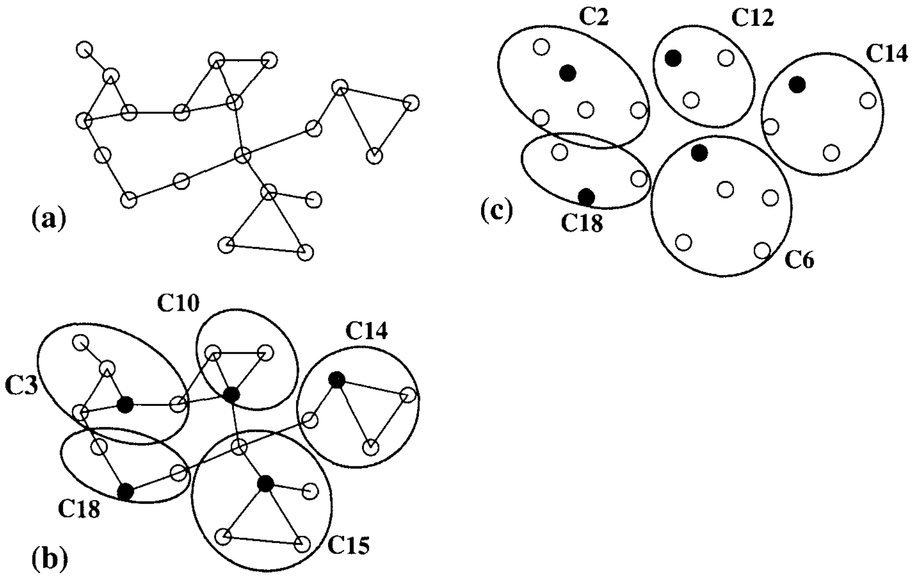


Figure 2-1. Energy-efficient clustering protocol (Adopted from [36]).

2.1.3 Equalizing cluster sizes

Rotating the CH role balances power consumption among the nodes. Power consumption can be further better balanced if the clusters have similar burdens. Many clustering algorithms therefore assign (approximately) the same number of nodes to each cluster or CH [34, 47-51].

During the cluster formation phase, base-station controlled dynamic clustering protocol [34] by Muruganathan first selects the nodes with higher residual energy to be CHs and then splits the network into clusters with similar size. After the CHs are determined, the network is split into two sub-clusters, and these sub-clusters are further split into smaller sub-clusters. The base station repeats the cluster splitting process until the desired number of clusters is attained. The iterative cluster splitting algorithm ensures that the selected CHs are uniformly placed throughout the network by maximizing the distance between CHs in each splitting step. Furthermore, in order to evenly distribute the load among all CHs, each cluster will have approximately the same number of sensor nodes so that the cluster sizes are equalized.

Krishman [49] proposes 'Rapid' and 'Persistent' algorithms, as shown in Fig. 2-2, to design the clusters with expected sizes.

In the 'Rapid' algorithm, a cluster size limit of *B* units (including the CH) is assigned to each CH. The CH distributes (*B-1*) units among its neighbors. Fig. 2-2(a) provides an example when *B* equals six. In case the CH distributes a bond which is beyond the total number of the reachable neighbors of the corresponding node, the size of the cluster will be less than expected. Figure 2-2(b) provides an example when the cluster size is expected to be nine. However the CH assigns one node four units which in fact cannot find any reachable node, resulting in a cluster size of only six. The 'Rapid' algorithm is improved by the 'Persistent' algorithm, which can acquire the cluster size as desired. In the 'Persistent' algorithm, if the node is assigned more units of nodes that can be reached by one hop or more hops, it will send back this information to the CH. In Fig. 2-2(b), the node that is assigned four units yet cannot find any neighbors will send the residual three units back to the CH, which will re-distribute these three units to other nodes and the cluster size nine is guaranteed, as shown in Fig. 2-2(c). Thus the cluster sizes can be equalized.

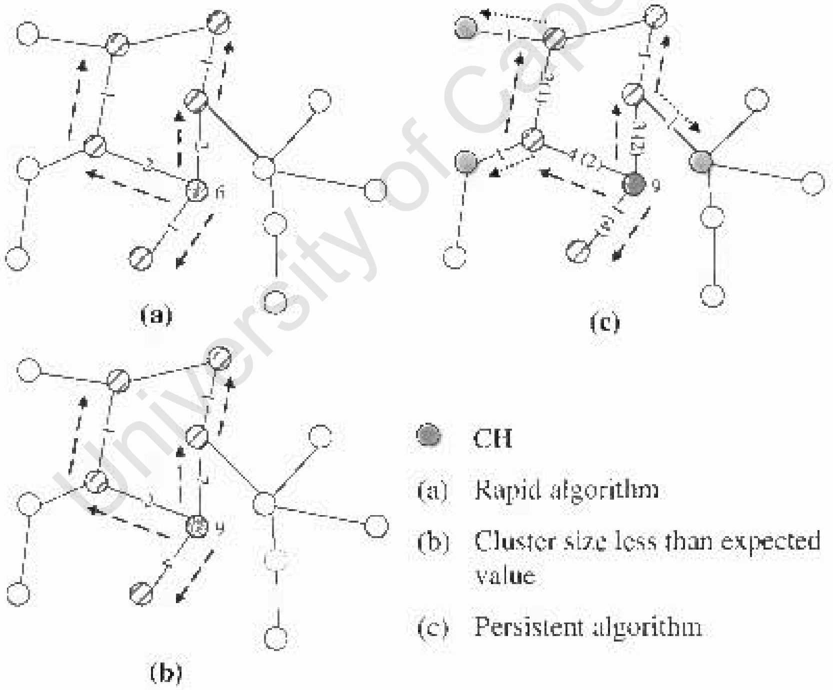


Figure 2-2. Rapid and Persistent algorithms (Adopted from [49]).

2.1.4 Locating CH in the central area of the cluster

The total power consumption of a cluster is also relative to the location of the CH in the cluster. Selecting a node in the central area of a cluster as CH [13, 14, 35, 47] reduces the total power consumption of intra-cluster communication. In a multi-hop cluster, in which a member node may send data to the CH by a multi-hop route, the total number of hops between the member nodes and the CH is minimized, thereby saving energy for relaying data. In a single-hop cluster, in which all member nodes send data to the CH by a single-hop route, the sum of squares of the distances between the member nodes and the CH is reduced, thereby reducing the power consumption of transmitting data.

Chen [35] proposes an algorithm which minimizes the total number of hops from member nodes to the CH. Fig. 2-3 shows his method to select CH. After the clusters are formed, each node in the cluster calculates how many hops that it needs to access any other node in the cluster. The numbers in Fig. 2-3 mean the minimum number of hops needed by that node to reach any other node within that cluster. The sensor with a number two can reach all other sensors within two hops, while a sensor with a number four requires as many as four hops to reach all other sensors. After all the nodes acquire the information about their respective minimum number of hops to reach any other node, they will exchange this information so that all nodes know the minimum number of hops of other nodes. The node with the smallest number is selected as the CH, thereby minimizing the total number of hops between the member nodes and the CH. Yet the flooding caused by counting the number of hops to others and by exchanging such information will consume significant energy.

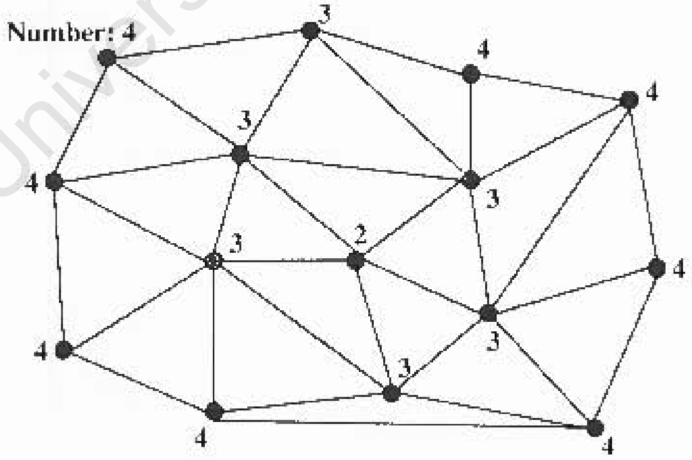


Figure 2-3. Sensors with maximum hop number to any other sensor (Adapted from [35]).

Besides optimization of CH selection, Ghiasi [47] further considers optimizing the cluster formation to minimize the sum of the squares of the distances between the member nodes and the CHs overall the network. Fig. 2-4 shows the optimization of the cluster formation. Eight nodes are organized into two clusters. Each cluster has four nodes. Fig. 2-4 shows two pairs of clustering options named A-B and C-D. The sum of the squares of the distances between the member nodes and the CH in clusters A-B is smaller than that in clusters C-D. According to (2.1) when $n=2$, energy can be saved by the A-B partition. The above minimization is subject to the constraint that the power consumption of the CHs can be maintained to be approximately equal to each other.

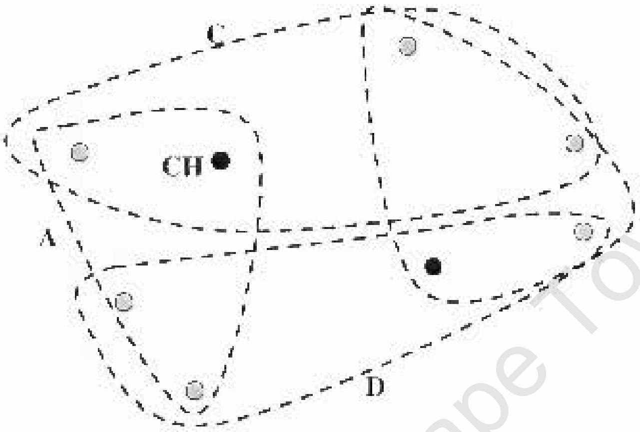


Figure 2-4. Clustering nodes to reduce intra-cluster power consumption (Adopted from [47]).

2.1.5 Assigning the lowest needed power to the nodes

The power consumption of each link of the algorithm in this category is minimized by assigning (i) to the member nodes or the CHs the lowest power needed for intra-cluster communication, and (ii) to the gateway nodes the lowest power needed for inter-cluster communication.

The algorithm in [52] assumes that the node has the capability of adjusting the transmission power so that the lowest necessary power of each link can be assigned to different nodes. It first brings together the closer nodes in the network to reduce the transmission ranges. Then it assigns each node in the cluster with the lowest possible transmission power while keeping the intra-cluster connection. The pair of nodes that have the minimum distance between any two clusters are chosen as gateway nodes. The communication between the CHs is realized through these nodes. The last step is to assign the gateway nodes the lowest possible transmission power to maintain the connection of the backbone network. By assigning these minimum needed transmission powers to the nodes, the total power consumption of the route is reduced.

2.1.6 Optimizing routes by using different power level clusters

Most clustering algorithms assume a WSN with uniform node distribution. The sensor nodes, however, are not always uniformly dispersed in the network. Different areas in the WSN may have different node density. The nodes in one area may access each other by using a low transmission power through multiple hops. However, if the nodes in this area want to access the nodes in another area, they may not find any other nodes to relay the data so that they need a higher transmission power to access that area. The nodes may therefore be organized into clusters with different power levels. By optimizing the routes consisting of different power levels, energy can be saved.

Kawadia [53] groups the nodes into different power level clusters, as shown in Fig. 2-5. The highest power level (100mW) is needed to connect all the nodes in the entire network through multiple hops. A lower power level (1mW or 10mW) will form clusters for only the nodes that are close enough to be connected by multiple hops using this lower power level. Each node may belong to different clusters of different power levels so that several routes are possible by taking different combinations of these power levels for each hop. Each node along the route finds the next hop using the lowest transmission power level such that the destination is reachable under the condition that the remaining hops cannot exceed this lowest transmission power level. Energy is then saved by optimizing these routes from source to destination.

2.2 Analysis of existing energy-efficient clustering algorithms for WSNs

Though the existing clustering algorithms have improved energy efficiency for WSNs, they still have some weaknesses as analyzed below.

The algorithms that aggregate data reduce the data that need to be sent, thereby reducing energy of routing data to the sink. For a CH to aggregate data before it sends data to the next hop or the sink, the CH needs some capability to perform data aggregation. Requiring each node along the route to aggregate data may present higher energy efficiency than just requiring a CH to aggregate data as it may more efficiently aggregate the data. However, this requires each node to evaluate the entropy (correlation) with other nodes, which demands the nodes to have more powerful processing capability. Furthermore, this process will also consume some energy.

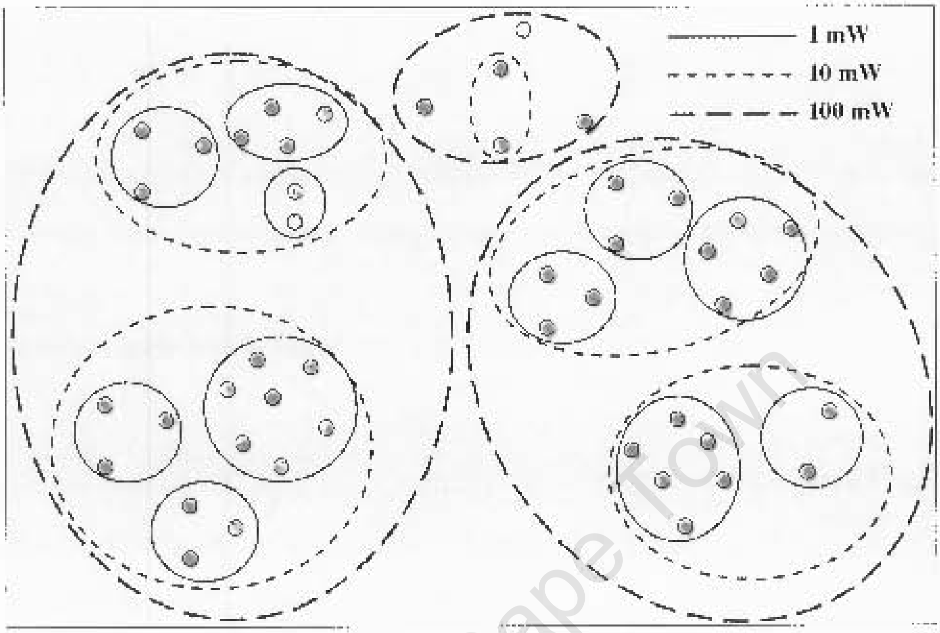


Figure 2-5. Group the nodes into different power level clusters.

As the CHs have higher burdens than member nodes, fixing the CHs throughout the lifetime makes these CHs die quickly, ending the useful lifetime of all nodes belonging to those clusters [13]. Rotating the CH role distributes this higher burden among the nodes, thereby preventing the CHs from dying prematurely. This CH rotation significantly extends the lifetime [13].

The node in a larger sized cluster usually has higher burden than the one in a smaller sized cluster. If it is a member node, it needs higher transmission power to reach the CH in a single-hop cluster or more hops to reach the CH in a multi-hop cluster. If it is a CH, it needs to receive, aggregate and transmit more data. Balancing the burdens of the clusters prevents the nodes with higher burdens from dying prematurely. However, the existing algorithms which assign (approximately) the same number of nodes to the clusters or CHs fail to optimize the energy efficiency because they do not take the directional data traffic towards the sink into account. This problem is to be analyzed in detail in Section 2.2.1.

Locating the CHs in the central area of the cluster reduces power consumption of intra-cluster communication. However, most algorithms locate the CH in the central area of the cluster by frequently re-clustering the nodes with ripple effect. The flooding caused by this frequently re-clustering may consume significant energy [13], thereby failing to optimize energy efficiency.

Assigning to each node the lowest needed power reduces the power consumption of each link. With the location information of the nodes, the transmission power can be efficiently assigned to each link. This algorithm can be applied in WSNs with the nodes being aware of their location information and having the capability to adjust transmission power.

Organizing the nodes into clusters with different power levels in WSNs with uneven node distribution not only improves energy efficiency by optimizing the different power lever hops, but also increases network capacity. This algorithm, therefore, can be used for energy saving in WSNs with uneven node distribution and limited bandwidth.

All the above algorithms improve energy efficiency of WSNs, though most of them have limitations. Energy efficiency of WSNs, however, can be further improved if the following two features are considered: the directional data traffic towards the data sink and the different power consumption when the CH is located in different areas in a cluster.

2.2.1 Directional data traffic towards data sink

In a WSN, all data generated by the sensor nodes need to be sent to the sink for further analysis, resulting in data traffic towards the sink. This directional data traffic burdens nodes/clusters differently. The nodes/clusters with higher burdens in a WSN formed by identical nodes will die prematurely [54, 55, 57, 58]. If a CH sends data to the sink over a single-hop route, the clusters that are farther away from the sink will die earlier than other clusters that are nearer the sink as they need higher transmission power to reach the sink. If a CH sends data to the sink over a multi-hop route, the clusters that are nearer the sink will die earlier than other clusters that are farther away from the sink as they need to relay more data from other clusters. The premature node death takes some areas out of coverage, thereby degrading the network performance. The one-dimensional linear WSN shown in Fig. 2-6 is used to explain the impact of the directional data traffic towards the sink.

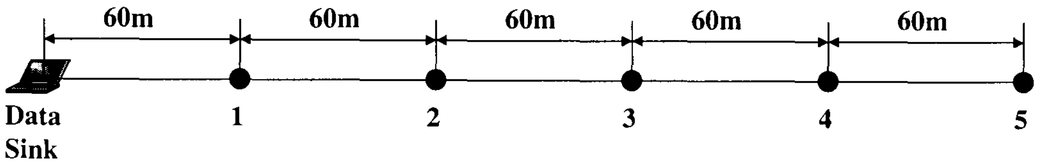


Figure 2-6. One-dimensional linear WSN.

Five identical nodes are evenly distributed in the network with a distance of 60m between two neighboring nodes. The initial energy of each node is 0.5J. The lifetime of the linear network is measured by the number of rounds. During each round, each node sends a packet of 50 bytes to the sink either by a single-hop route or by a multi-hop route. The node is assumed to be dead when its residual node energy is lower than 0.01J. The energy model described in (2.1) is used to evaluate the lifetime. As the value of e_{amp} greatly affects the power consumption of sensor nodes and different e_{amp} values have been used in some references [14, 34, 37], the author analyzes the power consumption below by using different e_{amp} values and sets n=2.

Single-hop route

Table 2-3 shows the lifetime of each node when it sends data to the sink over a single-hop route. The node lifetime increases when the node gets closer to the sink. The node nearest the sink has the longest lifetime which is about 22 times that of the node farthest away from the sink when e_{amp}=100 pJ/b/m^2. When e_{amp} increases, the nodes need more energy to send data to the sink so that the node lifetime decreases.

Table 2-3. Node lifetime (rounds) (single hop).

Node	e_{amp}=100 pJ/b/m^2	e_{amp}=50 pJ/b/m^2	e_{amp}=10 pJ/b/m^2
1	2988	5327	14245
2	832	1591	6315
3	373	734	3726
4	211	419	1957
5	136	270	1290

Multi-hop route

Table 2-4 shows the lifetime of each node when it sends data to the sink over a multi-hop route. The nodes keep the transmission range of 60m when they are transmitting data. The node lifetime decreases when the nodes get closer to the sink. The node farthest away from the sink has the longest lifetime which is more than 25 times that of the node nearest to the sink when e_{amp}=100 pJ/b/m^2. When e_{amp} increases, the nodes need more energy to relay data to the sink so that the node lifetime decreases.

Table 2-4. Node lifetime (rounds) (multiple hops with the same transmission range).

Node	e_{amp}=100pJ/b/m^2	e_{amp}=50pJ/b/m^2	e_{amp}=10pJ/b/m^2
1	545	908	1945
2	685	1145	2480
3	922	1551	3422
4	409	2402	5519
5	2988	5327	14254

However, keeping a transmission range of 60m cannot guarantee all data packets to be delivered to the sink. When node 1 dies prematurely, no packet can be received by the sink if node 2 still transmits data with a transmission range of 60m. To make sure that the sink receives the data, when node 1 dies node 2 needs transmission range of 120m (at least). This process is iterative and finally when node 4 dies, node 5 needs a transmission range of 300m. The node lifetime in this scenario is shown in Table 2-5.

Table 2-5. Node lifetime (rounds) (multiple hops with necessary transmission ranges).

Node	e_{amp}=100 pJ/b/m^2	e_{amp}=50 pJ/b/m^2	e_{amp}=10 pJ/b/m^2
1	545	908	1945
2	585	986	2230
3	629	1073	2578
4	686	1187	3078
5	790	1396	4088

When e_{amp}=50 pJ/b/m^2, the first node dies at 270 rounds in the single-hop network (see Table 3) whereas at 908 rounds in the multi-hop network (see Table 5). If it is assumed that the network dies when 50% of the areas in the network are out of coverage (as the performance of the network is greatly degraded), the network dies at 734 rounds in the single-hop system whereas at 1073 rounds in the multi-hop system. Sending data to the sink over the multi-hop route extends the lifetime.

The results in Table 2-3 and Table 2-5 show that the lifetime of different nodes varies according to their distance to the sink. If the lifetime of all nodes is equalized, no node will die prematurely. Furthermore, as transmission range of the node is limited, it is infeasible to require all nodes to send data to the sink over a single-hop route in a large sized WSN. In Fig. 2-6, if the maximum transmission range of the nodes is limited within 150m, when node 2 dies, node 3 cannot reach the sink any longer when the nodes send data to the sink over multiple hops, causing the connectivity between the network and the sink to be lost. The residual energy in node 3, 4 and 5 is wasted. Therefore, in a WSN-especially in a large sized WSN-equalizing node lifetime improves both energy efficiency and network performance.

If each node in this linear system represents a cluster, as shown in Fig. 2-7, the lifetime of the nodes (cluster) can be equalized by allocating energy to each node (cluster) in accordance with its burden. In a clustered WSN, this can be realized by assigning different numbers of nodes to each cluster in accordance with its burden. This also means that in a homogeneous WSN with identical nodes and uniform node distribution, equalization of the cluster lifetime will make the clusters with different sizes.

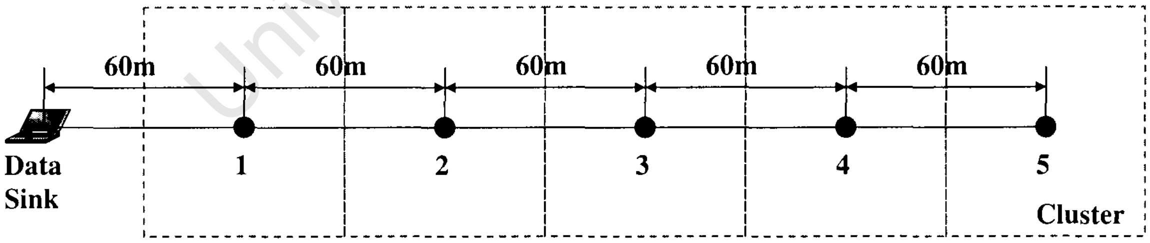


Figure 2-7. Each node in the linear system represents a cluster.

Equalizing node lifetime by allocating energy to each node in accordance with its burden

Using $\sum P$ to express the total power consumption of a node, which includes power consumption of transmitting (P_{tr}), receiving (P_{re}) and relaying (P_{relay}) data in this case, and

$E_{initial}$ to express the initial energy of that node, to equalize node lifetime, (2.4) must be observed, in which *const* means a constant.

$$\frac{E_{initial}}{\sum P} = \frac{E_{initial}}{P_{tr} + P_{re} + P_{relay}} \approx const \tag{2.4}$$

The system in Fig. 2-6 has five nodes with each node has 0.5 J energy. Its total energy is thus 2.5J. Table 2-6 shows the equalized lifetime of each node and the allocated energy to each node in the multi-hop system.

As previously defined that the network dies when 50% of nodes run out of energy, when e_{amp}=50 pJ/b/m^2, compared to single-hop system and multi-hop system which allocate each node the same energy, equalizing the node lifetime increases the network lifetime by about 111% and 45%, respectively.

Table 2-6. Equalized node lifetime (rounds) and re-allocated energy (J).

Node	e_{amp}=100 pJ/b/m^2		e_{amp}=50 pJ/b/m^2		e_{amp}=10 pJ/b/m^2	
	Lifetime	Allocated Energy	Lifetime	Allocated Energy	Lifetime	Allocated Energy
1	929	0.8459	1564	0.8545	3452	0.8799
2	926	0.6729	1559	0.6772	3441	0.6899
3	922	0.5000	1551	0.5000	3422	0.5000
4	912	0.3271	1533	0.3227	3379	0.3100
5	883	0.1541	1474	0.1456	3201	0.1201

Related work

Some existing clustering algorithms also take the directional data traffic into account and design clusters to make them have similar lifetime.

Ye [40] improves LEACH [13] by assigning fewer nodes to the CHs farther away from the sink so that this higher burden is compensated. Unfortunately, to send the data to the sink over a long single hop is infeasible or not energy efficient, especially in a large sized WSN.

Muruganathan [34] requires the CH to send data to the sink over a multi-hop route. This algorithm organizes clusters with similar sizes. By randomly selecting a CH along the route from the source to the sink to send the data to the sink using a single-hop route, the higher burdens of the clusters nearer the sink are alleviated. However, this algorithm requires the nodes to have high enough transmission power so that any CH along the route can reach the sink using a single-hop route, which limits its application in large sized WSNs.

The directional data traffic towards the sink is also accommodated in [37] by using a chessboard configuration for heterogeneous WSNs formed by different sensor nodes. The network is separated into clusters (grids) as shown in Fig. 2-8. Each grid has a supernode (CH) which has more initial energy than the member nodes. Within each grid, a member node nearer the CH will have a higher burden than the one farther away from the CH since it needs to relay more data from other member nodes. The chessboard configuration separates the network lifetime into two periods. In the first period, it schedules the CHs in the grids with darker (lighter) shade to be active. After these CHs run out of energy, the first period expires. The CHs in the grids with lighter (darker) shade are active when the second period initiates. The nodes which are closer to the CH with higher burden in the first period are farther away from the CH in the second period, thereby distributing the higher burden. This algorithm is suitable for heterogeneous WSNs which have some more powerful nodes than others to be CHs.

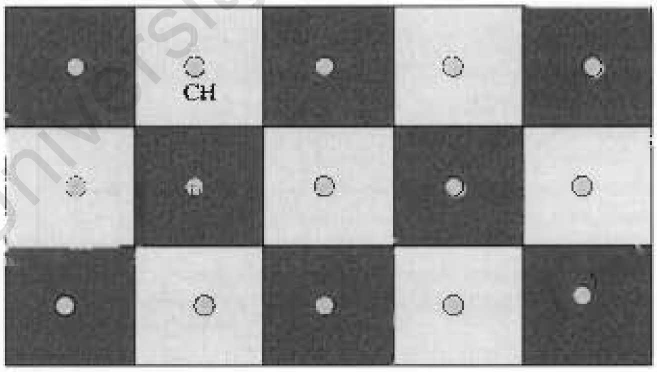


Figure 2-8. Chessboard configuration (Adapted form [37]).

Different to the stable nodes and the data sink in [34, 37, 40], Ma [59] and Lin [60] consider using mobile CHs or a mobile sink to balance the power consumption of the nodes. The nodes which are nearer the CHs or the sink at one time may be farther away from them at the next stage, thereby distributing the higher burden.

2.2.2 Power consumption and location of CH

The directional data traffic burdens the nodes/clusters differently. The selection of CH also affects the power consumption of the network. As previously stated, power consumption can be diminished by locating the CH in the central area of the cluster. This subsection uses examples to analyze it in more detail.

Intra-cluster communication

Fig. 2-9 shows a cluster with 20 nodes distributed within an area of 60m×60m. Each node is allowed to be CH and others are member nodes. When any node becomes CH, each member node will send a packet with 50 bytes to it. The total dissipated energy of intra-cluster communication E_{total}, which only considers sending, receiving and aggregating data, is expressed in (2.5).

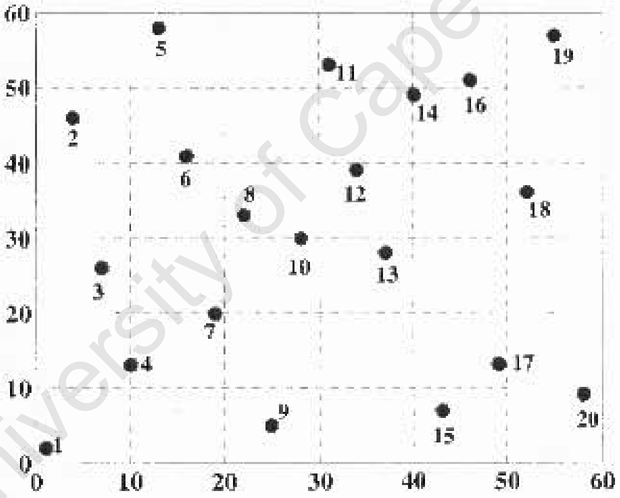


Figure 2-9. Cluster to analyze power consumption when different node becomes CH.

$$E_{total} = a \times \left(\sum_{N-1} \left(e_t + e_{amp} \times r_i^{2} \right) + (N-1) \times e_r + N \times e_{Agg} \right) \tag{2.5}$$

in which a means the packet size, N means the number of nodes in the cluster, r_i means the hop distance from the member node to the CH, e_{Agg} means the radio dissipation of aggregating data, $a \times \sum_{N-1} \left(e_t + e_{amp} \times r_i^{2} \right)$ is the energy consumed by sending data,

$a \times (N-1) \times e_r$ is the energy consumed by receiving data and $a \times N \times e_{Agg}$ is the energy dissipated by aggregating data.

The total dissipated energy when different nodes become CH is shown in Fig. 2-10. It can be seen that when node 8, 10, 12, or 13 as shown in Fig. 2-9 becomes CH, the total energy dissipation of the cluster is lower than that when other node like 1, 2, 19 or 20, which is at the edge of the cluster, becomes the CH.

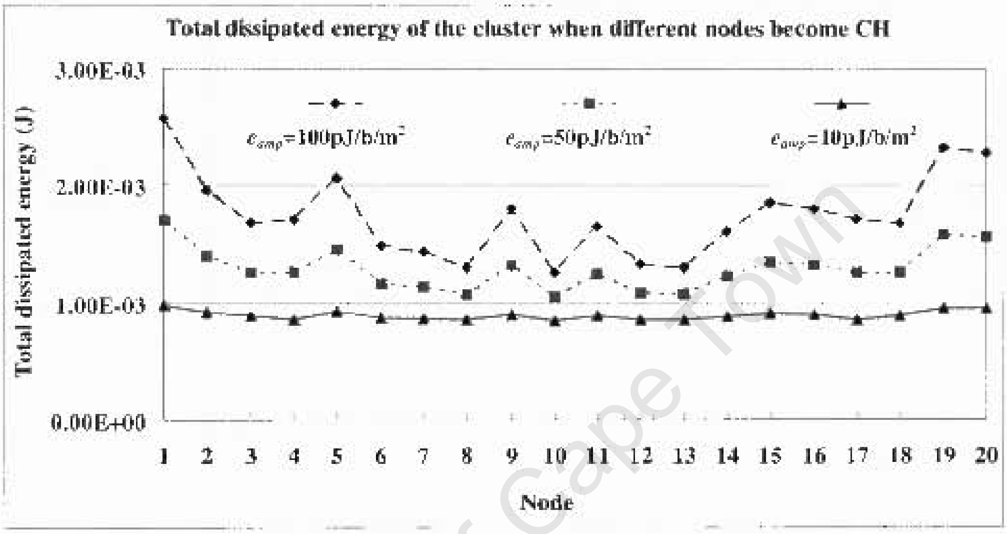


Figure 2-10. Total dissipated energy of intra-cluster communication when different nodes become CH.

It can therefore be determined that energy is saved by locating the CH in the central area of the cluster. However, the improvement needs to be further evaluated according to different e_{amp} values.

Inter-cluster communication

The linear cluster, as shown in Fig. 2-11, is used to analyze inter-cluster communication. CH-A is sending an a-bit data message to its neighboring CH, CH-B. Two scenarios are compared: CH-B is at center of the linear cluster or it is at anywhere in the linear cluster.

The energy of sending the message to the next hop when CH-B is at the centre of the linear cluster with the hop distance of (r+m/2) is (2.6):

$$E = a\times\left(e_t + e_{amp}\times\left(r+\frac{m}{2}\right)^2\right) + a\times e_r = a\times(e_t+e_r) + a\times e_{amp}\times\left(r+\frac{m}{2}\right)^2 \qquad (2.6)$$

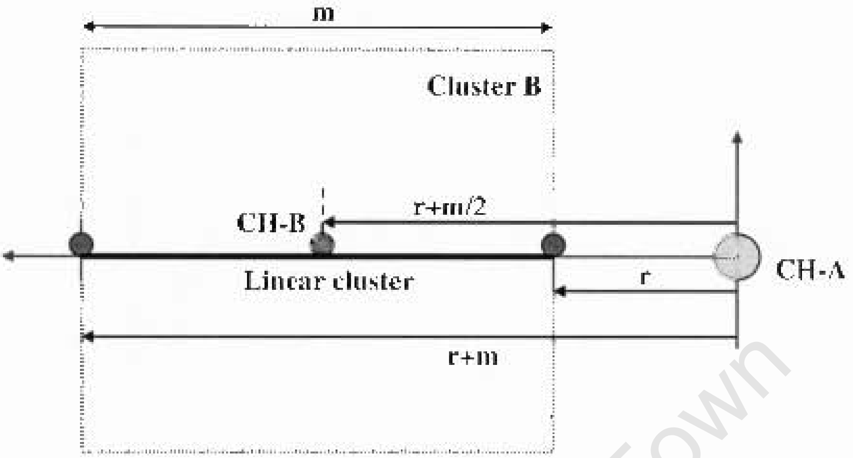


Figure 2-11. A linear cluster to analyze energy dissipated in inter-cluster communication.

The expected dissipated energy of sending this message to the next hop when CH-B is randomly located in the linear cluster is (2.7):

$$\overline{E} = \frac{\int^{r+m}\left(a\times(e_t + e_{amp}\times x^2) + a\times e_r\right)dx}{m} = a\times(e_t+e_r) + a\times e_{amp}\times\frac{1}{3m}\left((r+m)^3 - r^3\right) \qquad (2.7)$$

Locating the CH-B at the centre of the linear cluster saves energy over randomly located the CH-B in the linear cluster by (2.8):

$$\overline{E} - E = a\times e_{amp}\times\left(\frac{1}{3m}((r+m)^3 - r^3 - \left(r+\frac{m}{2}\right)^2\right) = \frac{a\times e_{amp}\times m^2}{12} \qquad (2.8)$$

Thus power consumption of inter-cluster communication is also reduced by locating the CH in the central area of the cluster.

Related work

Some clustering algorithms have considered maintaining the CH in the central area of the cluster, such as in [14, 35]. However, this maintenance is realized by frequently re-clustering the nodes with ripple effect. The flooding caused by this frequent re-clustering with ripple effect may consume more energy than the improvement achieved by maintaining the CH in the central area of the cluster [14], thereby failing to optimize the energy efficiency. On the other hand, if the CH can be maintained in the central area of the cluster by re-clustering the nodes without ripple effect, energy efficiency can be improved.

2.3 Summary

Existing energy-efficient clustering algorithms for WSNs have been investigated in this chapter. These algorithms are classified into: aggregating data, rotating the role of CH, equalizing cluster sizes, locating CH in the central area of the cluster, assigning the lowest needed power to the nodes and optimizing routes using different power level clusters. The key features of each category were summarized and some representative algorithms were described in detail. The strengths and weaknesses of each were analyzed thereafter. The important feature of directional data traffic towards the data sink in WSNs was emphasized. Energy can be saved by equalizing cluster lifetime in WSNs. The relationship between power consumption of the intra- and inter-cluster communication and the location of CH was also discussed. Energy can be saved by maintaining the CH in the central area of the cluster through re-clustering without the ripple effect. These two issues will be taken into account in the proposed clustering algorithm in Chapter 4.

3 Review of energy-efficient clustering algorithms for MANETs

This chapter investigates existing energy-efficient clustering algorithms for MANETs. Section 3.1 classifies the algorithms and describes some representatives for each category. Section 3.2 analyzes these algorithms and summarizes their strengths and weaknesses.

3.1 Energy-efficient clustering algorithms for MANETs

Mobile nodes in a MANET normally depend on battery power during operation. The energy limitation therefore poses a severe challenge for network performance [12, 61, 62]. A MANET should strive to improve energy efficiency in order to prolong the network lifespan. This chapter classifies the existing energy-efficient clustering algorithms for MANETs into four categories, as described in Table 3-1.

Table 3-1. Classification of energy-efficient clustering algorithms for MANETs.

Categories	Objective
Rotating the CH role	To distribute the higher burden of the CH among the nodes to better balance power consumption
Limiting cluster size	To alleviate the difference in burden caused by unbalanced cluster sizes
Reducing power consumption of cluster maintenance	To more efficiently maintain the clusters to reduce power consumption of cluster maintenance
Locating CH in the central area of cluster	To reduce the power consumption of intra-cluster communication

3.1.1 Rotating the CH role

As with the CH in WSNs, a CH in MANETs is more likely to die first. The lack of mobile nodes due to energy depletion may cause network partition and communication interruption [63]. Hence, it is important to balance power consumption among mobile nodes to avoid premature node failure.

The movement of the nodes in a MANET already makes the topology unstable. The frequent rotation of CH role will further exacerbate the stability of the cluster and thus the network, though it distributes the higher burden of CH among the nodes. Furthermore, the flooding triggered by exchanging node information when the rotation occurs also consumes some energy. To achieve the tradeoff between the stability of the network and the balance of the power consumption among the nodes, many algorithms prolong the service time of a CH by defining some special rules [64-71]. The rotation process is triggered only when the conditions of these rules are met.

The mobility- and energy- aware clustering algorithm [64] by Xu selects CHs and rotates the role of CH according to the mobility attributes and the energy attribute. The mobility attribute is the sum of the neighboring time between the node and each of its current neighbors. If a node has a large number of neighbors and it has been with these neighbors for a long time, this attribute will have a large value, which indicates that it is a stable node in the mobility sense. Energy attribute means the residual node energy. A higher value of energy attribute means a more stable node with respect to energy. When the roles of the nodes are determined, the information relating to the node ID, the mobility attribute and the energy attribute is tabled and updated. The member node keeps the ID of the respective CH and the CH keeps the IDs of the respective member nodes. When no such information is found, which means that a CH cannot find any member node and a member node cannot find any CH, re-clustering is triggered. The node with a relatively higher mobility attribute and higher energy attribute in its area will be selected to be CH. When the connectivity among the nodes is set, each node will sort its neighbors according to the mobility attribute. Those that are lower than the threshold will not be considered for further CH selection. The node with the highest residual energy among the nodes with higher mobility attribute than the threshold will be a potential CH. The higher burden of a CH is therefore distributed among the nodes.

The weighted clustering algorithm by Chatterjee [65] also defines some conditions for selecting CHs and rotating their roles. Unlike the algorithm by Xu [64], which only considers the node speed and residual node energy, the weighted clustering algorithm defines four parameters, named Δ_v, D_v, M_v and P_v to select CHs and to rotate their roles. It uses Δ_v, the degree-difference of the node v, to measure the number of neighboring nodes of v; uses D_v, the sum of the distances between node v and all its neighbors, to measure the energy efficiency in intra-cluster communication; uses M_v, the average speed of node v, to measure the stability of v; and uses P_v, the period that node v serves as CH, to measure the stability of the cluster. The values of Δ_v, D_v, M_v and P_v are kept updated. Combining these four parameters, the equation to select CHs and to rotate their roles can be determined, as shown in (3.1):

$$W_v = w_1\Delta_v + w_2 D_v + w_3 M_v + w_4 P_v \tag{3.1}$$

where W_v is the combined weight, w_1, w_2, w_3 and w_4 are the weighting factors of Δ_v, D_v, M_v, and P_v respectively. The node with the lowest value of W_v in its local area is selected to be CH. The factors of these four parameters are adjustable according to different application purposes and requirements. In a network that needs to maintain the cluster sizes within exact values, $w_1\Delta_v$ may play an important role. In a network that needs to maximize the network lifespan, w_2D_v is a major concern. In a network that needs stable CHs, w_3M_v needs to be carefully considered. In a network that needs stable clusters to reduce the frequency of CH rotation, w_4P_v is crucial. The CH election algorithm is invoked at the very beginning of cluster formation or when the current CHs are not able to cover all mobile nodes. The CHs are therefore rotated and the higher burdens of CHs are distributed.

The algorithm by Amis [66] assigns each mobile node a variable virtual ID (VID). The value of VID is set as its ID number when the network begins to work. The mobile node with the highest ID in its local area becomes CH. The time period that a CH serves a cluster (CH period) is limited. When this period expires, a CH resets its VID to 0 and it becomes a member node. If two CHs move into the reachable range with each other, the one with the lower VID becomes a member node. Each member node keeps a circular queue for its VID and shifts the VID value by one every CH period in one direction. Therefore, when a CH resigns, a member node with the largest VID value in the neighborhood can resume the CH function and the higher burden of a CH is distributed among the nodes.

The selection and rotation of CHs in the vote-based clustering algorithm [67] by Li are realized by considering the number of neighboring nodes and the residual node energy. Each node calculates its own vote that is the weighted sum of its normalized number of neighbors and its normalized residual energy. The node with the highest vote over its neighbors will be periodically selected as CH.

3.1.2 Balancing power consumption among CHs or clusters

A data sink is unnecessary for a typical MANET. MANETs consequently do not have the feature of directional data traffic towards the sink. In a clustered MANET, organizing the nodes into clusters with similar sizes actually balances power consumption among the clusters. However, the movement of the nodes may change the size of the cluster and thus the cluster load. To alleviate the difference in loads, some clustering algorithms limit the maximum size of the cluster [72]. When the cluster size reaches the limited value, no node is allowed to join. Some not only limit the maximum size, but also restrict the minimum size of the cluster [73-76]. When the cluster size exceeds the pre-defined maximum value or is smaller than the pre-defined minimum value, a re-clustering procedure is invoked. Some algorithms balance the burdens among the CHs by taking the residual energy of the CH into account [77, 78]. The number of nodes assigned to each CH is determined by the residual energy of the CH.

Ohta [73] proposes an algorithm to maintain multi-hop clusters based on load-balancing clustering. This algorithm sets upper (U) and lower (L) bounds on the number of member nodes that a CH can manage. When the number of member nodes in a cluster is less than L, the merge mechanism is invoked. The cluster tries to find a neighboring cluster to satisfy the requirement that the sum of the number of member nodes in these two clusters be less than U. If this sum value for all neighboring clusters is larger than U, it tries to find a cluster to minimize the sum value. When two clusters merge, the CH with more member nodes becomes CH and the other CH becomes a member node. When the number of member nodes in a cluster exceeds U, the division mechanism is triggered. The cluster will be separated into two clusters having similar sizes. The difference in the loads among the clusters is therefore alleviated.

Chiasserini [77] proposes an algorithm to balance traffic loads of clusters and to regulate the transmission power of CHs to evenly drain energy from the CHs. This algorithm assumes a centralized network and a CH can dynamically adjust its cluster size through power control. In this algorithm, after the clusters are formed and the CHs are selected, the new values of residual energy of CHs are calculated and updated. If the difference in the residual energy of two successive updates exceeds the threshold, the network will perform a re-clustering process. A different number of nodes will be assigned to each CH according to their residual energy to better balance the loads among the clusters and CHs.

3.1.3 Reducing power consumption of cluster maintenance

To maintain the clusters in a clustered MANET, some extra messages to update the links between nodes are necessary. The frequently changed network topology results in frequent cluster topology updates, thereby drastically increasing the control messages for cluster maintenance. These behaviors may consume a large portion of the network bandwidth, drain significant energy of the node and override the scalability and performance of the network [11, 71]. It is therefore important to reduce the cost of maintaining the clusters for energy saving considerations. Some clustering algorithms reduce the frequency of rotating the role of CH by extending the service time of the CH [72] or by selecting the relatively stable nodes to be CHs [79-81]. Some reduce the flooding or control messages (control overhead) [19, 70, 82-84] caused by routing discovery or cluster maintenance. Some sense the mobility of the nodes so that the clusters can be adaptively maintained to reduce the unnecessary message exchanges [72, 85-87].

1. Extending the service time of the CH

Gavalas [72] proposes an algorithm which adds a handover penalty coefficient to a CH during the CH rotation to allow a node to serve as CH for a longer period. This algorithm rotates the CH role according to the CH competence (*CHC*) (3.2):

$$CHC = (c1 \times d + c2 \times b) - p \tag{3.2}$$

where $c1$ and $c2$ are weighted coefficients and $c1+c2=1$, d is the number of neighbors, b is the residual energy of the node and p is the handover penalty coefficient. The *CHC* value of a node is kept updated, and the node with the lowest CHC value in its local area is selected as CH when CH rotation occurs. However, to improve the stability of the cluster, different penalty coefficients are allocated to the nodes. The value of zero is assigned to a member node whereas a positive integer value is assigned to a CH. Therefore, the CHs have the advantage in the consequent CH competition. Furthermore, re-selection of the CH only happens when the residual energy of the CH is lower than the threshold or when the topology of the network is dramatically modified [72]. This algorithm extends the service time of a CH and thus reduces the message exchanges caused by CH rotation. The selected CHs, however, may drain energy quickly and die prematurely.

The algorithm in [79] selects the most stable node in its area as CH to improve the stability of the cluster, so that the CH can stay in the cluster for a longer time. This algorithm reduces the amount of re-clustering caused by the absence of the CH. In this algorithm, a willingness factor (W) is adopted as a metric to elect CHs. W describes the respective mobility of a node and the relative mobility of a group of nodes, which are jointly represented by a stability factor S and the number of relatively stable nodes M. W can be expressed as a function of S and M, $W=f(S, M)$. A larger value of W means a more relatively stable node. The node with the largest W in its local area is selected as CH. The value of W is kept updated. The selection of a new CH will be invoked only when its W value is lower than the threshold or two CHs move into the reachable range. The node with the highest W value will be the new CH in its local area. This algorithm reduces the frequency of re-clustering nodes, thus improving energy efficiency.

The algorithm by Basu [80] also selects the nodes with relatively low speeds to be the CHs during cluster formation when the network starts to work. The selection of CHs is a local activity so that a CH is determined by its neighbors and itself. The local speed of a mobile node is determined by calculating the variance of the speed of the node relative to each of its neighbors. A node with a higher variance value indicates it is a relatively more mobile node than its neighbors. Consequently, a node with lowest variance value in its neighborhoods becomes CH. The cluster maintenance phase is invoked when two CHs move into the reachable range or any node cannot access any CH. To reduce unnecessary re-clustering caused by the CH selection, such as when two CHs incidentally pass by each other during a short period, a threshold of time period is set. If the reachable time of two CHs is greater than the threshold, one CH will give up its role and become a member node. Otherwise, both will keep their roles. This algorithm also improves the stability of the clusters, therefore reducing the cost of maintaining the clusters.

2. Reducing flooding or control messages (control overhead)

Flooding in ad hoc networks is usually used to find a feasible route to a destination or to advertise routing information [82]. Clustering control messages are used to exchange the information of each link during the cluster maintenance and to build a virtual backbone or infrastructure [11]. Energy can therefore be saved by reducing flooding or control messages [19, 70, 82-84].

The adaptive clustering algorithm proposed by Lin [70] considers reducing the control messages during cluster organization. In this algorithm, every node maintains its own ID and the IDs of its neighbors within one hop. Each node that declares itself as CH will set its own ID as its cluster ID (CID). Initially, a mobile node with the lowest ID in its local area becomes CH. The CID information includes a node's ID and the CID. If the node's ID is the same as its CID, then the CID information is a CH claim. Otherwise, it is just a claim of a member node. If a node gets CID information from a neighbor, it compares it with its own CID. If its own CID is unspecified or larger than the ID that it receives, it will join this node as a member node. This process continues till all mobile nodes access any cluster. This mechanism promises small control messages for cluster formation because each mobile node broadcasts only one CID message [11].

Passive clustering [82, 83] needs no dedicated protocol-specific control messages to form the clusters [11]. The nodes in this algorithm have four states: *initial*, *CH*, *gateway node*, and *member node*. When the network starts to work, all the mobile nodes are set in initial state. Only a node with initial state has the chance to be a potential CH. When a potential CH with initial state is ready to send out a message, it will include the claim as CH in its messages. When its neighboring nodes receive this message, two scenarios will happen: if a neighboring node only receives one claim from the CH, it will become a member node and join this CH; if a neighboring node receives more than one claim from the CHs, it will then become a gateway node. Because this algorithm does not broadcast any control message to maintain the clusters, each node needs to update its own cluster status by keeping a timer. When a member node does not receive any packet from its CH for a given period, its status reverts to initial [11, 82]. Besides, in passive clustering, only the CHs and gateway nodes forward the flooding packets, which dramatically reduces the number of flooding packets. However, this clustering algorithm may generate many gateway nodes. To reduce the unnecessary gateway nodes, it allows a node to become a gateway node only when it observes a criterion of $\alpha \times NC + \beta > NG$, where α is the coefficient which is properly chosen based on the desired degree of gateway redundancy, that means if more gateway nodes are expected, α will be decreased; β is the factor of gateway-node redundancy; NC is the number of CHs, and NG is the number of the gateway nodes. Passive clustering performs cluster formation and cluster maintenance when the mobile nodes have data to send. It therefore completely eliminates the control messages (control overhead) for clustering [11, 82, 83].

3. Sensing mobility of nodes

Mobility is a prominent characteristic of MANETs, and is also a main factor that causes topology change and route invalidation. It is therefore important to take the mobility into account in cluster construction to form a stable cluster [11]. This stable cluster reduces the cost of cluster maintenance. Mobility-sensitive clustering indicates that the cluster architecture is determined by the mobility behaviors of mobile nodes. Therefore, the re-affiliation and re-clustering rate can be naturally decreased [11].

Gavalas [72] proposes a clustering algorithm to broadcast the control messages adaptively. When the node speed is low, the broadcasting period is prolonged. Otherwise, it is shortened. The unnecessary broadcasting when the node speed is low is then decreased, thereby resulting in higher energy efficiency.

McDonald [85] proposes a distributed (α, t) cluster framework to adjust the cluster sizes adaptively according to the node speed. The (α, t) criteria is that, every node in a cluster has a path to every other node, that will be available over a specified interval of time t with a probability larger than α, regardless of the hop distance between them. In the (α, t) cluster framework, when a source node enters the network it should find a cluster to join. The source node can join the cluster if the destination is reachable via an (α, t) path. According to the speeds of the nodes, the size of the clusters in the (α, t) cluster framework can be adaptively adjusted. If the speeds are low, (α, t) clustering provides an infrastructure that is more proactive and clusters with large size are needed. However, on-demand routing will dominate when the speeds of the nodes become high. Cluster sizes will then be diminished. Unfortunately, if the node speed changes frequently between low and high, this framework may be busy adjusting the cluster sizes, thereby making it inefficient [11].

The algorithm by An [86] groups the nodes with similar mobility characteristics into clusters, as shown in Fig. 3-1, thereby making cluster maintenance easier. A node named n in this algorithm will periodically disseminate its own velocity information to its neighboring nodes. Upon reception of this velocity information, a neighboring node called *m* calculates its relative velocity between itself and node *n* respectively. Defining *Sm* as the set that includes node *m* and all the nodes from which node *m* receives mobility information, the node that has a value lower than the threshold and has the lowest ID is selected as a tentative CH. If two tentative CHs move into the area that can reach each other, these two clusters will be merged. However, this mergence is subject to the limitation of the maximum size. When a node *p* in cluster *Ci* moves into a cluster *Cj*, if the CH of cluster *Cj* has a lower mobility value than the threshold, node *p* will request to join cluster *Cj*. The procedure of cluster organization improves the stability of the cluster, thereby decreasing the power consumption of cluster maintenance.

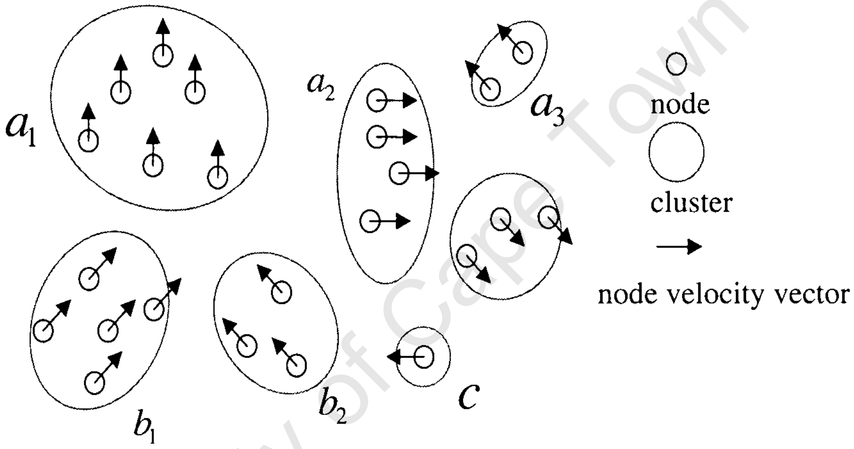


Figure 3-1. Mobility-based clustering protocol (Adopted from [86]).

Sivavakeesar [81] proposes an algorithm to manage the mobility of the nodes through a prediction-based clustering approach. In this approach, each mobile node can anticipate the availability of its neighbors through a mobility prediction algorithm called (*p*, *t*, *d*)-model, where *p* is the probability that one node, which has the distance *d* from the center of the cluster it belongs to, stays within the cluster for some specific time period *t*. In the (*p*, *t*, *d*)-model, each node is supposed to have complete location information of the center of the virtual cluster, which is a circular area. If greater accuracy of mobility prediction is required, each virtual cluster can be further split into a number of equal sub-clusters. A node generates the strings of virtual-cluster identifiers and maintains its mobility database at a specific time to predict future movement. Each CH knows the predicted mobility patterns of its member nodes. The mobility management functions will then be performed in a proactive manner with low cost. This algorithm improves the stability of the cluster, thereby also reducing power consumption of cluster maintenance.

3.1.4 Reducing power consumption of intra-cluster communication

Similar to WSNs, the CH selection in MANETs is also related to the total power consumption of the cluster. Locating the CH in the central area of the cluster reduces the power consumption of intra-cluster communication. However, the realization has more challenges in a MANET than that in a WSN because of the movement of node.

The algorithm in [88] aims at minimizing the power consumption of data communication summed by all CH-member node pairs. Single-phase clustering and double-phase clustering are proposed.

In the single-phase clustering, initially every CH pages at the same maximum power level and the member node will join the CH from which the highest power was received, which is assumed to be the nearest CH. It therefore reduces the power consumption of data communication between the CH and the member node. Since a CH can only serve a limited number of member nodes, it first allocates channels for member nodes that only receive a single signal from itself. If there are still some channels available after allocation, they will be allocated to member nodes that receive more than one signal in the order of the power level of the signal received from the CH. The member nodes that receive more than one signal try to communicate with the nearest CH. The communication cost of each pair of CH and member node is possibly minimized, thereby improving energy efficiency.

To further lower the call-drop rate, double-phase clustering adds a channel re-allocation phase on the basis of single-phase clustering. Each CH, with free channels after its first-round allocation, re-allocates these channels to the member nodes which did not receive a channel in the first round in its range. The channel allocation procedure also follows the strength of the received signal. Though this algorithm is reported to save energy for MANETs, the allocating process may consume significant energy for both CH and member nodes. Furthermore, no detailed method to select CHs and to maintain the clusters is provided [11].

3.2 Analysis of existing energy-efficient clustering algorithms for MANETs

Rotating the CH role in MANETs distributes the higher burden of the CH among the nodes. The way to rotate the CH role in clustered MANETs, however, becomes a challenge, because this rotation exacerbates the stability of the clusters, which are already changed frequently by the movement of the nodes. Therefore, in a MANET, the tradeoff between the cluster stability and the balance of power consumption needs to be achieved. Many clustering algorithms though improve the stability of the cluster and thus the network by prolonging the service time of CH. This means that the CHs in the network will drain energy quickly and die prematurely. Therefore, besides the need to improve the stability of the cluster, the distribution of the higher burden of CH also needs to be carefully considered.

Balancing power consumption among the clusters or CHs prevents premature node death in MANETs. Limiting the maximum size of the cluster may still make the clusters experience significant difference in sizes. Limiting both the maximum and the minimum sizes of the clusters better balances the burdens among the clusters. However, maintaining the cluster sizes within a strictly limited range is quite expensive in a highly mobile MANET. If the cluster sizes can be adaptively balanced according to the node speed, energy efficiency of maintaining the clusters and cluster sizes can be optimized.

In a MANET, significant energy is consumed in maintaining the clusters. Extending the service time of a CH reduces the message exchanges for re-selecting the CHs. However, the node that serves as CH drains energy rapidly and will die prematurely. Selecting a relatively stable node as CH improves the stability of a cluster. However, the capability of estimating the speed of node requires more powerful nodes and this algorithm is also not suitable for the networks with frequent speed 'burst'. Forming and maintaining the clusters only when the nodes have data to send dramatically reduces the control messages for cluster organization and maintenance. However, for a network with 'bursty' traffic, the cluster structure is difficult to maintain and cannot be guaranteed to be able to serve upper-layer routing or data forwarding [11]. Adaptively adjusting cluster sizes to make the network more proactive or reactive according to the speeds improves the efficiency of the routing protocol. However it may not be suitable for the network in which the node speed changes frequently, because these frequent changes in speed will make the network busy with re-clustering and alternating routing protocols.

Locating the CH in the central area of cluster reduces power consumption of intra-cluster communication. This algorithm proposes an effective method to determine the distance between a member node and a CH by evaluating the strength of the signal that this member node receives from the CH.

These existing clustering algorithms have prolonged the lifespan of MANETs. The energy efficiency of the network can be further improved if the unnecessary broadcasting to maintain the clusters is reduced and the burdens of the clusters are balanced adaptively.

3.2.1 Unnecessary broadcasting in a clustered MANET

In a typical clustering algorithm, intra-cluster communication usually uses a proactive routing protocol whereas inter-cluster communication usually applies a reactive protocol. In a single-hop cluster, in which any member node sends data to the CH by only one hop, a member node may need higher transmission power to access the CH as the nodes are usually distributed sparsely in a MANET [11], thereby reducing the energy efficiency. A multi-hop cluster is a more energy efficient organization, especially in a MANET with large size, as it usually requires large sized clusters. Some multi-hop clustering algorithms, such as *K*-clustering algorithms [89-95], have been proposed. A K-clustering algorithm defines the cluster size in such a way that any node within the cluster can access any others within K hops [89].

Each member node in a multi-hop cluster maintains a routing table for intra-cluster communication. Movement of the node may change cluster and routing information, thereby making maintaining cluster and intra-cluster routing information necessary. A typical method used in most clustering algorithms is to broadcast control messages periodically to update the link information. However, in a realistic MANET, the node speed may change from time to time. Typically, when the speed is higher, the clusters will be changed more frequently. Therefore, a clustered MANET with higher node speed needs more frequent broadcasting. To maintain the clusters efficiently in a MANET, the control messages need to be broadcast to meet the requirement of the potentially maximum node speed. When the node speed is low, the unnecessary broadcasting wastes energy.

Adaptively broadcasting the control messages in accordance with the node speed can reduce the unnecessary broadcasting and will result in higher energy efficiency. The following provides an example to show the difference in power consumption of adaptive and periodic broadcasting.

It is assumed that a cluster has 15 nodes, including the CH, as shown in Fig. 3-2. The node at the centre of the cluster is the CH. The size of the control message is 25 bytes. The required transmission range is assigned to each member node according to its distance to the CH. It is assumed that the frequency of the adaptive broadcasting is proportional to the node speed and the control messages are broadcast every 100ms when the nodes are moving at 20m/s. Ten levels of speed are set from 2m/s to 20m/s. Each level increases by 2m/s over the former one. The nodes move at the speed of each level for 10s. The parameters of power consumption are the same as those in [13, 14, 34].

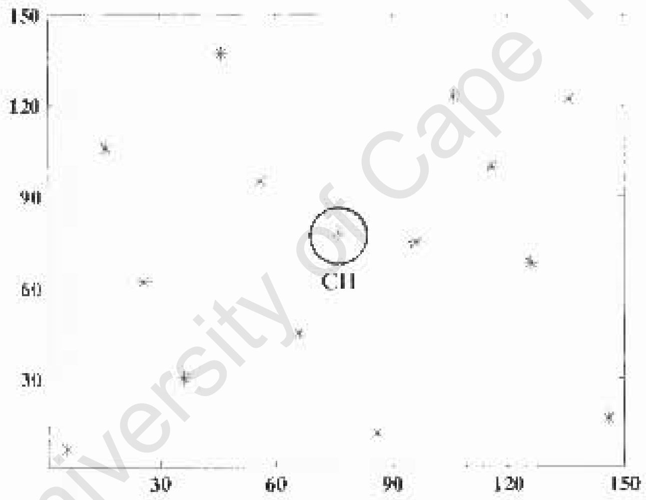


Figure 3-2. Cluster to measure power consumption of periodic and adaptive broadcasting.

The total consumed energy by adaptive and periodic broadcasting within 100s is shown in Fig. 3-3. The adaptive broadcasting saves by as much as 50% energy over the periodic broadcasting.

Related work

Gavalas [72] has proposed an algorithm to broadcast control messages adaptively. However, this algorithm only broadcasts the control messages arbitrarily and no evidence can be found that the adaptive broadcasting actually meets the requirement of maintaining the cluster at each respective node speed. Secondly, the selected CHs in this algorithm will die prematurely as they need to serve as long as possible. Thirdly, no evidence is found in the simulation results to indicate that this algorithm improves energy efficiency. Therefore, a method which assures that the broadcasting actually meets the requirement of each respective speed and improves energy efficiency of the network is necessary.

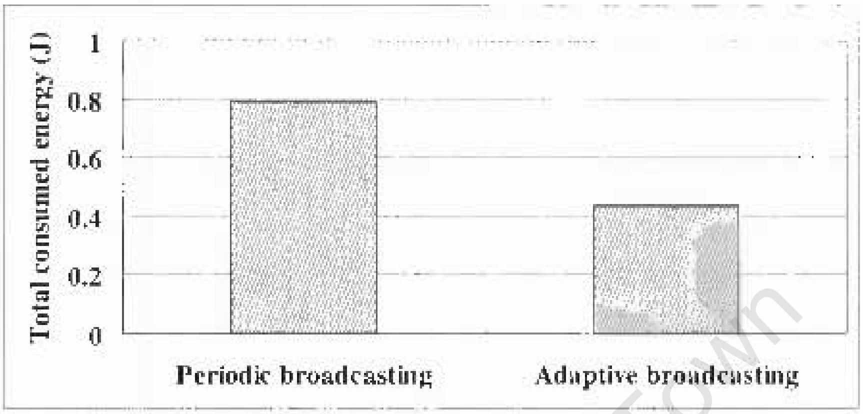


Figure 3-3. Total power consumption of adaptive and periodic broadcasting.

3.2.2 Different burdens among clusters

Though balancing the cluster sizes in a MANET is far more difficult than in a WSN, equalizing the cluster sizes balances the power consumption among the nodes/clusters, thereby preventing premature node death. The following provides an example for comparison of the total power consumption of two clusters with different sizes.

It is assumed that there are two clusters with different sizes, as shown in Fig. 3-4. The larger sized one has 20 nodes and the smaller sized one has 5 nodes, both including the CH. To make them comparable, it is assumed that both CHs are at the centre of the cluster and each member node has one timeslot to send data to the CH during each CH period. The packet size that the member nodes send to the CH is 500 bytes. Three items are compared: (i) the total energy consumed by intra-cluster communication (including the energy consumed by the member nodes to send data to the CH and the energy consumed by the CH to receive the data from the member nodes); (ii) the total energy consumed by the CH (including the energy consumed by receiving data from the member nodes and the energy consumed by relaying the received data to the next hop) and (iii) the total energy consumed by the cluster (including the energy consumed by the member nodes to send data to the CH and the energy consumed by the CH to receive these data and to relay them to the next hop).

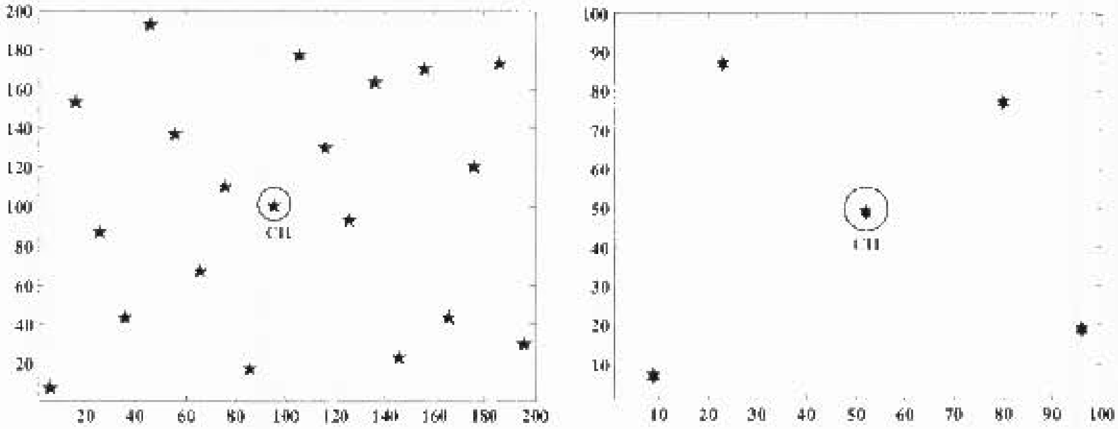


Figure 3-4. Clusters with different sizes for comparison of power consumption.

The consumed energy of different items is shown in Fig. 3-5. In a homogeneous MANET with identical nodes, the energy stored in the larger sized cluster is four times that of the smaller sized one. All components of energy consumed in the larger sized cluster are more than four times that of the smaller sized one. If these two clusters are in the same network, the nodes in the larger sized cluster will die earlier than the nodes in the smaller sized cluster. Equalizing cluster sizes will balance the burdens of the clusters, thereby preventing premature node death.

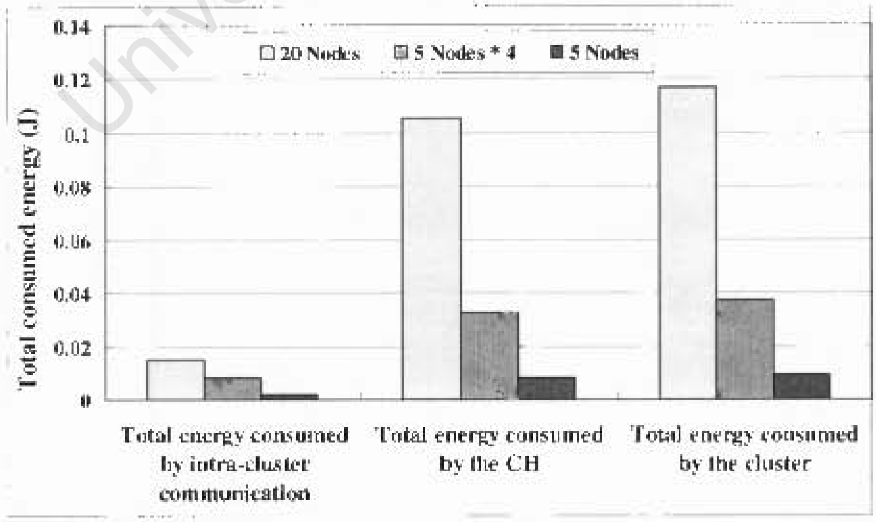


Figure 3-5. Total power consumptions of the clusters with different sizes.

It can be seen that the nodes in a smaller sized cluster have lower burden than those in a larger sized one. This, however, is not to say that a smaller sized cluster organization is always more energy efficient than a larger sized cluster organization. When the cluster sizes are smaller, more CHs will be selected and more clusters will be formed. These additional CHs will complicate the network backbone, thereby increasing the flooding in inter-cluster communication and thus the power consumption. Therefore, the cluster sizes also need to be subject to the balance between the power consumption and the backbone complexity.

Related work

Some algorithm has considered alleviating the difference in cluster sizes by limiting the maximum cluster size, such as in [72]. Some has considered limiting both maximum size and minimum sizes, such as in [73-76]. Some has considered balancing the burdens among the CHs by taking the residual energy of the CH into account, such as in [77, 78]. All algorithms, however, fail to optimize the energy efficiency as previously analyzed at the beginning of Section 3.2. On the other hand, adaptively maintaining cluster sizes further improves energy efficiency of the network. When the node speed is low, the clusters will be limited and maintained within a more strict range. Otherwise, more variance in cluster sizes is allowed, thereby releasing the network from frequent re-clustering when the nodes are moving quickly.

3.3 Summary

Existing energy-efficient clustering algorithms for MANETs have been investigated in this chapter. The algorithms were classified into: rotating the role of CH, balancing power consumption among CHs or clusters, reducing power consumption of cluster maintenance and reducing power consumption of intra-cluster communication. The general features of each category were summarized and some representative algorithms were described in detail. The strengths and weaknesses were analyzed. The issues of unnecessary broadcasting of control messages and unbalanced power consumption among clusters were then addressed to further improve the energy efficiency of MANETs. These two issues will be taken into account in the proposed clustering algorithm in Chapter 6.

4 Balancing and reducing power consumption (BRPC) in homogeneous clustered WSNs

4.1 Strategies to improve energy efficiency in clustered WSNs

Chapter 2 analyzed existing energy-efficient clustering algorithms for WSNs and concluded that a new clustering algorithm is necessary. Such an algorithm needs to balance power consumption throughout the network and maintain each CH in the central area of the cluster by re-clustering without the resultant ripple effect.

Typically, a clustering algorithm can be distributed or centralized. In a distributed clustered WSN, the nodes self-configure themselves into clusters without the global information of the network. In a centralized clustering algorithm, the data sink is required to participate in creating the clusters and managing the clustered network. Most clustering algorithms set the expected number of clusters and cluster sizes before the WSN starts to work. In a distributed clustering algorithm, however, the pre-set number of clusters and the cluster sizes are not guaranteed, because each node makes autonomous decisions and lacks the global information, thereby making cluster organization with optimal energy efficiency impractical [14]. On the other hand, the sink in the network usually has sufficient resources to administer the network. Requiring the sink to participate in managing the WSN helps to efficiently organize the nodes into clusters with expected cluster numbers and cluster sizes [14, 34, 96, 97], which is important to balance the power consumption throughout the network. This chapter therefore proposes a distributed clustering algorithm yet centralized to start the cluster organization when the network starts to work to better organize clusters.

Many clustering algorithms divide the network lifetime into rounds [14, 34, 37, 40, 42]. Each round begins with a set-up phase when the clusters are organized, followed by a steady state phase where several frames of data are transferred from the nodes to the CH and on to the data sink [14]. During the cluster organization, the clusters will be formed, the CHs will be selected and a TDMA scheduling scheme will be built for each CH. To form the clusters, the nodes need to exchange the information. The typical total number of clustering-related messages exchanged during the cluster formation is $O(nV)$, in which V

is the total number of nodes in the network and *n* is a constant coefficient which is different from algorithm to algorithm [11]. To reduce the total power consumption of intra-cluster communication between member nodes and CH and also to minimize inter-cluster interference, most existing algorithms re-cluster the nodes during each round with resultant ripple effect. This frequent re-clustering with resultant ripple effect may consume more energy than the improvement it achieves by optimizing the CH selection at each round [14]. On the other hand, a fixed cluster only needs to exchange the information within the cluster when re-clustering occurs. Therefore, in such a clustered network, there is no resultant ripple effect when re-clustering occurs, thereby resulting in higher energy efficiency. Furthermore, this fixed cluster makes it feasible to equalize cluster lifetime by making the cluster size proportional to the power consumption of the cluster. However, the fixed nature of the clustering algorithm may limit its realistic application [14]. To make the fixed clustering algorithm more relevant, when new nodes join or original nodes leave the network (which can be detected when information exchange occurs during each round), re-clustering with resultant ripple effect will take place.

The proposed algorithm serves to balance power consumption throughout the network and to improve energy efficiency. This algorithm therefore can lead to expectations of network lifetime. Typically there are three types of lifespan that have been achieved for a clustered WSN, as shown in Fig. 4-1 (a, b and c).

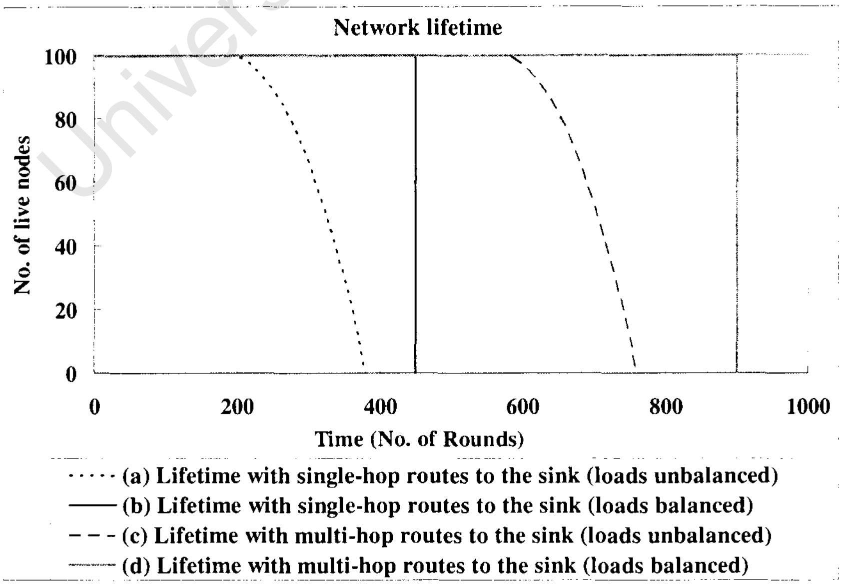


Figure 4-1. Lifetimes of different clustering algorithms.

In Fig. 4-1, curves (a) and (b) show the lifetime of the network in which the CHs send data to the sink over single-hop routes. The clusters in a network with lifetime of (a) are designed with similar sizes, such as LEACH [13, 14], whereas with lifetime of (b) are designed in such a way that their power consumptions are balanced so that all nodes have similar lifetime, such as EECS [40]. However, sending data to the sink over a single-hop route is not energy efficient and requiring all nodes to access the sink over a single-hop route is infeasible, especially in a network with large size. Curves (c) and (d) show the lifetime of the network in which the CHs send data to the sink over multi-hop routes. Curve (c) shows the lifetime of the network in which the clusters are designed with similar sizes, such as EECP [36]. This chapter proposes an algorithm to achieve the network lifespan as shown in curve (d), which further balances power consumption throughout the network and improves energy efficiency of the network.

4.2 Proposed clustering algorithm-BRPC

This subsection presents the proposed clustering algorithm in detail. It first introduces the network and energy models and then describes the cluster organization and data communication phases. Following this is the analysis of the methods to equalize cluster lifetime and maintain the CH in the central area of the cluster by re-clustering without the ripple effect.

4.2.1 Network model and energy model

A data sink in a WSN has sufficient resources, such as memory, processing capability and energy. It therefore is capable of performing cluster organization [34]. As energy efficiency is a major challenge in WSNs, the data sink is located near the network so that the power consumption of sending data to the sink is reduced.

The energy-limited identical sensor nodes, which have the capacity to vary transmission power, are uniformly distributed in the network. Some existing clustering algorithms [14, 34, 40, 42] assume that all nodes can reach the sink over a single-hop route, which can be witnessed from their simulation results, This, however, is not the actual situation in realistic application, especially in a large sized WSN. BRPC, therefore, assumes that the maximum transmission range of the nodes is limited and only some nodes, which are nearer the sink, can reach the sink over a single-hop route. However, to achieve higher energy efficiency, these nodes can use a multi-hop route to send data to the sink if this multi-hop route is more energy efficient than the single-hop one.

The sensor nodes are aware of their two-dimensional location information. With this information, the nodes can be geographically grouped into clusters with the expected sizes and shapes. The location information can be achieved by global positioning system (GPS) technology. However, if GPS technology is not supported and the relative node positions can be determined before the nodes are deployed into the network; virtual location information of the nodes is also applicable. This virtual location information of the nodes can be acquired by setting the data sink as the origin (0, 0) of the (x, y) coordinate system. The virtual/GPS location information does not need to be updated, as the nodes are stable after they are deployed in the network. The virtual location coordinates can also be acquired without the pre-set topology information before the nodes are deployed into the network, the relative detailed methods can be found in [98, 99].

A sensor node can be a member node or a CH. A member node senses data from the environment and then transmits the data to its CH. Besides generating data itself, a CH needs to collect all data from the member nodes, and then aggregate these collected data and relay them to the sink.

Typically, a sensor node has three components of power consumption: data generation, data processing and data communication [14, 34]. Considering that the main purpose of this research is to design an energy efficient clustering algorithm, only the power consumption relating to data aggregation and data communication is considered. The energy model is similar to the models in [14, 34, 36, 37, 40, 42], which is described in (4.1). The meaning of each parameter in (4.1) can be found at Table 2 in Section 2.1.

$$
\begin{aligned}
P &= PT + PR = \left(e_t + e_{amp} d^n\right) \times a + e_r \times a \\
&= \begin{cases} e_t \times a + e_{amp} d^2 \times a + e_r \times a & (d < d_{crossover}) \\ e_t \times a + e_{amp} d^4 \times a + e_r \times a & (d \geq d_{crossover}) \end{cases}
\end{aligned}
\tag{4.1}
$$

The power attenuation n depends on the distance between the transmitter and receiver. For short distances, the propagation loss is modeled as inversely proportional to d^2, while for long distances this loss is modeled as inversely proportional to d^4 [14, 36]. In BRPC, the similar parameters applied in [14, 34, 36, 37, 40, 42] are used: for distance $d < d_{crossover}$

(where $d_{crossover}$=87m), e_{amp} = 10pJ/bit/m^2; for distance $d \geq d_{crossover}$, e_{amp} = 0.0013pJ/bit/m^4; and e_t=e_r=50 nJ/bit. More detailed information about the parameters like transmitter height, receiver height and carrier frequency can be referred to [14].

4.2.2 Cluster organization phase

The cluster organization phase needs to form clusters, select CHs and determine a TDMA scheduling scheme for each CH.

Typically, there are two types of clusters: single-hop clusters and multi-hop clusters. A single-hop cluster means that any node in the cluster reaches the CH by a single-hop route whereas a multi-hop cluster means that some nodes in the cluster need to use a multi-hop route to reach the CH. For energy saving consideration, BRPC needs to select a more energy efficient cluster organization.

1. Single hop clusters or multi-hop clusters

The results of the examples in Section 2.2.1 show that a multi-hop route is more energy efficient than a long single-hop route in a linear network system in which the distance between any neighboring nodes is 60m. However, a cluster in a WSN does not necessarily need to be so big. This subsection therefore discusses a method to select a more energy efficient cluster organization by using the model shown in Fig. 4-2.

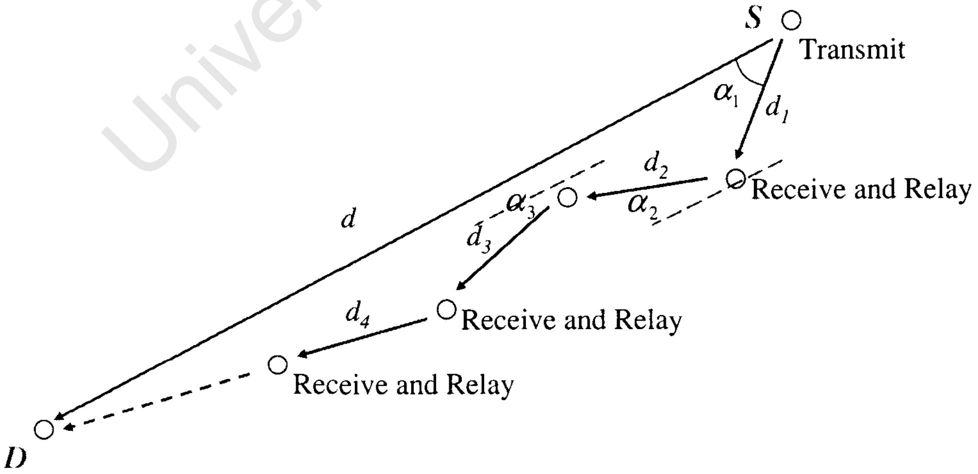


Figure 4-2. A multi-hop route vs. a single hop route.

A source node (S), which is supposed to be a member node, sends an a-bit data message to a destination node (D), which is supposed to be the CH of S, either by a single-hop or by a multi-hop route, as shown in Fig. 4-2. The number of hops in this multi-hop route is referred to as m. The distance of the single hop is d, the distance of the i^{th} hop in the multi-hop route is d_i. The angle between the i^{th} hop in the multi-hop route and the single-hop route is α_i. Referring to the energy model in (4.1), the total energy dissipated in transmitting and receiving data of the single-hop route is shown in (4.2), and that of the multi-hop route in (4.3). The relationship of d, d_i, and α_i is described in (4.4).

$$E_{Single} = a \times \left(e_t + e_{amp} d^n\right) + a \times e_r \tag{4.2}$$

$$E_{Multi} = m \times a \times e_r + \sum_m \left(a \times \left(e_t + e_{amp} \times d_i^{\,n}\right)\right) \tag{4.3}$$

$$\begin{aligned} & d_1 * \cos\alpha_1 + d_2 * \cos\alpha_2 + d_3 * \cos\alpha_3 + d_4 * \cos\alpha_4 + \cdots + d_m * \cos\alpha_m \\ & = \sum_m d_i * \cos\alpha_i = d \end{aligned} \tag{4.4}$$

To simplify the analysis, it is assumed that the distance of each hop d_i is the same and the average α_i is 30° (This value, however, is not a constant during the realistic application or simulation as the route that a node uses to relay data will be changed from time to time). S sends a packet of 500 bytes to D. The total energy consumed by routing data from S to D using different number of hops is shown in Fig. 4-3.

The bold curve in Fig. 4-3 shows the minimum-energy route to relay the packet from S to D. When d is smaller than 100m, a single-hop route is more energy efficient than any multi-hop route. When d increases, the multi-hop routes become more energy efficient. For example, when d is between 180m and 280m, a route with 3 hops is the most energy efficient one to relay the data.

The analysis in some existing clustering algorithms shows that the energy efficiency of the network can be optimized when 4% or 5% of the nodes become CHs [14, 40, 42, 43]. In a typical clustered WSN, the longest distance from a member node to the CH is usually limited within 100m [7, 14], which means that a single-hop route is more energy efficient than a multi-hop route (refer to Fig. 4-3). BRPC therefore determines to select a single-hop cluster organization.

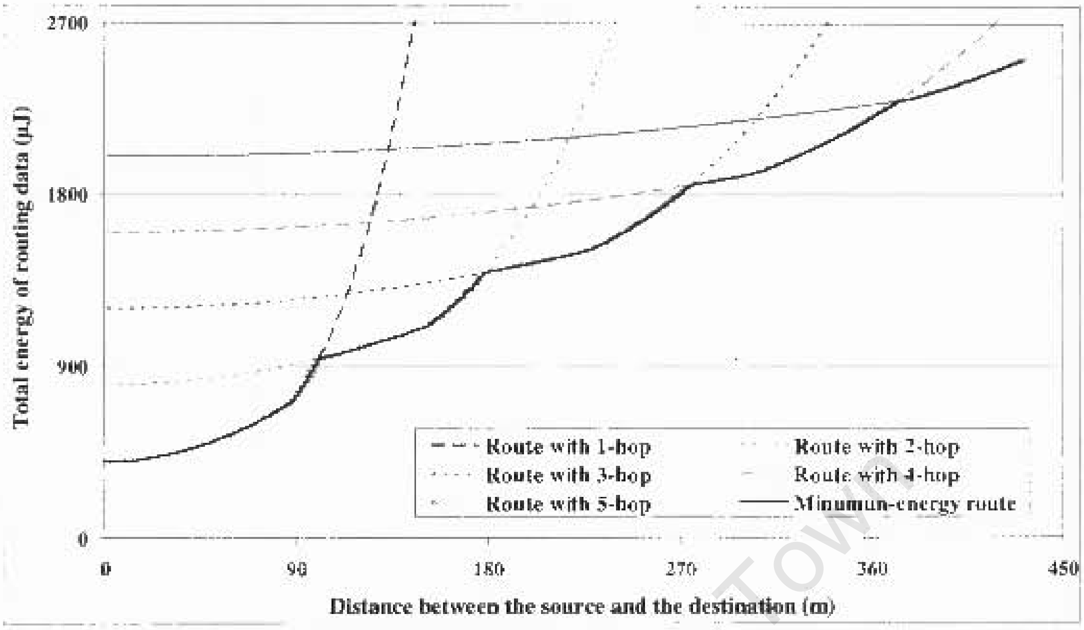


Figure 4-3. Total energy of routing data by different number of hops.

Below is an example which compares the total energy dissipated in intra-cluster communication in a single-hop cluster to a multi-hop one. The cluster shown in Fig. 4-4 has 20 nodes, which are distributed within an area of 60m×60m.

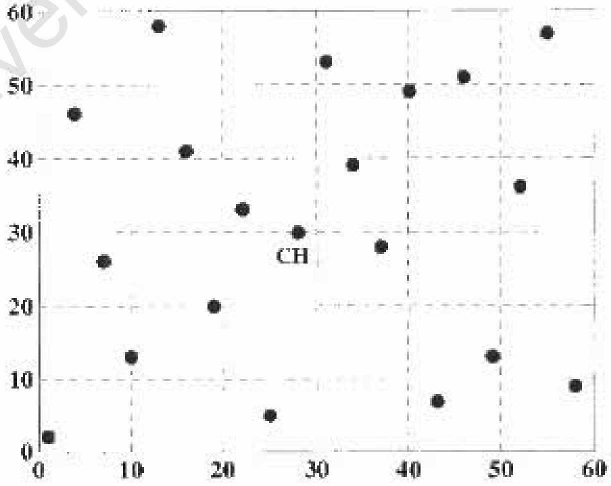


Figure 4-4. A cluster used to illustrate the total energy dissipated in a single-hop cluster and in a multi-hop cluster.

The node at the centre of the cluster is the CH for both single-hop and multi-hop scenarios. The CH collects the data packets, each of which is 500 bytes, from all member nodes. In a multi-hop scenario, it is assumed that when the distance between a member node and the CH is greater than 30m, a multi-hop route will be used. The total energy dissipated in intra-cluster communication is 9.69mJ in the multi-hop cluster whereas it is 8.10mJ in the single-hop cluster. Therefore, in such a cluster, a single-hop cluster organization is more energy efficient than a multi-hop one.

2. Clustering the nodes

The whole process of cluster organization is shown in Fig. 4-5. When the network starts to work, each sensor node needs to send its GPS/virtual location information to the sink. These information packets are quite small and the location information does not need to be updated. Therefore the energy consumed by sending these small packets to the sink is negligible and each node can send it to the sink over any multi-hop route.

On reception of the location information, the sink divides the network into geographically-based clusters, with the pre-set number of clusters and cluster sizes. The pre-set cluster sizes serve to balance power consumption throughout the network, which will be analyzed in detail in Section 4.2.4. The area that defines the cluster is based on the coordinates of the nodes: *Cluster_i* (min(x_i), min(y_i), max(x_i), max(y_i)), where i is the ID of the cluster, min(x_i) and min(y_i) are the minimum x and y coordinates of the cluster respectively, and max(x_i) and max(y_i) are the maximum x and y coordinates of the cluster respectively. After determining the cluster partitions, the sink broadcasts the partition information to the network. This partition information includes the cluster ID, location information of the cluster and the number of nodes in each cluster. If the sink has sufficient power to reach every node in the network, this information can be sent to the nodes by a single-hop route. If the network size is large and the sink has not enough power to reach every node by a single-hop route, this partition information will be relayed to each node by a multi-hop route. To avoid any loop, only the nodes that have a longer distance to the sink can relay the information from its neighboring nodes. The node's distance to the sink can be easily acquired from the location information.

On reception of the partition information, each node needs to compare its own location information with the one it receives from the sink to determine which cluster it belongs to.

For example, a cluster is defined as *Cluster_8* (50, 50, 100, 100), a node with the location of (75, 75) can determine that it belongs to *Cluster_8* and its cluster ID is 8. When each node knows its cluster ID, the partition phase is finished. Now the nodes in the same cluster need to reorganize each other.

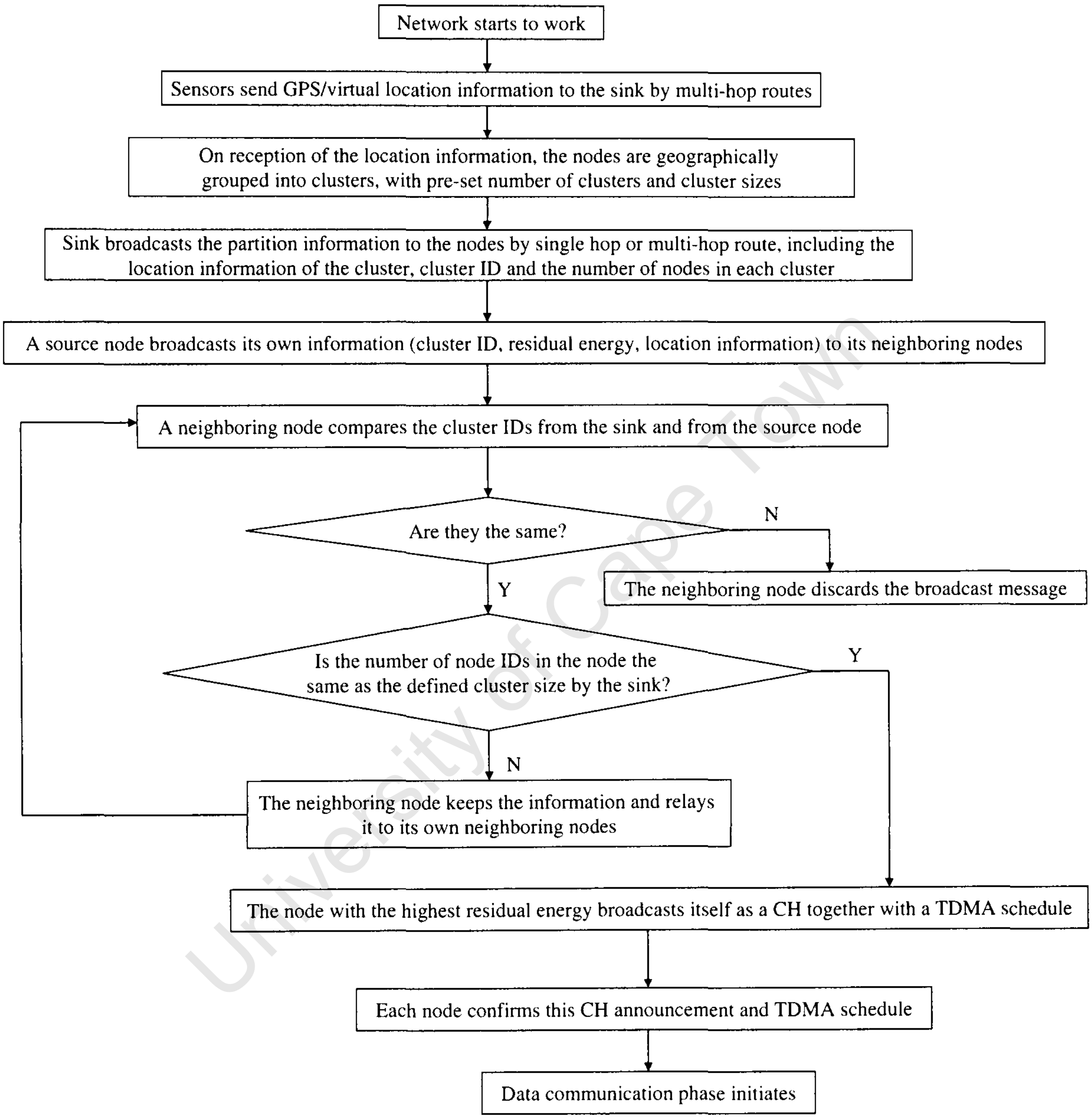


Figure 4-5. Flow chart of cluster organization phase of BRPC.

A node named *S* broadcasts a message, including its cluster ID, location information and residual energy, to its neighboring nodes. On reception of this message, a neighboring node *D* compares its own cluster ID to the one from *S*. If these two cluster IDs are different, *D* discards the message and so determines that it is not a cluster-mate of *S*. If they are the same, a relationship between *S* and *D* is established as cluster-mates. *D* then picks up this message and stores it. *D* continues to relay this message to its own neighboring nodes. This process will be repeated until all nodes receive the information of the others in the same cluster. The process is complete and no longer needs to be repeated when the number of the nodes in the cluster is the same as the one from the partition information that the sink sends to each node. A CH can be selected once all nodes in a cluster recognize each other.

3. Selecting CHs

Many existing clustering algorithms select those nodes which have more residual energy as CHs [14, 34, 40, 42]. They then require other nodes to join the CHs as member nodes in order to form the clusters. These algorithms, however, cannot guarantee the balance of the loads among the clusters as they cannot guarantee the cluster size, which is important to equalize cluster lifetime. BRPC, therefore, organizes the clusters before the CH selection. This ensures that the energy stored in the cluster can be proportional to the power consumption of the cluster.

Similar to most existing clustering algorithms, BRPC selects CHs and rotates their roles according to the residual node energy. However, to improve energy efficiency, the CHs are located in the central area of the cluster as long as possible. The cluster lifetime is separated into two periods: the first period (*period1*) only selects the node with the maximum residual energy in the central area of the cluster as CH. The nodes in the central area will run out of energy rapidly. When the residual energy of any node in the central area reaches a threshold, which is adaptive and can be pre-set, *period1* expires and the second period (*period2*) begins. The node with the highest residual energy in the whole cluster will be selected as CH. As the CH role is rotated frequently in *period1*, the higher burden of the CH is well distributed among the nodes. This means that, when the residual energy of any node in the central area drops below the threshold, the residual energy of all the other nodes should be at roughly the same value, i.e. just above the threshold. The details of how to optimize the service time of CHs in the central area of the cluster and the relationship between *period1* and *period2* are to be analyzed in detail in Section 4.2.4.

The selection of CHs in the first round or the rotation of CH thereafter in BRPC is quite simple. During the cluster formation when the network starts to work or when each round initiates thereafter, all nodes broadcast the cluster ID, node ID and residual energy information to the nodes in the same cluster to exchange the information. All information received will be tabled and updated in each node. Should all nodes have identical energy (i.e. when the network starts to work) in the cluster, the node nearest the center of the cluster becomes CH. The node with the highest residual energy in the central area during *period1* or in the entire cluster area during *period2* becomes CH for the next round.

After the clusters have been formed and CHs have been selected, each CH needs to set up a TDMA schedule and transmit this schedule, which also includes the cluster ID and node ID, to the member nodes. This TDMA schedule allows each node to send data to the CH during its own timeslot. Besides, it also ensures that there are no collisions among data messages and allows the radio components of each member node to be turned off at all times except during their transmission time. This minimizes the energy dissipation of the individual sensors [14]. The number of timeslots is determined by the number of the nodes in the clusters. The information of the number of nodes in each cluster is obtained during the cluster formation phase. After the TDMA schedule is known by all nodes in the cluster, the cluster organization phase is complete and the data communication phase can start.

4.2.3 Data communication phase

The flow chart of the data communication phase is shown in Fig. 4-6. Similar to [14, 34, 40, 42], the lifetime of the network in BRPC is divided into rounds. Each round initiates with a set-up phase in which clustering/re-clustering occurs. This phase is followed by a data communication phase which is separated into frames, as shown in Fig. 4-7. Each clustering organization needs to select CHs and build TDMA schedules. Besides this, the first round when the network starts to work also needs to organize nodes into clusters. Each member node sends its packet to the CH once per frame during its allocated transmission slot. The number of frames of each CH is determined by the number of the nodes which vary from cluster to cluster.

When the member nodes send data to the CH, each member node uses power control so that the needed minimum power can be applied to reduce energy dissipation. This needed power to reach the CH can be acquired by the location information and (4.1). Furthermore, the radio components of each member node are turned off until its own allocated transmission slot. Besides energy savings, the application of a TDMA schedule is also an efficient way to use the limited bandwidth, thereby reducing the latency [14].

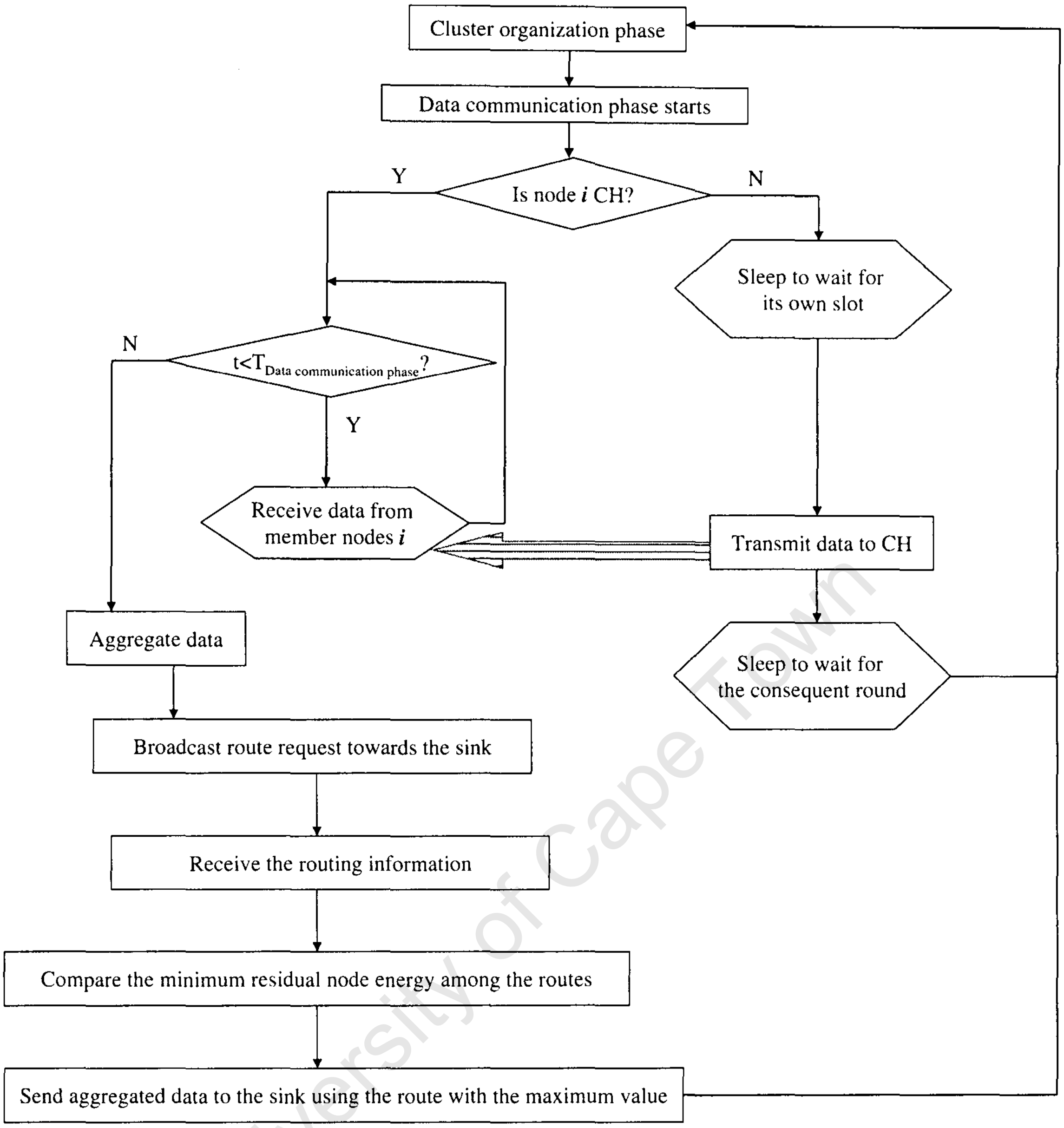


Figure 4-6. Flow chart of data communication of BRPC.

After the CH has collected all data from the member nodes, it performs a data aggregation process to aggregate the data in a ratio of λ:1. This means that for every '1' unit of data that must be sent to the sink when no data aggregation is performed, only 'λ' (which is smaller than 1) units must be sent to the sink after aggregation is performed.

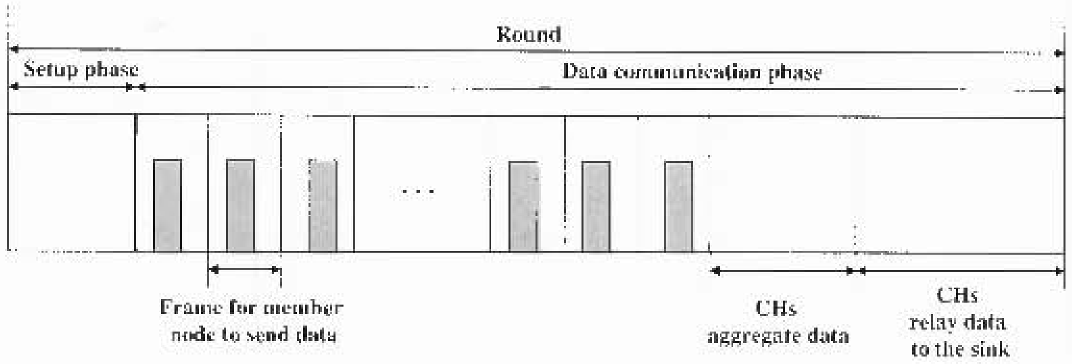


Figure 4-7. Timeline of BRPC.

After aggregating the data, the CH relays the aggregated data to the sink over a multi-hop route. Much research has been conducted to select an energy-efficient route to relay the data. Typically, a shortest route [100-105] or a minimum-energy route [106-117] is used [118]. A shortest route means that the route has a minimum number of hops from a source node to a destination among all valid routes. A minimum-energy route means that the route needs minimum energy to relay data from a source node to a destination among all valid routes. Both of these two types of routes save energy of routing data. The route selected to relay the data, however, may be used frequently and the nodes in the route will, therefore, drain energy rapidly. To prevent the nodes from dying prematurely which may disconnect the network and degrade the network performance, the power consumption needs to be better balanced among the nodes [57, 119, 120, 121]. BRPC, therefore, selects the route based on residual node energy consideration.

At each round, different nodes become CHs, with the result that the cached backbone routing information needs to be updated in the new CH for each round. Each time a CH is ready to send the aggregated data to the sink, it broadcasts a routing request. To avoid any loop and to reduce the flooding, it only allows the CHs nearer the sink (which can be determined from the cluster ID, a lower cluster ID means a cluster nearer the sink) to relay this request. When the sink is reached, the routing information is sent back to the source CH, which usually receives more than one valid route. The source CH then selects a route based on residual node energy. The minimum residual energy of the CH in the route is set as a comparative parameter. The route with the maximum value is selected to relay the data. This selection exempts the CHs with lower residual energy from relaying data and better balances power consumption among the CHs and routes [96, 97, 119]. A more detailed example is shown in Fig. 4-8. The source CH called CHS may find some routes to reach the sink. To meet the requirement of route selection in BRPC, the one consisting of (*CHS-1-4-7-Sink*) will be selected to relay the data.

4.2.4 Equalizing cluster lifetime

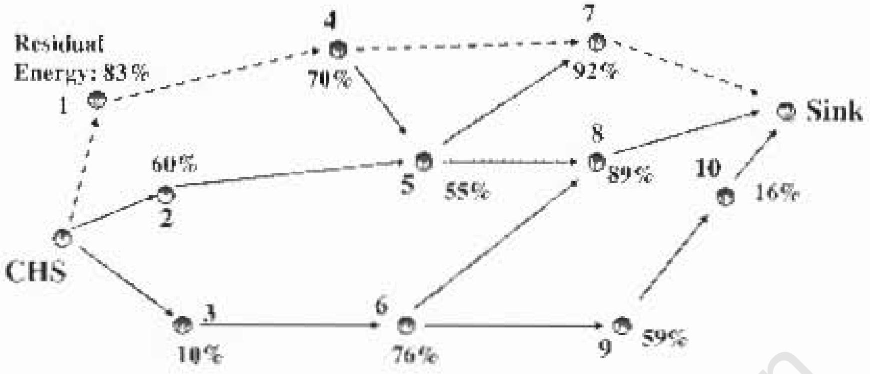


Figure 4-8. Selecting route based on residual node energy consideration.

Most existing clustering algorithms organize clusters with similar size without considering the directional data traffic towards the sink, thereby failing to optimize energy efficiency. BRPC therefore aims at equalizing cluster lifetime by taking directional data traffic into account. BRPC organizes clusters in such a way that their sizes are proportional to their respective total power consumption. The total energy stored ($\sum E_{Cluster}$) in any cluster I divided by its total power consumption ($\sum P_{Cluster}$) is the lifespan of the cluster. If (4.5) can be observed, in which *const* means constant, the cluster lifetimes are equalized.

$$\frac{\sum_{I} E_{Cluster}}{\sum_{I} P_{Cluster}} \approx const \qquad (4.5)$$

This subsection details the method to equalize the cluster lifetime. Following the analysis of the relationship between the cluster size and the directional data traffic towards the sink, the inter-cluster connectivity and routing rule in BRPC are addressed. The relationship between the data traffic and the cluster sizes is then investigated to equalize the cluster lifetime.

1. Cluster size and directional data traffic

Directional data traffic requires the clusters near the sink to relay more data in a WSN, where the CHs send data to the sink over multi-hop routes. From this point, the clusters nearer the sink need to be larger, so that the higher total energy in the cluster compensates for the higher burdens of the CHs. However, the increase in cluster size will accordingly increase the total power consumption of the intra-cluster communication, as more member nodes need to send data to the CH [54]. This subsection determines to increase or decrease the cluster size when it is closer to the sink by discussing the relationship between the cluster size and the directional data traffic.

It is assumed that there are two clusters, as shown in Fig. 4-9, named M and N with m and n nodes (including the CH) respectively. Each node in both clusters has initial energy of $E_{initial}$.

The CH of cluster M (CHM) receives i units of data from other CHs. After collecting all data from its member nodes, CHM performs data aggregation in the ratio of λ:1 and then relays the aggregated data and the data from other CHs to CHN. CHN also carries out the same process: (i) Receiving the data from CHM, (ii) collecting data from its member nodes, (iii) aggregating the collected data and (iv) relaying the aggregated data and the data from CHM to the next hop.

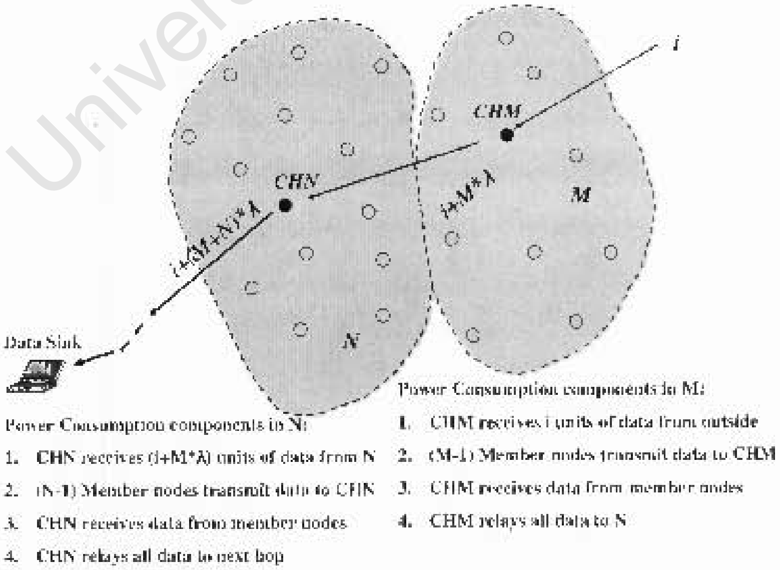


Figure 4-9. Cluster size and directional data traffic.

To simplify the analysis without losing generality, it is assumed that:

(i) A node needs P_{rea} power to receive data;

(ii) A member node needs P_{tra} power to transmit data to the CH;

(iii) A CH needs P_{agg} power to aggregate data; and

(iv) A CH needs P_{trA} power to transmit aggregated data to next hop.

To equalize the cluster lifetimes of *M* and *N*, (4.6), (4.7), and (4.8) must be observed.

$$\frac{m * E_{initial}}{\sum P_m} = \frac{n * E_{initial}}{\sum P_n} \tag{4.6}$$

$$\sum P_m = i * P_{rea} + (m-1)*(P_{tra} + P_{rea}) + m * P_{agg} + (m*\lambda + i)*P_{trA} \tag{4.7}$$

$$\sum P_n = (m*\lambda + i)*P_{rea} + (n-1)*(P_{tra} + P_{rea}) + n * P_{agg} + ((m+n)*\lambda + i)*P_{trA} \tag{4.8}$$

where $\sum P_m$ and $\sum P_n$ are the total power consumption of clusters *M* and *N*, respectively.

Without loss of generality, it is further assumed m=15, P_{rea}=50×10^{-9} (W), P_{tra}=75×10^{-9} (W) (transmission power for 50m hop distance), P_{trA}=114×10^{-9} (W) (transmission power for 80m hop distance), and P_{agg}=5×10^{-9} (W), (4.9) can be acquired from (4.6)-(4.8). The relationship between the data amount *i* that *CHM* receives from outside, data aggregation rate λ and the cluster size *n* according to (4.9) are shown in Figs. 4-10 and 4-11.

$$i = \frac{36900 * \lambda + 125 * n - 1875}{(n-15)*164} \tag{4.9}$$

Fig. 4-10 (when $n<m$) shows that the lifetimes of cluster *M* and *N* can be equalized only when the received amount of data *i* by *CHM* is less than zero, which obviously is impossible. Fig. 4-11 (when $n>m$) shows that when *N* has a larger size than *M*, the lifetime of *M* and *N* can be equalized when such amount of data *i* have been received by *CHM* from other clusters at different values of λ. Therefore, it can be concluded that to equalize the cluster lifetimes as the clusters get closer to the sink, more nodes need to be assigned to the clusters that are nearer the sink.

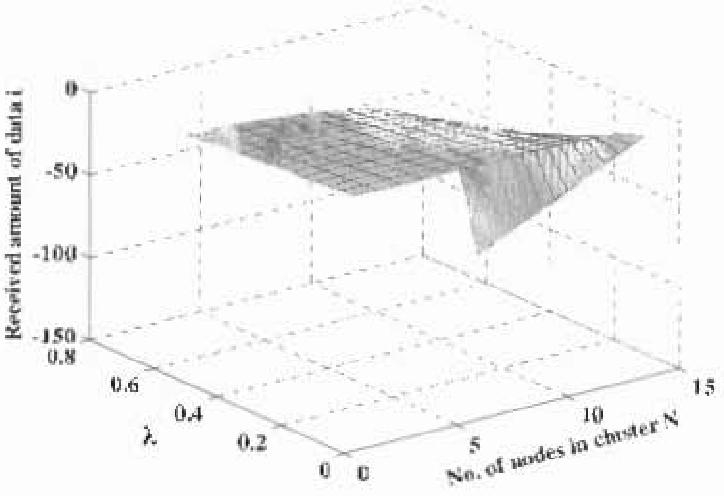


Figure 4-10. Received data i vs. data aggregation rate λ when $n<m$.

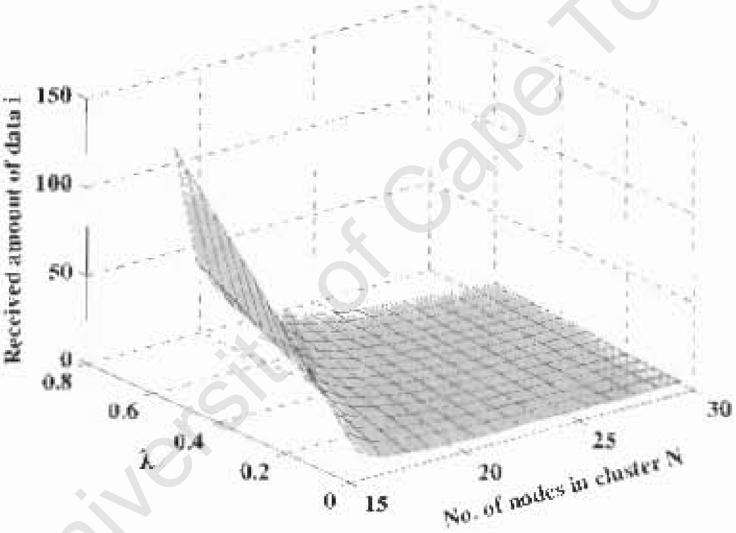


Figure 4-11. Received data i vs. data aggregation rate λ when n>m.

2. Inter-cluster connectivity and routing rules

It is important to guarantee the connectivity between the CHs as all CHs need to relay the aggregated data to the sink. This connectivity is relative to the transmission range of the nodes. As the transmission range of the node is variable within the maximum transmission range, the selected transmission range among CHs may vary to ensure a certain degree of connectivity and to control interference [42].

To reduce the interference from other clusters, it is suggested that the minimum possible power level to reach a destination is used [30]. As the route that every CH applies to relay data is always directional towards the sink, BRPC only allows the neighboring CHs nearer the sink to relay the message of routing request, thereby reducing the flooding during the inter-cluster routing discovery and resulting in a more 'direct' route to the sink.

The cluster sizes increase when the clusters are closer to the sink. This size increment, however, needs to be subject to the limitation of the maximum transmission range of the sensor nodes. Otherwise, some data may be lost during the data communication phase. To guarantee the connectivity between clusters, a CH at least needs to find one CH within its maximum transmission range. This question has also been addressed in Geographical Adaptive Fidelity (GAF) [122]. The nodes in GAF are virtually organized into square grids with the size of $r \times r$. In each grid only one node is active. To guarantee the connectivity between the grids, the inter-cluster transmission range $R_t = (1+\sqrt{5}) \times r$ needs to be assigned to the active nodes. Unlike the square grids in GAF, the grids in BRPC are rectangular. The maximum transmission range of the nodes is referred to as R_{max}. To guarantee that each CH can find at least one neighboring CH which is nearer the sink, the cluster size should be subject to this R_{max}.

It is assumed that there are four neighboring clusters, as shown in Fig. 4-12. The clusters are named according to their sides, i.e. a cluster with sides a_i and b_j is designated a_ib_j. Clusters a_ib_j, $a_{i-1}b_j$, a_ib_{j-1} are the neighboring clusters of cluster $a_{i-1}b_{j-1}$ that are nearer the sink (judged from the cluster ID). To guarantee the inter-cluster connectivity between the CHs in this proposed algorithm, (4.10) must be observed.

$$R_{\max} > \min\left(\sqrt{b_{j-1}^{\,2} + (a_i + a_{i-1})^2}, \sqrt{a_{i-1}^{\,2} + (b_{j-1} + b_j)^2}, \sqrt{(a_{i-1} + a_i)^2 + (b_{j-1} + b_j)^2}\right) \quad (4.10)$$

3. Data traffic and cluster size

It is now known that to equalize cluster lifetimes, the cluster size needs to increase the closer it is to the sink. This subsection further discusses the method to equalize the cluster lifetime. The clustered WSN, as shown in Fig. 4-13, is used to explain this method.

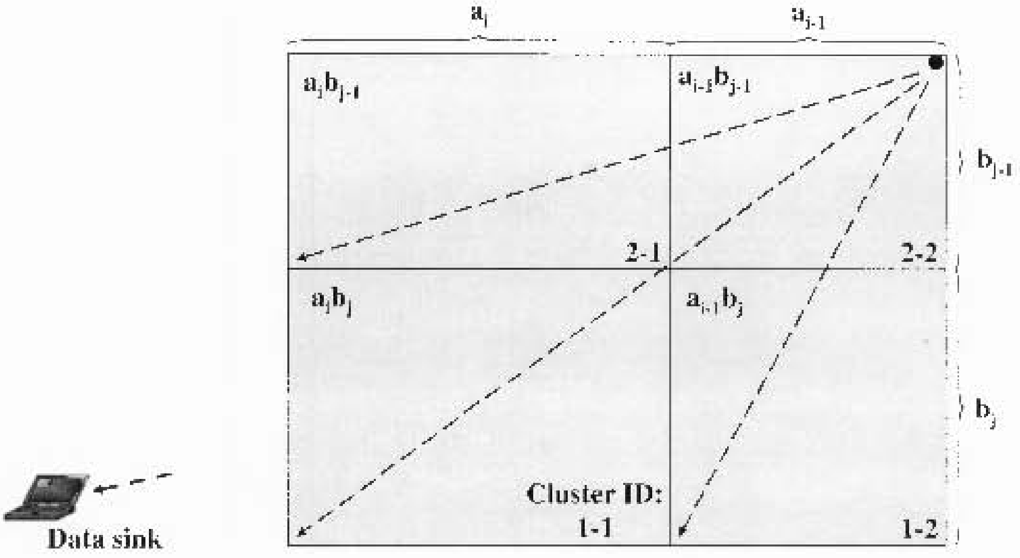


Figure 4-12. Cluster size and transmission range.

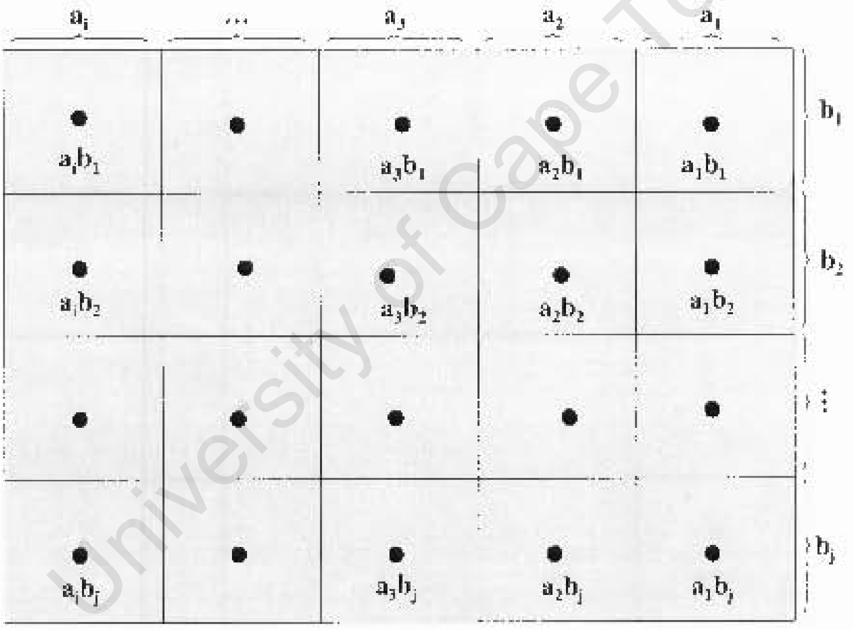
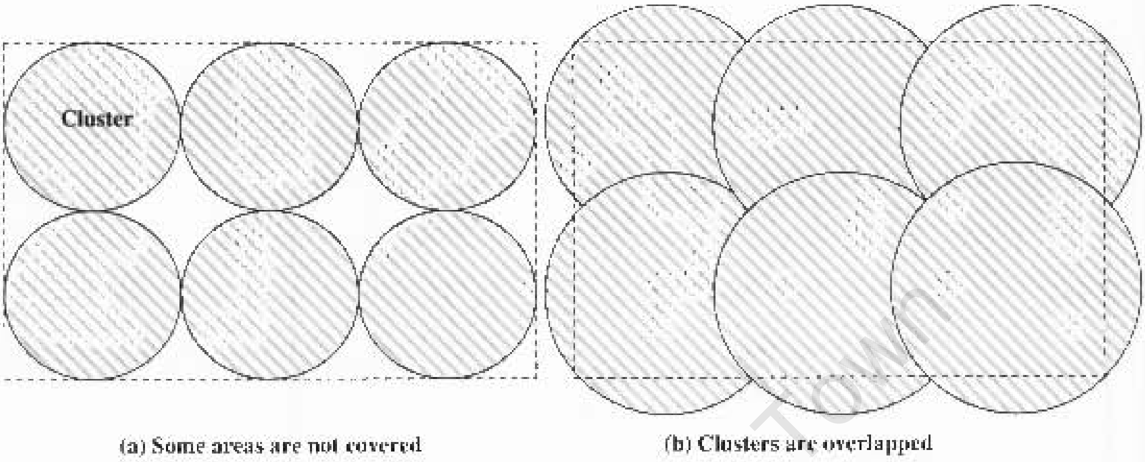


Figure 4-13. Clustered WSN to explain cluster-lifetime equalization.

In Section 4.2.2, it has been described that the nodes in BRPC are geographically organized into rectangular clusters by the sink as: Cluster_*i* (min(x_i), min(y_i), max(x_i), max(y_i)). Many clustering algorithms assume clusters with circular shape. Circular clusters as shown in Fig. 4-14, unfortunately, either cannot cover the entire network (Fig. 4-14(a)) or are overlapped by each other in covering the entire network (Fig. 4-14(b)).

The identical sensors are uniformly distributed in the network with a node density of n_d. The expected number of nodes in the cluster a_ib_j is therefore $n_d \times a_i \times b_j$. Each node has initial energy $E_{initial}$. To simplify the analysis, it is assumed that each cluster selects the node at the centre of the cluster to be CH. The transmission ranges of intra-cluster communication and inter-cluster communication of cluster a_ib_j are r_{ij} and R_{ij} respectively.



(a) Some areas are not covered

(b) Clusters are overlapped

Figure 4-14. Circular clusters in a WSN.

The power consumption of a typical cluster $P_{Cluster}$ consists of three components as shown in (4.11):

$$P_{Cluster} = P_{Intra-cluster} + P_{Agg} + P_{Inter-cluster} \tag{4.11}$$

where

i. $P_{Intra-cluster}$ is the power consumption of intra-cluster communication;

ii. P_{Agg} is the power consumption of the CH to aggregate the collected data; and

iii. $P_{Inter-cluster}$ is the power consumption of the inter-cluster communication, which is relative to the data traffic.

It is assumed that each node generates an a-bit data message and the parameter of transmitter attenuation n is 2. Then, for cluster a_ib_j, $P_{Intra-cluster}$ and P_{Agg} can be further expressed by (4.12) and (4.13) respectively, in which e_{Agg} is the radio dissipation of aggregating data.

$$P_{Intra\text{-}cluster} = (a_i \times b_j \times n_d - 1) \times a \times e_r + a \times \sum_{a_i \times b_j \times n_d - 1} \left(e_t + e_{amp} \times r_{ij}^2\right) \tag{4.12}$$

where $(a_i \times b_j \times n_d - 1) \times a \times e_r$ is the power consumption of the CH when it is receiving data from the member nodes and $a \times \sum_{a_i \times b_j \times n_d - 1} \left(e_t + e_{amp} \times r_{ij}^2\right)$ is the power consumption of the member nodes when they are transmitting data to the CH.

$$P_{Agg} = a_i \times b_j \times n_d \times e_{Agg} \tag{4.13}$$

Unlike the power consumption of intra-cluster communication $P_{Intra\ cluster}$ and of aggregating data P_{Agg}, which are only related to the respective cluster itself, the power consumption of inter-cluster communication $P_{Inter\ cluster}$ is related to the data traffic.

It has been stated that the data traffic is directional towards the sink and only the neighboring clusters nearer the sink can relay the route requests during the backbone routing discovery phase. The following outlines the analysis of the data traffic in neighboring clusters.

It is assumed that there are four neighboring clusters, as shown in Fig. 4-15. When the CH in cluster $a_{i-1}b_{j-1}$ is ready to transmit the aggregated data, it floods a route request towards the sink and only the CHs of the neighboring clusters a_ib_j, a_ib_{j-1}, and $a_{i-1}b_j$ will relay this request. This process will be iterated until the sink is reached and the routing information is sent back to the CH of cluster $a_{i-1}b_{j-1}$. To simplify the analysis, it is assumed that the probability of a neighboring CH nearer the sink of cluster $a_{i-1}b_{j-1}$ to relay the request is β. This probability, however, is a dynamic parameter for realistic application or during the simulation. The total data of the CH in cluster a_ib_j ($Data_{a_ib_j}$) that need to be relayed is separated into the data that the CH receives from others ($R(a_ib_j)$) and the data that the CH transmits to the next hop ($T(a_ib_j)$), as shown in (4.14).

$$Data_{a_ib_j} = T(a_ib_j) + R(a_ib_j) \tag{4.14}$$

The total data traffic that the CH in cluster a_ib_j receives from its neighboring cluster is shown in Table 4-1.

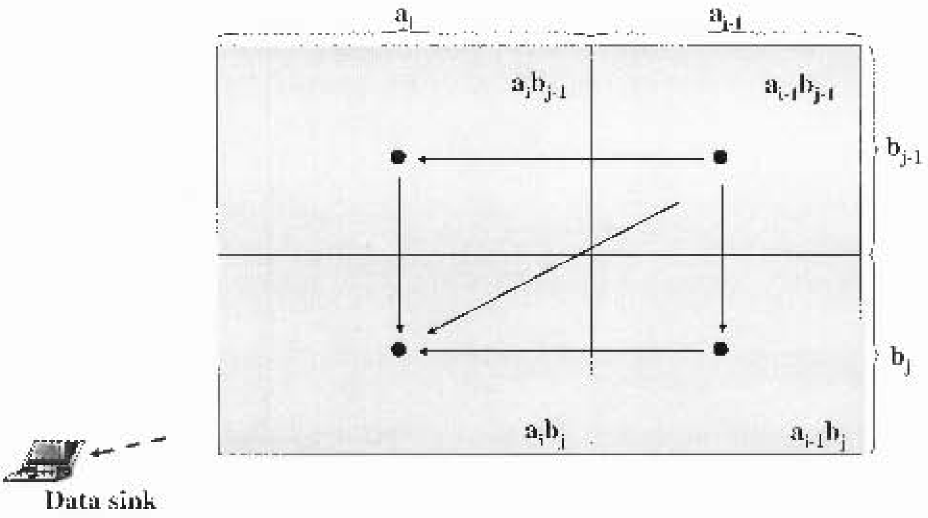


Figure 4-15. Different data traffic in neighboring clusters.

Table 4-1. The possible data traffic to cluster a_ib_j.

Source CH	**Probability of the data traffic sent to a_ib_j**
CH in $a_{i-1}b_{j-1}$ ($CH_{(i-1,j-1)}$)	$\beta+\beta\times\beta+\beta\times\beta$ (either by one hop or by two hops)
$CH_{(i,j-1)}$	β (by one hop)
$CH_{(i-1,j)}$	β (by one hop)
$CH_{(others)}$	0

From Table 8, (4.15) and (4.16) can be acquired:

$$
\begin{aligned}
R(a_ib_j) &= T(a_{i-1}b_j)\times\beta+T(a_ib_{j-1})\times\beta \\
&+T(a_{i-1}b_{j-1})\times\beta+2\times T(a_{i-1}b_{j-1})\times\beta^2) \\
&= \sum_{(m>i-1)}\sum_{(n>j-1)}((i+j-m-n)\times a_m a_n\times\beta^{i+j-m-n}
\end{aligned}
\tag{4.15}
$$

$$
\begin{aligned}
T(a_ib_j) &= \lambda\times a_i\times b_j\times n_d\times a+R(a_ib_j)= \\
&= \lambda\times a_i\times b_j\times n_d\times a+\sum_{(m>i-1)}\sum_{(n>j-1)}((i+j-m-n)\times a_m a_n\times\beta^{i+j-m-n}
\end{aligned}
\tag{4.16}
$$

The total power consumption of the CH $P_{Inter-cluster}$ in cluster a_ib_j to receive data $R(a_ib_j)\times e_r$ and to transmit data $T(a_ib_j)\times\left(e_t+e_{amp}\times R_{ij}^{\,2}\right)$ is then determined by (4.17).

$$P_{inter\ cluster} = R(a_i b_j) \times e_r + T(a_i b_j) \times \left(e_t + e_{amp} \times R_{ij}^{\ 2}\right) \tag{4.17}$$

The total power consumption of cluster $a_i b_j$ shown in (4.11) can therefore be obtained.

The total initial energy store in cluster $a_i b_j$ is (4.18):

$$E_{Cluster} = a_i \times b_j \times n_d \times E_{initial} \tag{4.18}$$

By acquiring (4.19), the cluster lifetimes are equalized.

$$\frac{E_{cluster}}{P_{Cluster}} = const \tag{4.19}$$

4.2.5 Reducing power consumption by maintaining CHs in central area of a cluster

Some existing clustering algorithms have maintained the CH in the central area of a cluster by re-clustering nodes with resultant ripple effect. This maintenance, unfortunately, fails to optimize the energy efficiency of the network, as stated in Section 2.2.2. This subsection therefore discusses the process to save energy by maintaining a CH in the central area of a cluster through re-clustering without resultant ripple effect.

As stated in Section 4.2.2, the cluster lifetime in BRPC is separated into two periods. During *period1*, only the nodes in the central area of the cluster will be considered for CH selection during each round. When any node in the central area has residual energy lower than the threshold, *period1* expires and *period2* initiates. All nodes in the cluster participate in competing to be the CH according to the residual node energy. This threshold, however, needs to be optimized. If it is too large, the nodes in the central area of the cluster cannot serve as CH long enough, thereby failing to maximize energy efficiency. If it is too small, the nodes in the central area will die prematurely during *period2*, thereby degrading the network performance.

1. Optimizing the service time of the CH in the central area of cluster

A cluster shown in Fig. 4-16 is used to discuss the optimization of the service time of the CH in the central area of the cluster. The cluster with size of *m*×*n* receives *i* units of data from other clusters. The nodes are uniformly distributed in the cluster with a node density of n_d and each node generates an a-bit data message. The data are aggregated into a ratio of λ:1. Therefore, the data amount that the CH transmits to the next hop is $(i+a\times m\times n\times n_d\times\lambda)$. The intra-cluster communication range is d and the inter-cluster communication range is D. The central area is defined by the radius of r and other areas are defined as outside area. The initial energy of each node is $E_{initial}$ and the energy threshold of *period1* is $E_{threshold}$.

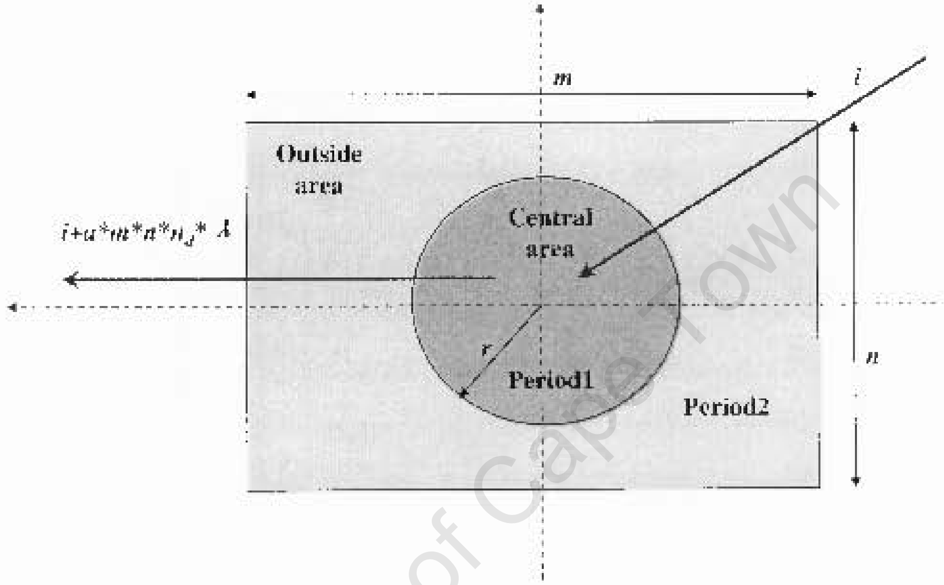


Figure 4-16. Two periods of the cluster lifetime.

It has been stated that the value of $E_{threshold}$ should be optimized to maximize the service time of the CH when it is in the central area. This optimum value means that all the nodes in the cluster will almost use up energy simultaneously.

The components of different power consumptions are defined in (4.20)-(4.25) with the definition of each item shown in Table 4-2.

The total power consumption of the cluster $P_{Cluster}$ is:

$$\begin{aligned}P_{Cluster}&=i\times e_r+(i+a\times m\times n\times n_d\times\lambda)\times\left(e_t+e_{amp}\times D^{k}\right)+a\times(m\times n\times n_d-1)\\&\times\left(e_t+e_{amp}\times d^{k}\right)+a\times(m\times n\times n_d-1)\times e_r+a\times m\times n\times n_d\times e_{Agg}\end{aligned} \tag{4.20}$$

The total power consumption of the CH P_{CH} is:

$$P_{CH} = i\times e_r + (i + a\times m\times n\times n_d\times\lambda)\times\left(e_t + e_{amp}\times D^n\right) + a\times(m\times n\times n_d - 1)\times e_r + a\times m\times n\times n_d\times e_{Agg} \tag{4.21}$$

Table 4-2. Meaning of each item in power-consumption components.

Items	**Definitions**
$P_{Cluster}$	Total power consumption of the cluster
P_{CH}	Total power consumption of the CH
$P(central)1$	Total power consumption of the central area in *period1* when the CH is in this area
$P(central)2$	Total power consumption of the central area in *period2* when the CH is not in this area
$P(outside)1$	Total power consumption of outside area in *period1* when the CH is not in this area
$P(outside)2$	Total power consumption of the outside area in *period2* when the CH is in this area
$i\times e_r$	Power consumption of the CH when receiving data from other clusters
$(i+a\times m\times n\times n_d\times\lambda)\times(e_t+e_{amp}\times D^2)$	Power consumption of the CH when transmitting data to the next hop
$a\times(m\times n\times n_d-1)\times(e_r+e_{amp}\times d^2)$	Power consumption of the member nodes when transmitting data to the CH
$a\times(\pi\times r^2\times n_d-1)\times(e_r+e_{amp}\times d^2)$	Power consumption of the member nodes in the central area when transmitting data to the CH
$a\times(m\times n\times n_d-1)\times e_r$	Power consumption of the CH when receiving data from member nodes
$a\times m\times n\times n_d\times e_{Agg}$	Power consumption of the CH when aggregating data

The total power consumption of the central area with radius of r in *period1* when the CH is in this area $P(central)1$ is:

$$P(central)1 = i\times e_r + (i + a\times m\times n\times n_d\times\lambda)\times\left(e_t + e_{amp}\times D\right)^n + a\times\left(\pi\times r^2\times n_d - 1\right)\times \left(e_t + e_{amp}\times d^n\right) + a\times(m\times n\times n_d - 1)\times e_r + a\times m\times n\times n_d\times e_{Agg} \tag{4.22}$$

The total power consumption of the central area with radius of r in *period2* when the CH is NOT in this area $P(central)2$ is:

$$P(central)2 = a \times \left(\pi \times r^2 \times n_d - 1\right) \times \left(e_t + e_{amp} \times d^n\right) \tag{4.23}$$

The total power consumption of the outside area in *period1* when the CH is in the central area $P(outside)1$ is:

$$P(outside)1 = a \times \left(\left(m \times n - \pi \times r^2\right) \times n_d - 1\right) \times \left(e_t + e_{amp} \times d^n\right) \tag{4.24}$$

The total power consumption of the outside area in *period2* when the CH is NOT in the central area $P(outside)2$ is:

$$\begin{aligned} P(outside)2 &= i \times e_r + (i + a \times m \times n \times n_d \times \lambda) \times \left(e_t + e_{amp} \times D^n\right) + \\ &a \times (m \times n \times n_d - 1) \times e_r + a \times \left(\left(m \times n - \pi \times r^2\right) \times n_d - 1\right) \times \left(e_t + e_{amp} \times d^n\right) \\ &+ a \times m \times n \times n_d \times e_{Agg} \end{aligned} \tag{4.25}$$

It has been stated that when *period1* expires, the minimum residual node energy in central area of the cluster is $E_{threshold}$. *Period1* can then be expressed by (4.26).

$$Period1 = \frac{\pi \times r^2 \times n_d \times (E_{initial} - E_{threshold})}{P(central)1} \tag{4.26}$$

The lifetime of the nodes in the central area in *period2*, which is referred to as *Period2(central)*, is described in (4.27).

$$Period\,2(central) = \frac{\pi \times r^2 \times n_d \times E_{threshold}}{P(central)2} \tag{4.27}$$

Period2 can be expressed by (4.28).

$$Period2 = \frac{\left(m \times n - \pi \times r^2\right) \times n_d \times E_{initial} - Period1 \times P(outside)1}{P(outside)2} \tag{4.28}$$

To make the nodes in the central area and the outside area of the cluster have the same lifetime, (4.29) must be observed.

$$\begin{aligned}&Period\,2(central) = Period\,2 \Rightarrow \\ &\frac{\pi \times r^2 \times n_d \times E_{threshold}}{P(central)2} = \frac{\left(m \times n - \pi \times r^2\right) \times n_d \times E_{initial} - Period\,1 \times P(outside)1}{P(outside)2}\end{aligned} \tag{4.29}$$

It is assumed that $a=1$(bit), $m=n=50$, $n_d=0.01$ (node/m^2), $\lambda=0.2$, $E_{initial}=2$ (J), $d=75$(m) and $D=120$(m). Equation (4.30) can then be derived from (4.29), in which *m1*, *n1*, *m2* and *n2* are expressed by (4.31), (4.32), (4.33) and (4.34), respectively.

$$\frac{m1}{n1} = \frac{m2}{n2} \tag{4.30}$$

$$m1 = \pi \times r^2 \times n_d \times E_{threshold} = 3.14 \times 10^{-2} \times r^2 \times E_{threshold} \tag{4.31}$$

$$n1 = P(central)2 = 3.33625 \times 10^{-9} \times r^2 - 1.0625 \times 10^{-7} \tag{4.32}$$

$$\begin{aligned}m2 &= (m \times n - \pi \times r^2) \times n_d \times E_{initial} - Period1 \times P(outside)1 = 50 - 6.28 \times 10^{-2} \times r^2 - \\ &\frac{1.6014 \times 10^{-7} \times r^2 - 2.095165 \times 10^{-10} \times r^4 - \left(8.007 \times 10^{-8} \times r^2 + 1.0475825 \times 10^{-10} \times r^4\right) \times E_{threshold}}{i \times 36.9568 \times 10^{-8} + 2.81659 \times 10^{-6} + 3.33625 \times 10^{-9} \times r^2}\end{aligned} \tag{4.33}$$

$$n2 = P(outside)2 = 36.9568 \times 10^{-8} \times i + 5.47284 \times 10^{-6} - 3.33625 \times 10^{-9} \times r^2 \tag{4.44}$$

Equation (4.35) can be derived from (4.30), in which *m3*, *m4*, *n3* and *n4* are expressed by (4.36), (4.37), (4.38) and (4.39) respectively.

$$n1 \times m2 = n2 \times m1 \Rightarrow E_{threshold} = \frac{m3 - m4}{n3 - n4} \tag{4.35}$$

$$m3 = 50 - 6.28 \times 10^{-2} \times r^2 \tag{4.36}$$

$$m4 = \frac{1.6014 \times 10^{-7} \times r^2 - 2.095165 \times 10^{-10} \times r^4}{i \times 36.9568 \times 10^{-8} + 2.81659 \times 10^{-6} + 3.33625 \times 10^{-9} \times r^2} \tag{4.37}$$

$$n3 = \frac{\left(36.9568 \times 10^{-8} \times i + 5.47284 \times 10^{-6} - 3.33625 \times 10^{-9} \times r^2\right) \times 3.14 \times 10^{-2} \times r^2}{3.33625 \times 10^{-9} \times r^2 - 1.0625 \times 10^{-7}} \tag{4.38}$$

$$n4 = \frac{8.007 \times 10^{-8} \times r^2 + 1.0475825 \times 10^{-10} \times r^4}{i \times 36.9568 \times 10^{-8} + 2.81659 \times 10^{-6} + 3.33625 \times 10^{-9} \times r^2} \tag{4.39}$$

The result of (4.35) is shown in Fig. 4-17. This means that for a cluster with the size of 50m×50m and with 2J initial energy in each node, the nodes in the network will die simultaneously when the condition of Fig. 4-17 is satisfied. The relationship between the radius r of the central area and $E_{threshold}$ has now been built and the nodes in the central area can serve as CH for as long as possible.

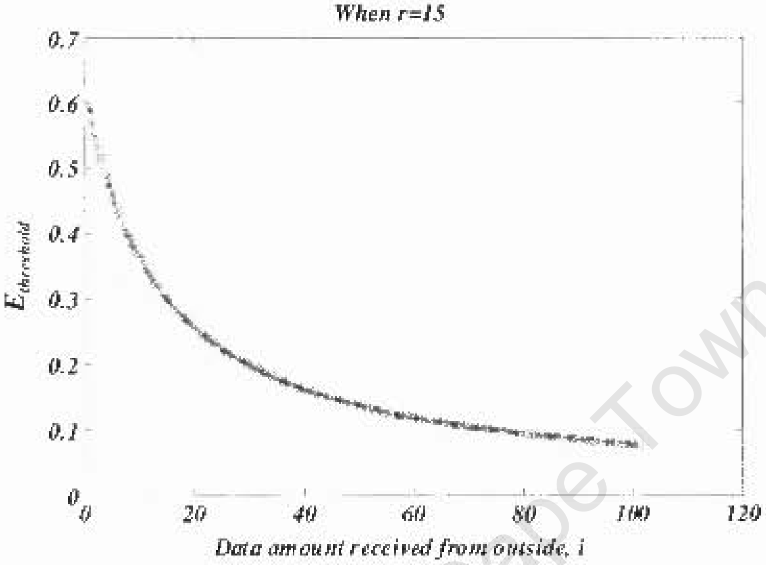


Figure 4-17. Relationship between $E_{threshold}$ and i when r=15.

2. Power consumption of intra- and inter-cluster communication

This subsection discusses the improvement gained by maintaining the CHs in the central area of the cluster. The mathematical model is shown in Fig. 4-18. Cluster M and N are neighboring clusters with size of $a_1 \times b$ and $a_2 \times b$ respectively. The CH in M (*CHM*) is supposed to be at the centre of the cluster. The coordinates of *CHM* are $((a_1+a_2)/2, 0)$. It is assumed that *CHN* is the next hop of *CHM*. The radius of the central area in cluster N is R. Other areas in cluster N are referred to as outside areas.

The lifetime of cluster N is separated into two periods. *Period1* only selects the nodes in the central area of the cluster to be a CH. When any node reaches $E_{threshold}$, *period1* expires and *period2* commences. Because the CH role is rotated frequently, the higher burden of the CH is therefore well distributed among the nodes. When the residual energy of any node in the central area in *period1* reaches $E_{threshold}$, almost all nodes in the central area will have similar residual energy as $E_{threshold}$. Therefore, the nodes in the outside area have more residual energy than those in the central area and thus have a greater chance to be CH during *period2*.

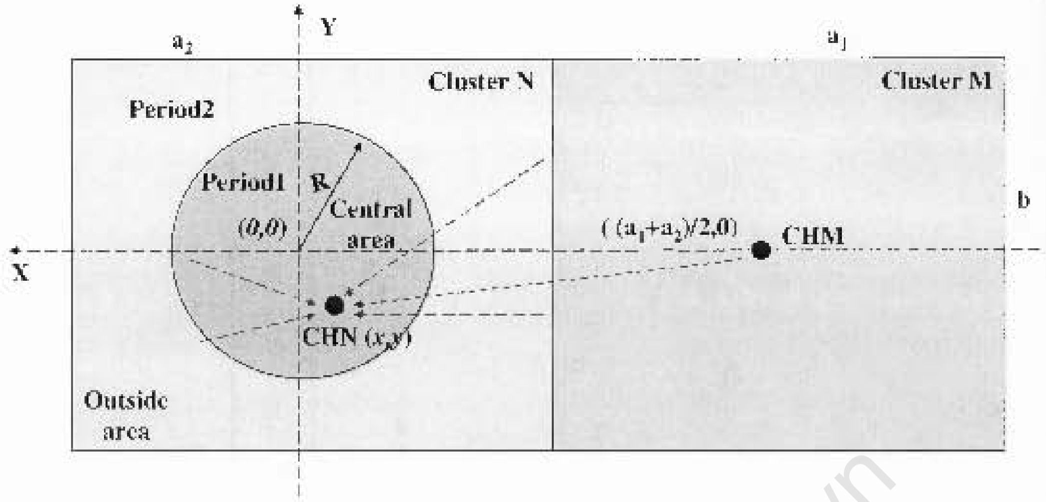


Figure 4-18. Mathematical model of intra- and inter-cluster communication.

This means that during *period1*, the power consumption of intra-cluster is reduced. However, this power consumption is increased in *period2*. One can therefore imagine that BRPC does not save power consumption of intra-cluster communication significantly. Also, from Fig. 4-18, it can be seen that, the power consumption of locating the CH in the outside area of the cluster is similar when it is randomly located in the whole cluster. This means that, BRPC only improves energy efficiency of inter-clustering communication during *period1*. The following discusses this improvement.

The power consumption of the communication between CHM and CHN is only affected by the distance of the hop. To simplify the analysis, the following equations only discuss the decrease in the expected square of the distance between these two CHs. During *period1*, the CH is located in the central area of cluster N, (4.40) can then be obtained. When CHN is located in the central area of the cluster, the expected square of the distance of this hop between CHM and CHN $E(d^2_{central})$ is shown in (4.41):

$$x^2 + y^2 \le R^2 \tag{4.40}$$

$$
\begin{aligned}
E(d^2{}_{central}) &= \frac{\int_{-R}^{R} dx \int_{0}^{\sqrt{R^2-x^2}} \left(\left(x - \frac{a_1+a_2}{2} \right)^2 + y^2 \right) dy}{\frac{\pi}{2} R^2} \\
&= \left(\int_{-\pi}^{0} \left(R^4 \sin^2 t \cos^2 t + \frac{R^2}{4}(a_1+a_2)^2 \sin^2 t + \frac{R^4}{3} \sin^4 t - R^3 (a_1+a_2) \sin^2 t \cos t \right) dt \right) / \left(\frac{\pi}{2} R^2 \right) \\
&= \left(R^4 \int_{-\pi}^{0} \sin^2 t \cos^2 t dt + \frac{R^2}{4}(a_1+a_2)^2 \int_{-\pi}^{0} \sin^2 t dt + \frac{R^4}{3} \int_{-\pi}^{0} \sin^4 t dt - R^3 (a_1+a_2) \int_{-\pi}^{0} \sin^2 t \cos t dt \right) / \left(\frac{\pi}{2} R^2 \right) \\
&= \left(\frac{R^4}{8}\pi + \frac{R^2}{8}(a_1+a_2)^2 \pi + 0 + \frac{R^4}{8}\pi \right) / \left(\frac{\pi}{2} R^2 \right) = \left(\frac{\pi}{4} R^4 + \frac{\pi}{8} R^2 (a_1+a_2)^2 \right) / \left(\frac{\pi}{2} R^2 \right) = \frac{R^2}{2} + \frac{(a_1+a_2)^2}{4}
\end{aligned}
\tag{4.41}
$$

The traditional method rotates the CH role according to the residual node energy of the node so that the CH will be randomly located in the cluster. When the CH is located in the cluster randomly, the expected square of the distance of this hop between CHM and CHN $E(d^2{}_{random})$ is shown in (4.42):

$$
\begin{aligned}
E(d^2{}_{random}) &= \frac{\int_{-a2/2}^{a2/2} dx \int_{0}^{b/2} \left(\left(x - \frac{a_1+a_2}{2} \right)^2 + y^2 \right) dy}{a_2 \frac{b}{2}} \\
&= \left(\frac{b^3 a_2}{24} + \frac{b a_2{}^3}{24} + \frac{b a_2 (a_1+a_2)^2}{8} \right) / \left(a_2 \frac{b}{2} \right) = \frac{b^2}{12} + \frac{a_2{}^2}{12} + \frac{(a_1+a_2)^2}{4}
\end{aligned}
\tag{4.42}
$$

The decrease in the expected square of the distance by maintaining the CHN in the central area of the cluster is then (4.43):

$$
E(d^2{}_{random}) - E(d^2{}_{central}) = \frac{b^2}{12} + \frac{a_2{}^2}{12} - \frac{R^2}{2} \tag{4.43}
$$

4.3 Performance evaluation

To evaluate the performance of BRPC, two existing clustering algorithms are used for comparison. One combines LEACH_C and LEACH_F [14]. It applies LEACH_C to acquire better cluster organization and applies LEACH_F to rotate the role of the CH through re-clustering without ripple effect to achieve higher energy efficiency. This algorithm is referred to as LEACH_C_F. The other clustering algorithm groups the nodes into clusters with similar sizes. The clusters are fixed and the CHs send data to the sink over a multi-hop route, which is similar to [36]. This algorithm is referred to as Similar Size with Multi-hop Routes (SSMR).

4.3.1 Simulation scenario

Two hundred identical sensor nodes are uniformly distributed within an area of 200m×100m, with the data sink located 10m away from the network. Each node has 5J initial energy. The maximum transmission range of the node is limited within 140m. Within this limitation, the node can vary its transmission range. The energy dissipation of aggregating data, e_{Agg}, is 5nJ/bit/signal [14]. The data size is fixed at 500 bytes, including the packet overhead. The operation of the algorithms is broken up into *rounds*, where each round begins with set-up phase, followed by a communication phase when data transfers to the data sink occur. The number of frames assigned to each cluster is based on the cluster size.

4.3.2 Assessments requirement

1. First-node lifetime and network lifetime

The network lifetime has traditionally been measured up to when any node dies (*first-node lifetime*). This measurement provides a standard to evaluate how power consumption is balanced throughout the network. Network lifetime has also been measured to the time when all nodes have died (*last-node lifetime*). However, although the connectivity between the network and the data sink is usually not always lost after the first-node lifetime, it usually has been lost long before all nodes have died [54]. Besides measuring the first-node lifetime, this simulation also defines *network lifetime* to evaluate the algorithms. In BRPC and SSMR, *network lifetime* means the time up to loss of the connectivity between the network and the sink (*connectivity lifetime*). In LEACH_C_F, network lifetime means up to only 50% of nodes remain live (*50% availability lifetime*). This is because unlike the CHs in BRPC and SSMR, which relay data to the sink over multi-hop routes, the CHs in LEACH_C_F transmit data to the sink over single-hop routes so that it is assumed that all nodes in LEACH_C_F can access the data sink. This means that the nodes in LEACH_C_F do not have the limitation of maximum transmission range (140m). This 50% is determined by the maximum transmission range of the node and the network size.

2. Total remaining energy at different lifetimes

The measurement of the total remaining energy at first-node lifetime and network lifetime evaluates the balance of power consumption throughout the network and the energy efficiency of the network.

3. Distribution of average residual node energy at network lifetime

The distribution of the residual node energy is an important parameter in measuring the balance of power consumption throughout the network. If it is well balanced, all nodes will have low residual energy at network lifetime. Otherwise, some nodes will have high residual energy.

4. Data packets delivered to the sink

The number of packets that are delivered to the sink is measured to estimate the effectiveness of delivering data packets. Two types of data packets are defined. The first one is the *entire-network data,* which are delivered to the sink from the entire network where all sensors are alive. The second one is the *partial-network data*, which are delivered to the sink from part of the network where sensors are still alive after first-node lifetime. Adding the entire-network data and the partial-network data are the *network data* that are delivered to the sink up to network lifetime.

5. Energy of delivering each aggregated data packet to the sink

Dividing the total consumed energy by the total number of data packets that the sink receives up to the first-node lifetime is the actual energy of delivering each aggregated data packet, as measured by most traditional clustering algorithms. Besides measuring the actual energy of delivering each aggregated packet, this simulation also evaluates the energy of delivering each aggregated packet by dividing the total initial energy by the total number of the data packets that the sink receives up to network lifetime. The reason behind this is that, though the network still has some residual energy at network lifetime, this energy cannot be used any longer and is wasted.

6. Details of live nodes up to network lifetime

The directional data traffic towards the sink makes some clusters die prematurely in SSMR and LEACH_C_F. If the power consumption is balanced throughout the network in BRPC, the lifetimes of the clusters will be the same or quite similar. The measurement of the live nodes up to network lifetime shows how the nodes die at different rounds. This measurement provides more information on the performance of the algorithms.

7. Details of fraction of energy not yet used up to network lifetime

The measurement of the fraction of energy not yet used up to network lifetime shows how energy is consumed at different rounds. This measurement also provides the comparison of energy efficiency in delivering the packets. The rate that the network consumes energy may change after first-node lifetime in LEACH_C_F and SSMR. In LEACH_C_F, this rate will become smaller when some nodes die. However, in SSMR this rate is relative to both the number of dead clusters and the increment in power to reach the sink. This higher power is needed to guarantee the data being delivered to the sink. In BRPC, the cluster lifetime is separated into two periods, as described in Section 4.2.5. *Period1* maintains the CH in the central area of the cluster and has higher energy efficiency than that in *Period2*. All information is expected to be reflected in this measurement.

8. Details of aggregated data packets delivered to the sink up to network lifetime

This measurement shows the total number of aggregated data packets that are delivered to the sink at different number of rounds. The rate of receiving packets may change after first-node lifetime in all algorithms, as the dead clusters will no longer deliver the data to the sink. However, this rate hardly changes in BRPC, as virtually no node will die prematurely before the network lifetime.

9. Impact of transmitter-amplifier parameter, e_{amp}

Some examples in Section 2.2.1 show that the increment in transmitter-amplifier parameter, e_{amp}, exacerbates the impact of directional data traffic. Evaluating the performance of the algorithms at different e_{amp} values not only assesses the effect of e_{amp} on the directional data traffic but is also helpful to realistic application, as different reflected in different literature and applications [14, 37].

4.3.3 Simulation Results

1. First node lifetime and network lifetime

It has been stated in Section 4.2.4 that a cluster has three components of power consumption. For the first two, the power consumption of intra-cluster communication and aggregating data is only relative to the respective cluster. For the third component, the power consumption of inter-cluster communication is relative to the directional data traffic. The data aggregating rate λ greatly affects the data traffic in inter-cluster communication. When λ=0, no data will be relayed to the sink so that no inter-cluster communication will occur. Of course, no impact of directional data traffic exists. When λ=1, no data aggregation will be performed and the impact of directional data traffic will become maximum. Therefore when λ increases, the improvement of BRPC is expected to be more significant.

The first-node lifetime of different algorithms, as shown in Fig. 4-19, decreases as λ increases. When λ increases, more data need to be sent to the sink, thereby increasing power consumption of inter-cluster communication and shortening the lifetime. When λ=0.1, the first-node lifetime in LEACH_C_F, SSMR and BRPC is 3874, 5544 and 8150 rounds respectively. BRPC shows a 47% improvement and a 110% improvement over SSMR and LEACH_C_F respectively. When λ=0.8, the ratio between the first-node lifetime of LEACH_C_F, SSMR and BRPC becomes 1:1.69:4.98. The improvement is more significant than the one when λ=0.1, which is 1:1.47:2.10.

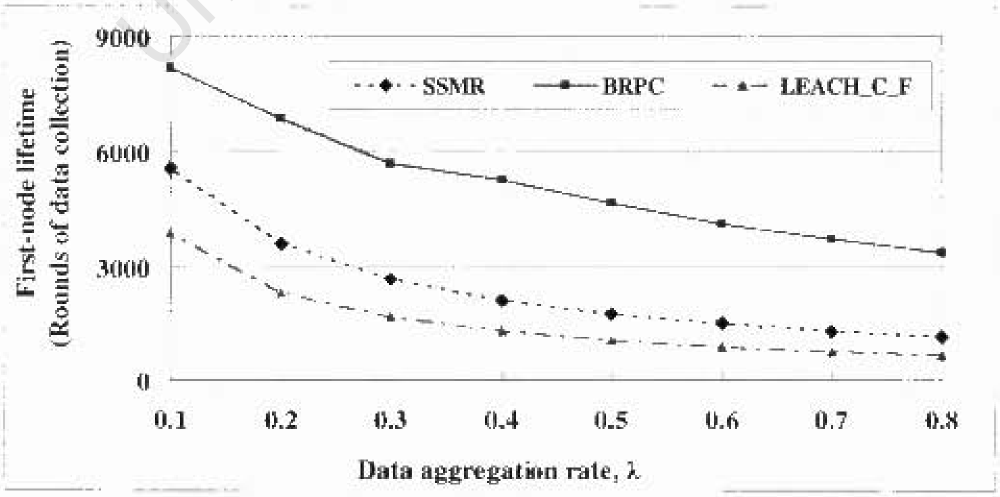


Figure 4-19. First-node lifetime at different data aggregation rate, λ.

Fig. 4-19 also shows that SSMR presents longer first node lifetime than LEACH_C_F. This is because that the multi-hop routes used to send data to the sink in SSMR is more energy efficient than the single-hop routes in LEACH_C_F. Furthermore, this result also shows that the multi-hop routes can better endure the impact of the directional data traffic.

After the first-node lifetime, though some areas are out of coverage due to the premature node death, the network is still sending data to the sink. The network lifetimes of these three algorithms are shown in Fig. 4-20. When λ=0.1, the ratio between the network lifetime of LEACH_C_F, SSMR and BRPC is 1: 0.96: 1.03. When λ=0.8 this ratio becomes 1:0.79: 1.76. This result again shows that when λ increases, the improvement of BRPC becomes more significant.

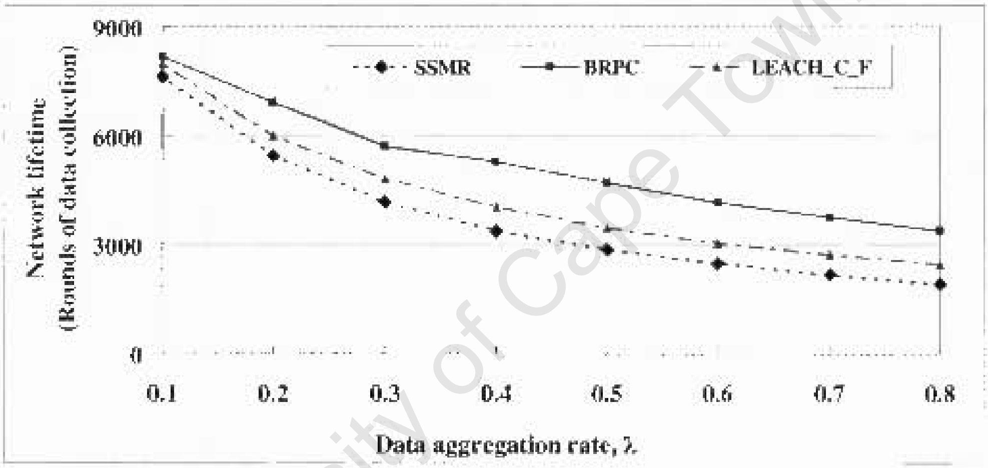


Figure 4-20. Network lifetime at different data aggregation rate, λ.

The first-node lifetime in SSMR is longer than in LEACH_C_F yet the connectivity lifetime reverses (refer to Fig. 4-19 and Fig. 4-20). In LEACH_C_F, the farthest cluster dies earlier than other clusters as it needs higher transmission power to reach the sink than others. The impact of the directional data traffic becomes smaller thereafter as other clusters need shorter transmission ranges to reach the sink. In SSMR, the cluster nearest the sink dies earlier. Its neighboring clusters, which still have residual energy and are farther away from the sink, need to increase transmission power to reach the sink to avoid any data loss, thereby increasing the power consumption. Therefore, the impact of directional data traffic becomes more serious to SSMR than to LEACH_C_F after first-node lifetime, making SSMR have a longer first-node lifetime yet a shorter network lifetime compared to LEACH_C_F.

Fig. 4-21 compares the first-node lifetime and network lifetime of each algorithm as a fraction of the network lifetime of BRPC. It can be seen that the first-node lifetime and the network lifetime of BRPC is quite similar as the ratio between them is almost 100%. BRPC actually balances power consumption throughout the network. Furthermore, BRPC presents longer first-node lifetime and network lifetime than LEACH_C_F and SSMR. This improvement becomes more significant as λ increases.

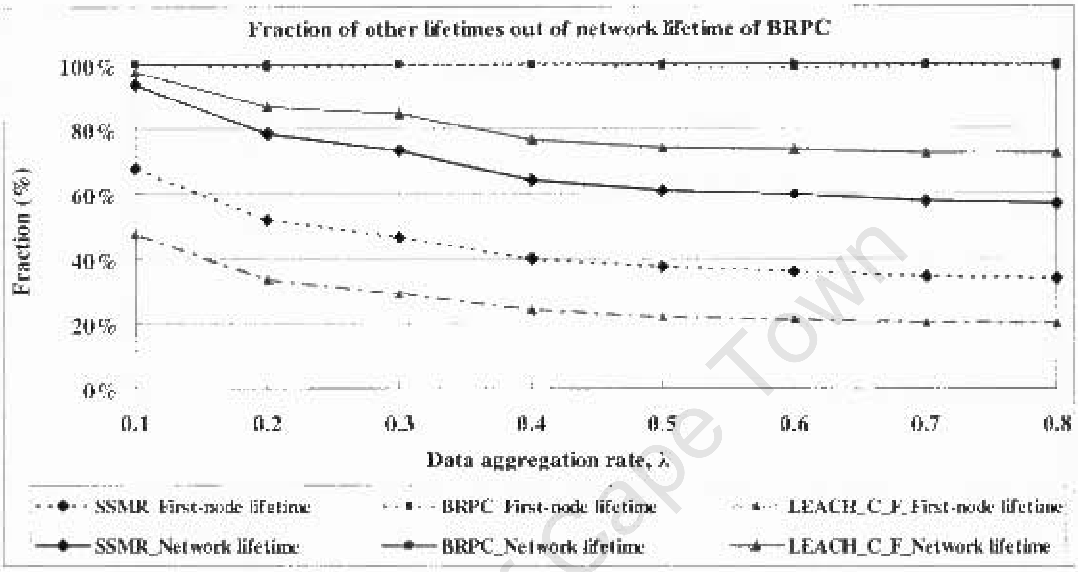


Figure 4-21. Fraction of other lifetimes out of network lifetime of BRPC.

It can also be seen from Fig. 4-21 that, the difference between the first-node lifetime and the network lifetime in LEACH_C_F is greater than in SSMR. This result again shows that a multi-hop route receives less impact from the directional data traffic.

2. Total remaining energy at different lifetimes

The total remaining energy at first-node lifetime is shown in Fig. 4-22. A lower remaining energy means better balance of power consumption throughout the network. BRPC balances power consumption much better than LEACH_C_F and SSMR as it presents much lower remaining energy. When λ=0.1, BRPC only remains 92J energy in the network, whereas LEACH_C_F and SSMR remain 461J and 388J energy respectively. This improvement becomes more significant when λ=0.8, at which LEACH_C_F, SSMR and BRPC have 645J, 651J and 101J energy remained respectively. This is because when λ=0.8, the directional data traffic makes the first node die much earlier than when λ=0.1 in LEACH_C_F and SSMR. Some researchers define that the network dies when any node runs out of energy [36]. The assumption behind this is that when any node dies, the network performance is deteriorated. From this definition, BRPC presents an improvement by more than 2 times compared to LEACH_C_F and SSMR when λ=0.8.

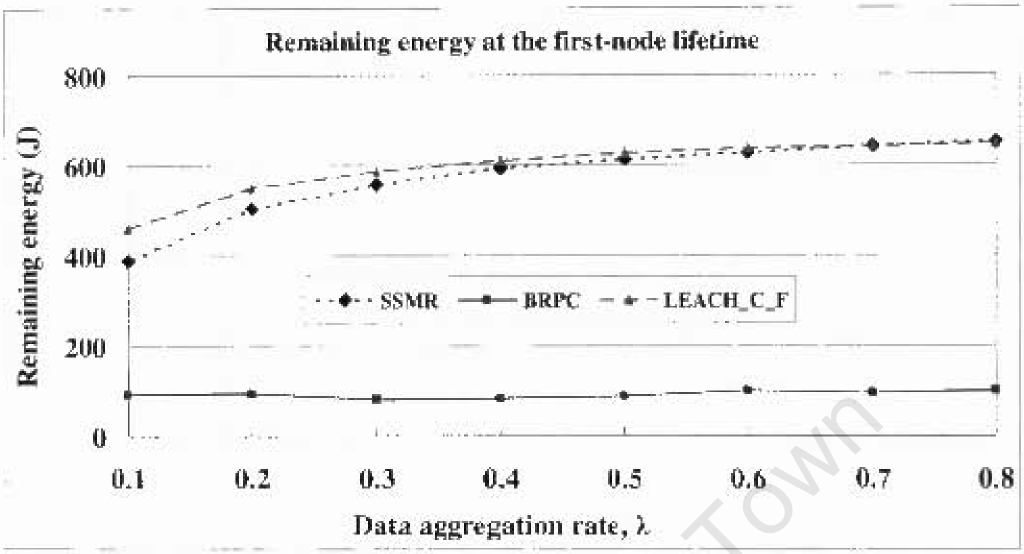


Figure 4-22. Remaining energy at the first-node lifetime.

The remaining energy at network lifetime is shown in Fig. 4-23. This remaining energy is also referred to as wasted energy. This is because after the network lifetime, this remaining energy cannot be used by the network any longer. BRPC has much less remaining energy than LEACH_C_F and SSMR. Comparing Fig. 4-22 with Fig. 4-23, it can be seen that at first-node lifetime, SSMR has less remaining energy than LEACH_C_F whereas it inverses at network lifetime. This is because more energy is consumed after the first-node lifetime in LEACH_C_F than in SSMR. The performance of LEACH_C_F and SSMR will be further discussed by evaluating the total delivered data packets and energy in delivering each data packet.

3. Distribution of average residual node energy at network lifetime

The distribution of the remaining node energy at network lifetime when λ=0.4 is shown in Fig. 4-24. It can be seen that almost 70% and 20% of the nodes in BRPC have residual energy below 10% and below 20%, respectively. However, in LEACH_C_F and SSMR, more nodes have significant energy left. In LEACH_C_F, about 50% of nodes have more than 40% energy left. In SSMR, more than 50% nodes have more than 50% energy left.

This means that the power consumption is much better balanced in BRPC than in LEACH_C_F and SSMR.

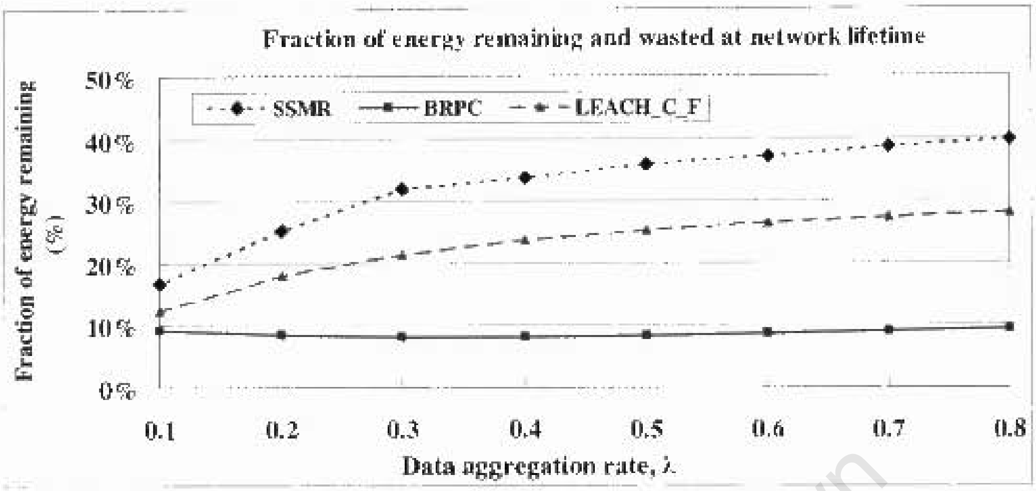


Figure 4-23. Fraction of energy remaining and wasted at network lifetime.

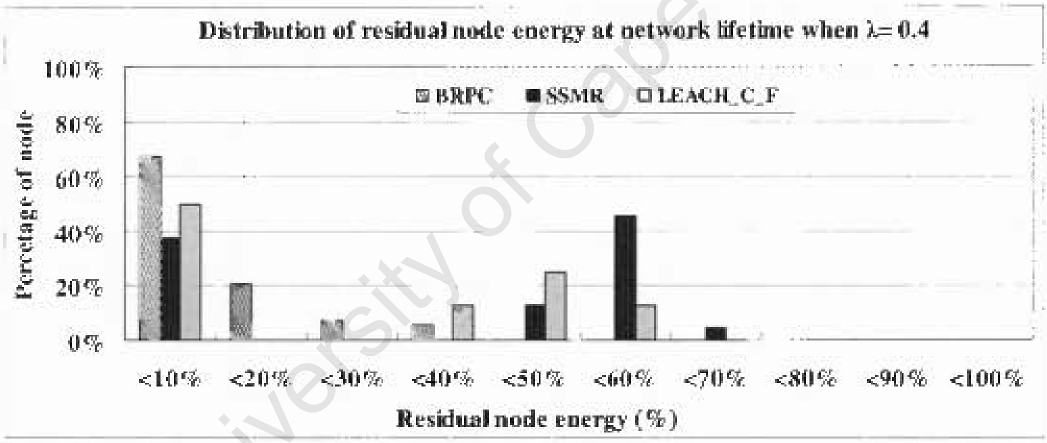


Figure 4-24. Distribution of residual node energy at network lifetime.

4. Data packets delivered to the sink

The total number of aggregated data packets delivered to the sink up to network lifetime is shown in Fig. 4-25. BRPC displays the best performance of delivering data packets. When λ increases, the size of the aggregated data packet becomes greater. Energy of delivering each packet in inter-cluster communication increases so that the total aggregated data packets delivered to the sink decreases. When λ = 0.1, LEACH_C_F, SSMR and BRPC deliver 55406, 59016 and 65270 aggregated data packets respectively with the ratio between them of 1:1.07:1.18. When λ = 0.8, this ratio becomes 1.07:1:1.84 and the improvement of BRPC is more significant than when λ = 0.1.

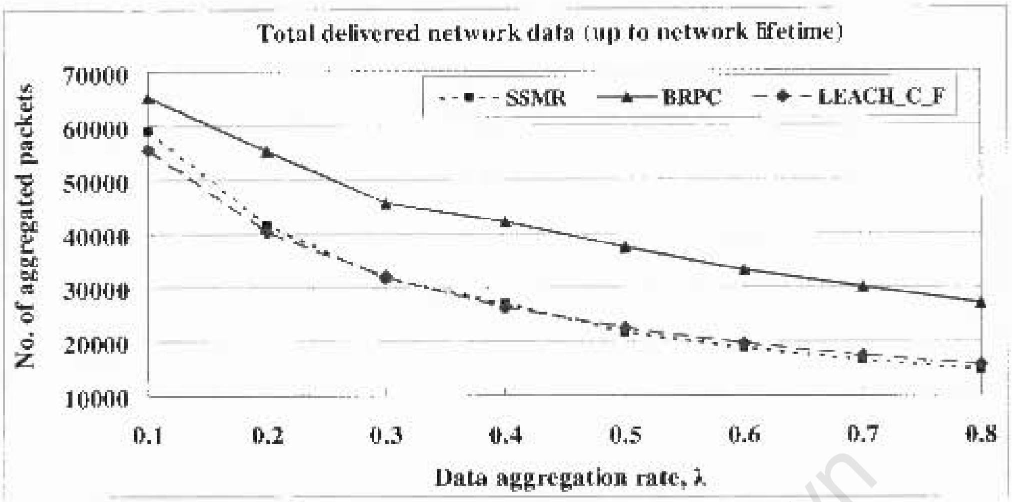


Figure 4-25. Number of aggregated network data delivered to the sink.

From Fig. 4-25, it can be seen that LEACH_C_F delivers a similar number of aggregated packets as SSMR up to network lifetime. This, however, does not necessarily mean that the performance of delivering data packets is similar in LEACH_C_F as in SSMR, as they send different number of partial-network data packets and entire-network data packets to the sink.

Fig. 4-26 presents the fraction of total delivered partial-network data out of total delivered network data. Among these three algorithms, BRPC performs much better than LEACH_C_F and SSMR. This fraction in BRPC is extremely small, which means that almost all data packets that BRPC delivers to the sink are from the entire network. However, it is quite significant in LEACH_C_F and SSMR, especially in LEACH_C_F. When λ=0.4, 61%, 35% and 0% packets delivered to the sink in LEACH_C_F, SSMR and BRPC are from the partial network, respectively. Though LEACH_C_F almost delivers the same number of network data to the sink as SSMR up to network lifetime, as shown in Fig. 4-25, the higher fraction of the partial-network data that it delivers to the sink presents a poorer performance of delivering data compared to that of SSMR.

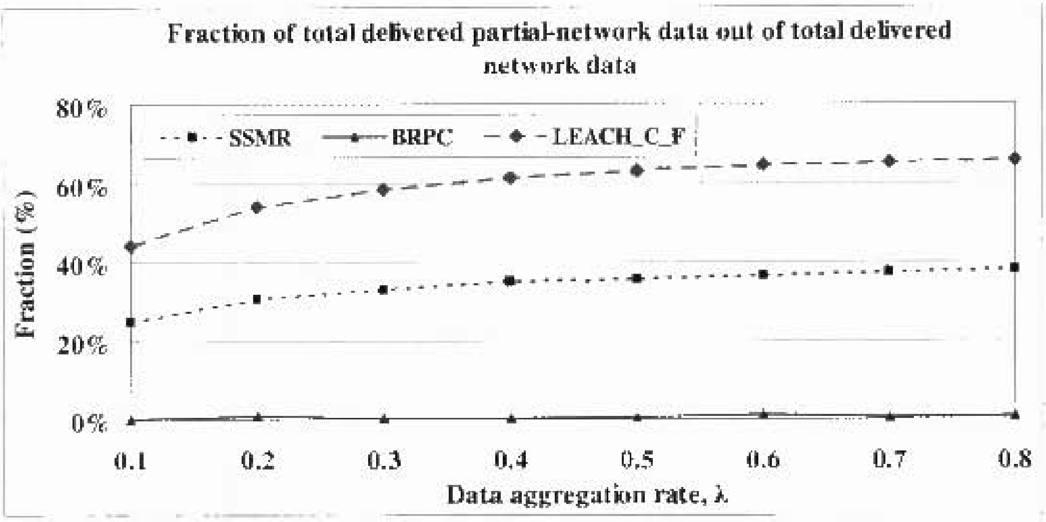


Figure 4-26. Fraction of delivered partial-network data out of network data.

The number of entire-network data packets delivered to the sink before the first-node lifetime is shown in Fig. 4-27. BRPC delivers much more entire-network data to the sink than LEACH_C_F and SSMR. When λ=0.4, LEACH_C_F, SSMR and BRPC deliver 10216, 16800 and 42064 packets respectively, i.e. BRPC delivers data packets by 4.12 times and 2.50 times compared to LEACH_C_F and SSMR, respectively. Given a network that needs high quality of service which requires the information from the entire network, BRPC is significant for realistic application.

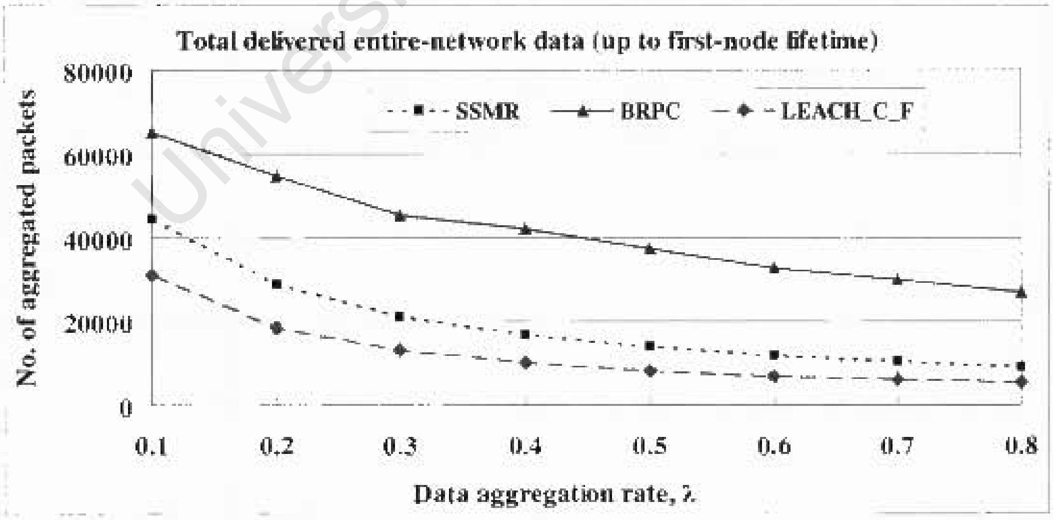


Figure 4-27. Total delivered entire-network data up to first-node lifetime.

5. Energy needed for delivering each packet to the sink

As described in Section 4.3.2, two types of energy needed to deliver each aggregated packet to the sink are defined. The first one is used to measure the actual energy of delivering each entire-network data packet by dividing the total energy consumed with the total packets that the network delivers to the sink up to the first-node lifetime. The second one considers that the residual energy cannot be used any longer after the network lifetime, and is defined by dividing the total initial network energy with the total network data packets that the network delivers to the sink up to network lifetime.

The energy to deliver each entire-network data packet to the sink is shown in Fig. 4-28. The needed energy increases according to data aggregation rate, λ. LEACH_C_F needs more energy to deliver each data packet as it uses a single-hop route to transmit the data to the sink. BRPC needs less energy of delivering each data than SSMR, though the improvement is not significant, especially when λ is low. This is because, when λ is small, the improvement of SSMR in inter-cluster communication becomes less significant (refer to Section 4.2.5), thereby making them with similar energy efficiency.

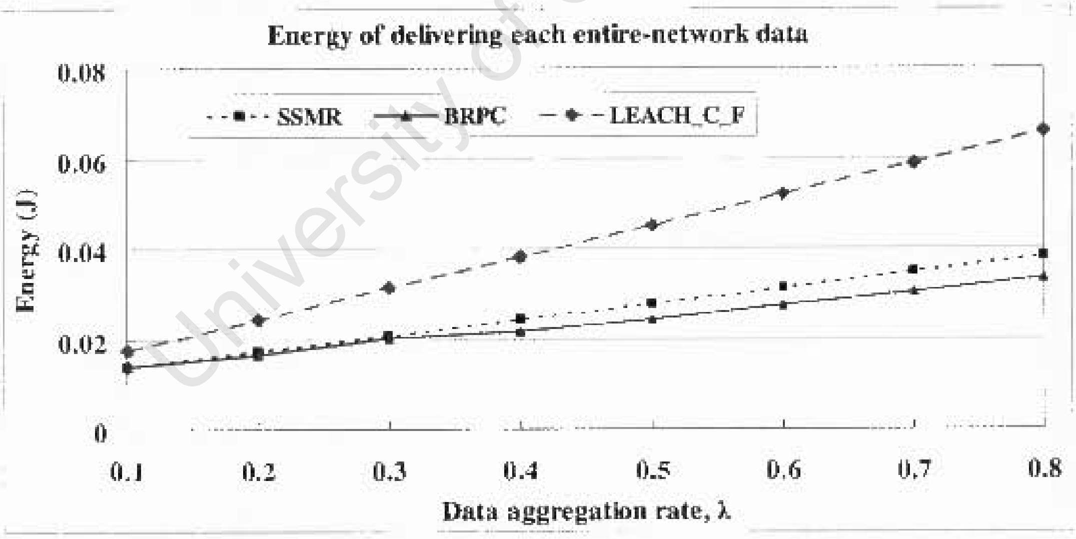


Figure 4-28. Energy of delivering each entire-network data packet.

The average energy to deliver network data up to network lifetime is shown in Fig. 4-29. BRPC shows higher energy efficiency than LEACH_C_F and SSMR. This improvement becomes more significant when λ increases. When λ=0.1, LEACH_C_F, SSMR, and BRPC need 0.018J, 0.017J and 0.015J energy respectively to deliver each aggregated packet to the sink with the ratio between them of 1.2:1.13: 1, which becomes 1.73:1.84:1 when λ=0.8. LEACH_C_F and SSMR present similar needed energy of delivering each network packet. However, SSMR delivers more entire-network data packets (refer to Fig. 4-27) so that it performs better than LEACH_C_F.

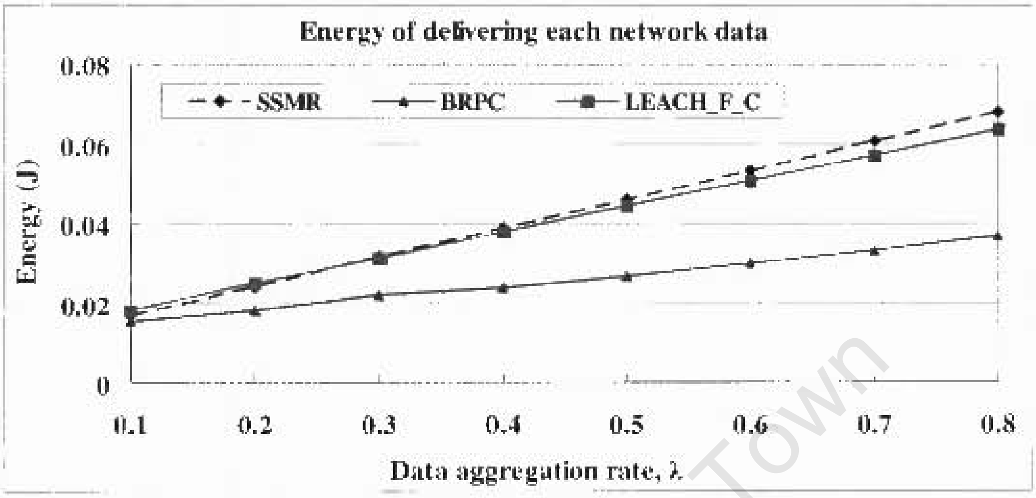


Figure 4-29. Energy of delivering each packet.

6. Details of live nodes up to network lifetime

The number of nodes that live up to network lifetime is shown in Fig. 4-30 when λ=0.6 and Fig. 4-31 when λ=0.2.

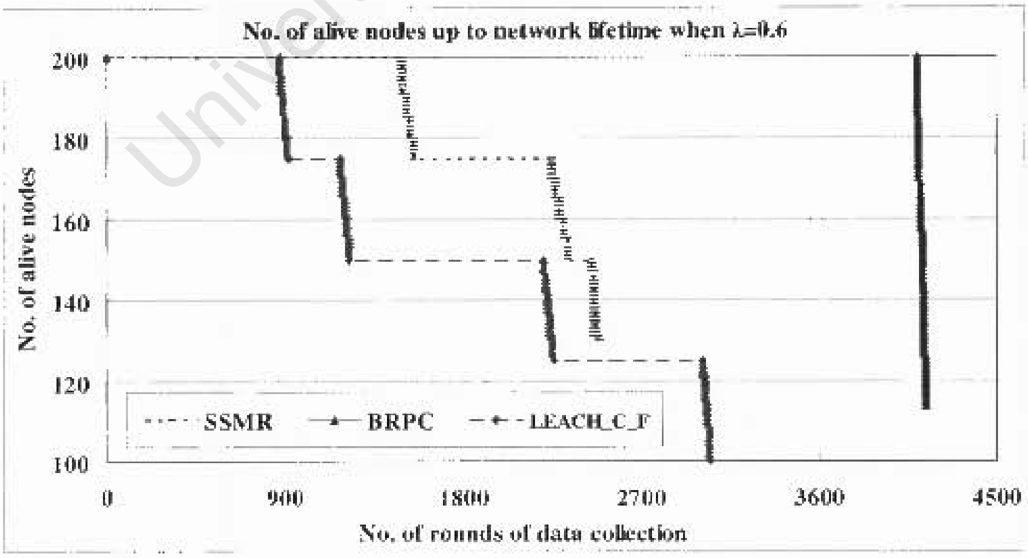


Figure 4-30. Number of live nodes up to network lifetime when λ=0. 6.

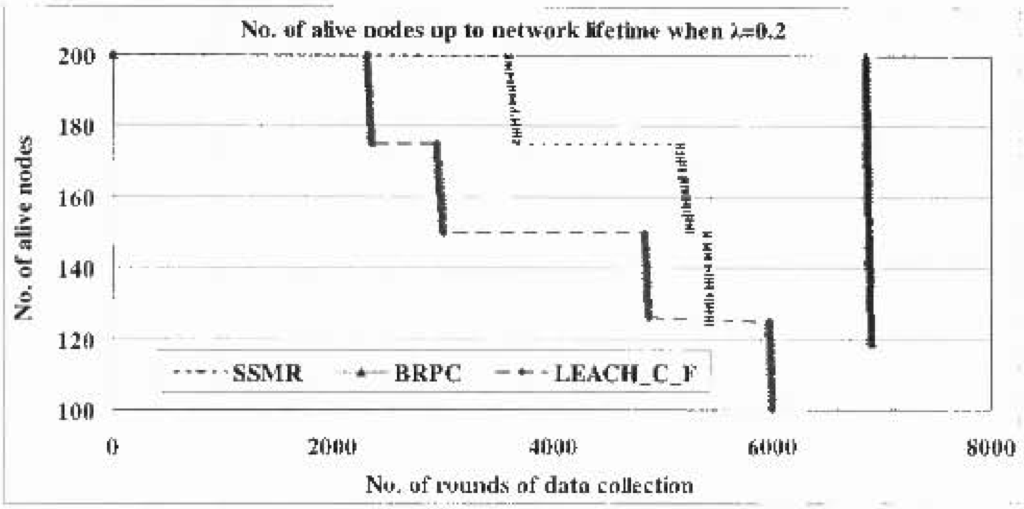


Figure 4-31. Number of live nodes up to network lifetime when λ=0.2.

Both figures show that BRPC has a longer first-node lifetime and network lifetime than LEACH_C_F and SSMR. The improvements become more significant when λ=0.6 than when λ=0.2. These two figures also show that, by rotating the CH role, the higher power consumption of CHs is well distributed among the nodes in the cluster in all algorithms, as almost all nodes in a cluster die simultaneously. BRPC further balances power consumption among the clusters as almost all nodes in the network die simultaneously. The expected lifetime of a clustered network as described in the strategy in Section 4.1 is therefore realized by BRPC design. On the other hand, the directional data traffic makes the clusters in LEACH_C_F and SSMR have significant difference in their lifetimes.

7. Details of fraction of energy not yet used up to network lifetime

The fraction of energy not yet used up to network lifetime is shown Fig. 4-32 when λ=0.6 and Fig. 4-33 when λ=0.2. The curve of this fraction terminates at the network lifetime of each respective algorithm. This residual energy cannot be used any longer so is referred to as wasted energy. It can be seen that BRPC has smaller fraction of energy not yet used at network lifetime. Therefore, BRPC performs better in using the network energy than do LEACH_C_F and SSMR.

The comparison of the fraction at the same number of the rounds before the first-node lifetime of all algorithms shows the energy efficiency of the network. For example, at 800 rounds when λ=0.6, this fraction is 67%, 80% and 83% in LEACH_C_F, SSMR and BRPC respectively, which means that BRPC consumes less energy than LEACH_C_F and

SSMR. BRPC therefore shows higher energy efficiency than other two algorithms by maintaining the CH in the central area of the cluster.

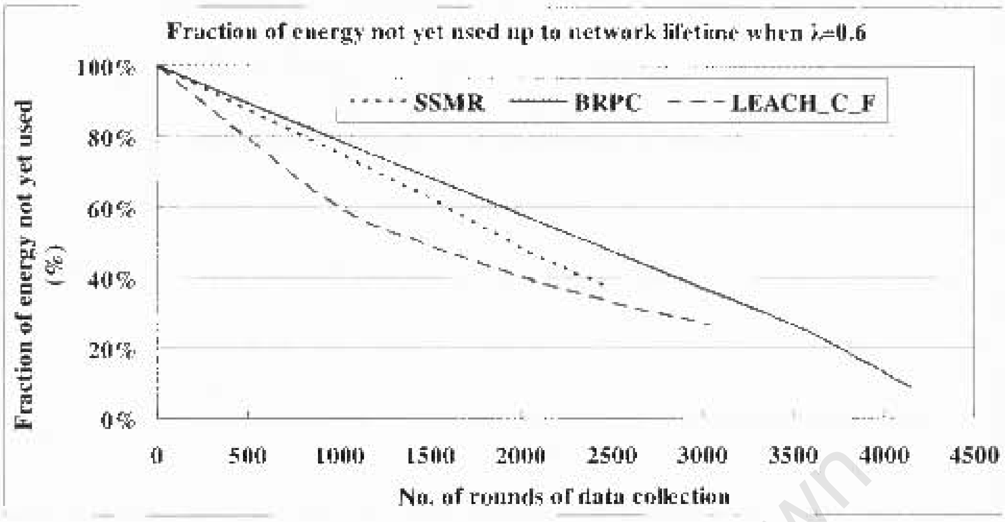


Figure 4-32. Fraction of energy not yet used up to network lifetime when λ=0.6.

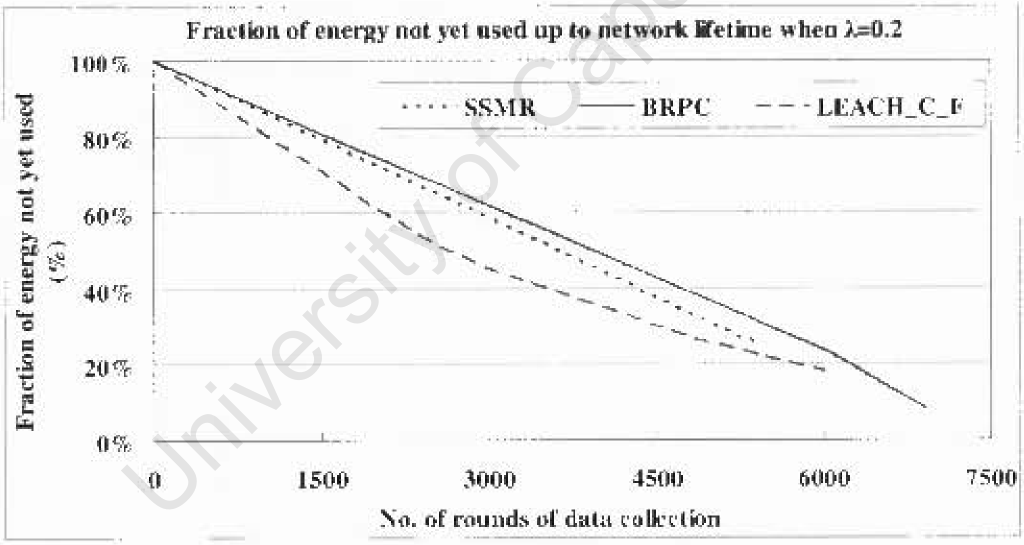


Figure 4-33. Fraction of energy that not yet used up to network lifetime when λ=0.2.

The absolute value of the slope of the fraction curve shows the energy efficiency of the network. A sharp curve means a lower energy efficiency of the algorithm. In a network with steady power consumption, the slope of the curve will be a constant. However, this slope is changed at different numbers of rounds in all algorithms.

BRPC separates the cluster lifetime into two periods. *Period1* has higher energy efficiency than does *period2*. The energy efficiency of *period2* therefore decreases and the absolute value of the slope becomes higher.

In SSMR, the cluster that is nearest the sink dies first. The premature cluster death reduces the power consumption as no energy is consumed any longer in this cluster. However, after such a cluster dies, other clusters that still have residual energy need to increase their transmission power to reach the sink, thereby increasing the power consumption. When only one cluster dies, the decrease in power consumption caused by premature cluster death is less significant than the increase in power consumption caused by the transmission power increase in other clusters, so that the absolute value of the slope of the fraction curve decreases. When more clusters die, the absolute value of the slope of the fraction curve increases as the decrease in power consumption caused by premature cluster death becomes more significant.

In LEACH_C_F, this absolute value of slope decreases after the first-node lifetime. When the clusters die prematurely, they will no longer consume energy. Therefore, less overall energy is consumed.

8. Details of packets delivered to the sink up to network lifetime

The total packets delivered to the sink at different rounds up to network lifetime are shown in Fig. 4-34 when λ=0.6 and Fig. 4-35 when λ=0.2.

BRPC delivers more packets than LEACH_C_F and SSMR. The rate of delivering packets (the slope of the curve in the figure) in BRPC is almost a constant. These rates in LEACH_C_F and SSMR, however, decrease after their first-node lifetime, because the prematurely dead clusters cannot deliver data to the sink any longer. Though LEACH_C_F and SSMR deliver similar network data up to network lifetime, the rate of delivering packets in LEACH_C_F becomes slower than SSMR after some specific number of rounds. This result again shows that LEACH_C_F delivers more partial-network data then SSMR.

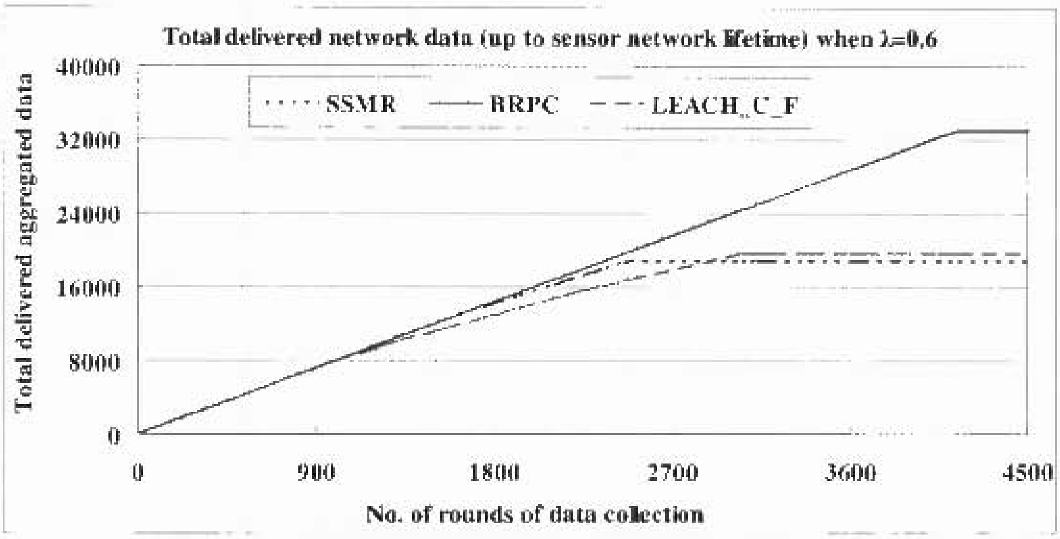


Figure 4-34. Total delivered network data up to WSN lifetime when $\lambda=0.6$.

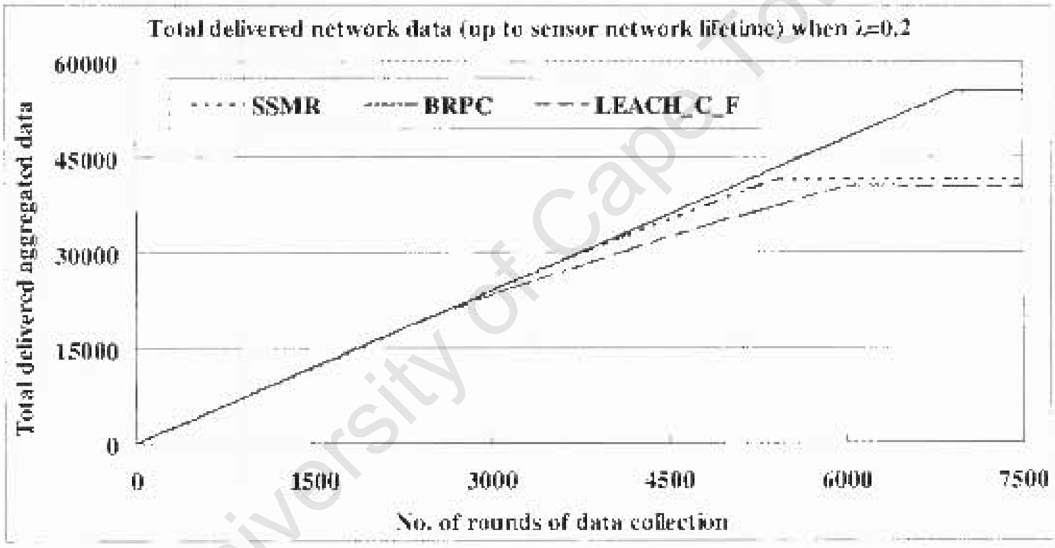


Figure 4-35. Total delivered network data up to WSN lifetime when $\lambda=0.2$.

9. Impacts of transmitter-amplifier parameter, e_{amp}

This subsection investigates the impacts of transmitter-amplifier parameter, e_{amp}, when $\lambda=0.4$. The first-node lifetime, the network lifetime, and aggregated data packets delivered to the sink are selected to assess this impact.

The first-node lifetime of all algorithms decreases when e_{amp} increases, as shown in Fig. 4-36. BRPC shows longer first-node lifetime than LEACH_C_F and SSMR. The impact of e_{amp}, however, is different for LEACH_C_F and SSMR. When e_{amp}=10pJ/bit/m^2, the first-node lifetime in LEACH_C_F is longer than in SSMR whereas when $e_{amp} \geqslant 50$pJ/bit/m^2, it reverses.

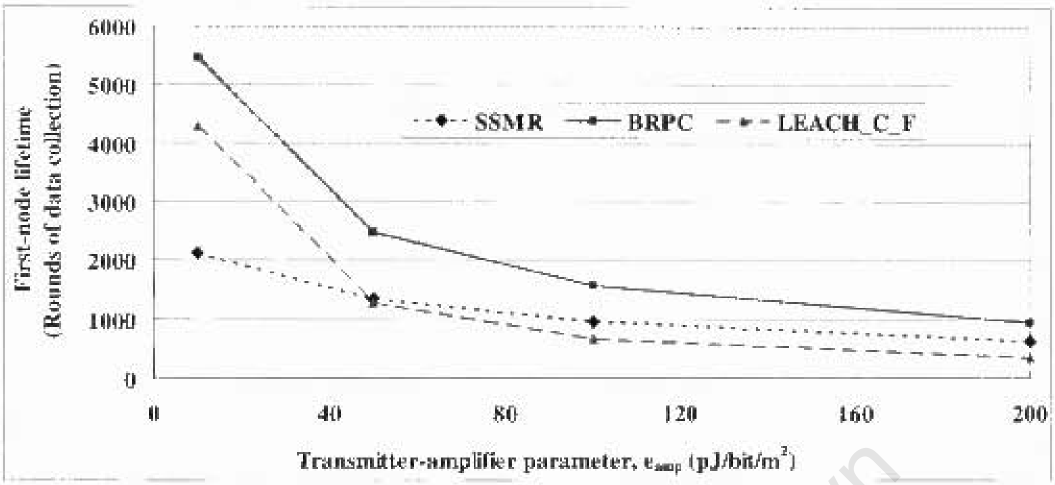


Figure 4-36. Fist-node lifetime at different e_{amp} values when λ=0.4.

The improvement in network lifetime by BRPC is not significant as shown in Fig. 4-37. When e_{amp}= 10pJ/bit/m^2, BRPC has even a shorter network lifetime than LEACH_C_F. It has been discussed in Section 4.2.2 that a two-hop route is more energy efficient when the distance between the source and the destination exceeds 100m. To achieve energy saving by separating a route of one hop into two hops, this distance between source and destination becomes longer when e_{amp}=10pJ/bit/m^2 than a greater e_{amp}. Therefore, a single hop route is more energy efficient than a multi-hop route in most scenarios of this simulation when e_{amp}=10pJ/bit/m^2. When the network size increases, BRPC will present higher energy efficiency than LEACH_C_F. Furthermore, the longer network lifetime does not necessarily mean that LEACH_C_F performs better than BRPC as some areas in LEACH_C_F are out of coverage earlier due to the premature cluster death.

The difference between the first-node lifetime and the network lifetime in these three algorithms is shown in Fig. 4-38. The difference in BRPC is extremely small whereas in LEACH_C_F and SSMR is quite significant. This means that many aggregated data packets delivered to the sink in LEACH_C_F and SSMR are from the partial network, thereby degrading network performance.

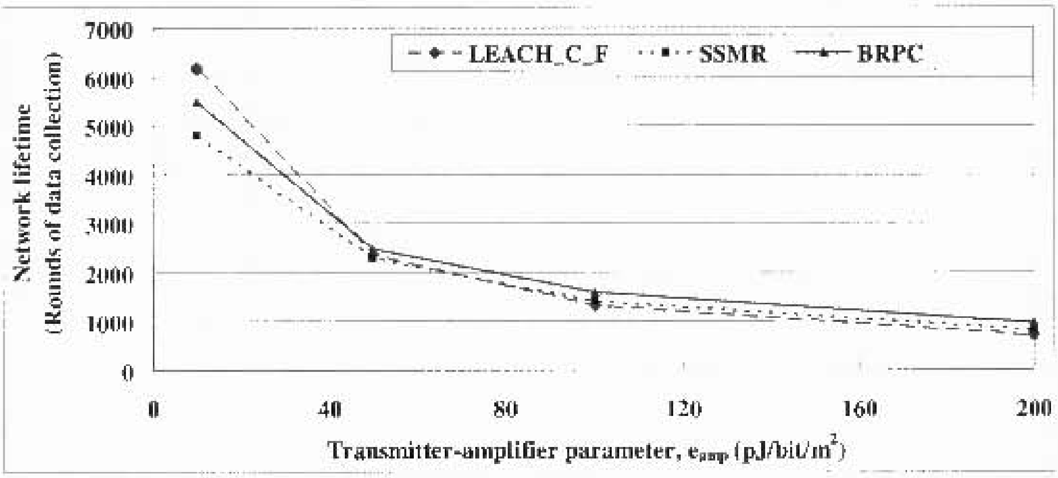


Figure 4-37. Network lifetime at different e_{amp} values when λ=0.4.

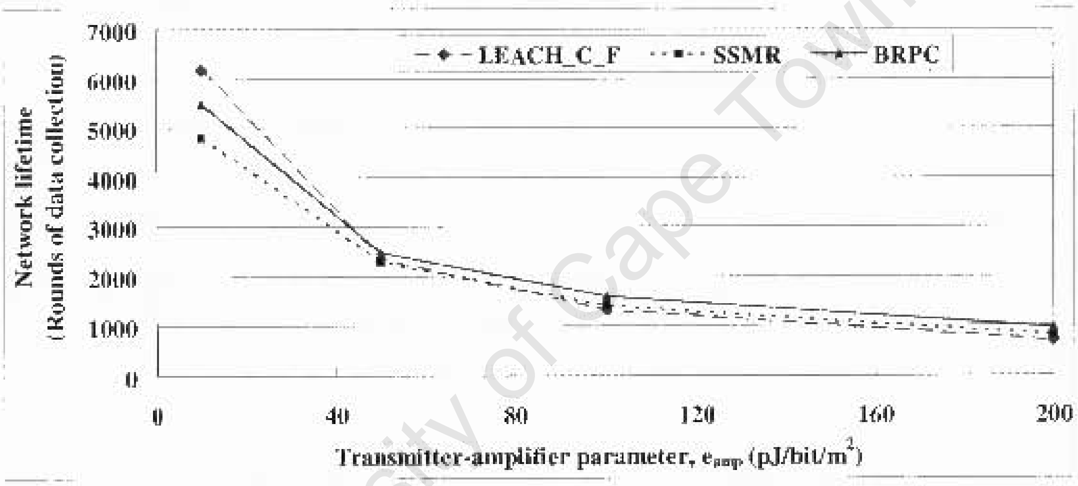


Figure 4-38. Difference between the first-node lifetime and network lifetimes when λ=0.4.

The total amount of network data delivered to the sink is shown in Fig. 4-39. BRPC delivers more network data than LEACH_C_F and SSMR when $e_{amp} \geqslant$ 50pJ/bit/m^2. When e_{amp} = 10pJ/bit/m^2, however, LEACH_C_F performs better in delivering network data than BRPC, which is consistent with the network lifetime (see Fig. 4-37). However, most data of BRPC are from the entire network whereas quite significant data of LEACH_C_F and SSMR are from the partial network, as shown in Fig. 4-40. From this point, BRPC still performs better that LEACH_C_F. From Fig. 4-40, SSMR performs better in sustaining the increase of e_{amp} than does LEACH_C_F. This is because when e_{amp} increases, a multi-hop route is more likely to have higher energy efficiency than a single-hop route.

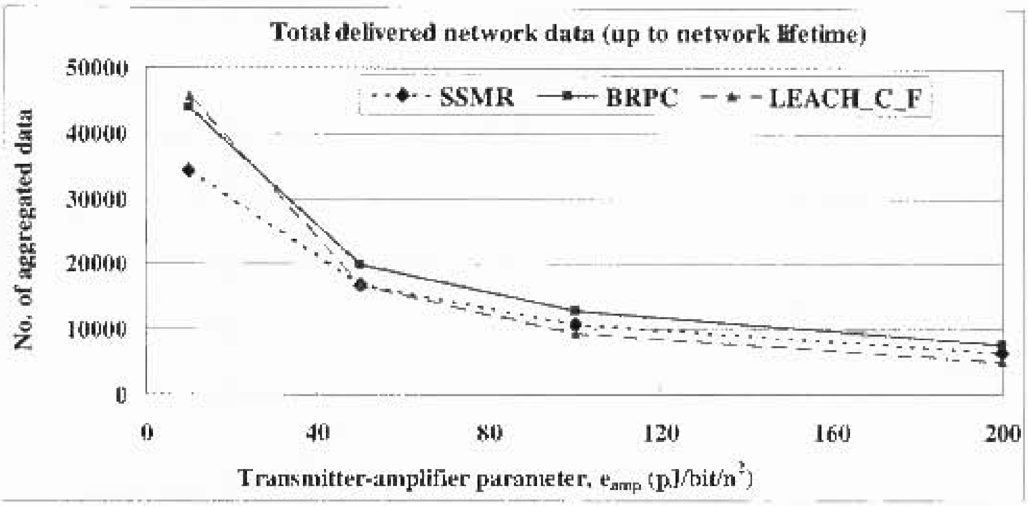


Figure 4-39. Total delivered network data up to network lifetime at different e_{amp} values.

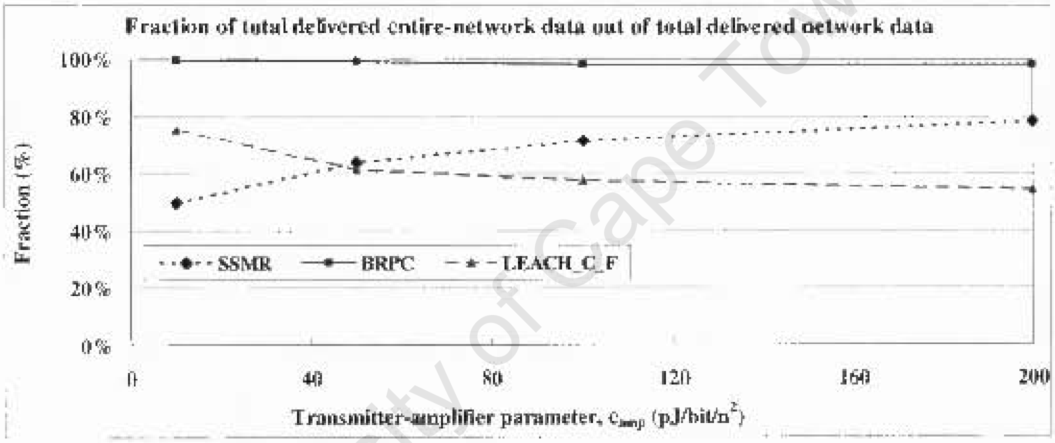


Figure 4-40. Fraction of entire-network data out of network data.

4.3.4 Inferences from the results

From the simulation results, the following inferences can be achieved:

The characteristic of directional data traffic towards the data sink in WSNs degrades the energy efficiency greatly, as well as the network performance because it results in premature node death. Clustering algorithm design by taking this characteristic into account greatly improves energy efficiency and network performance of WSNs.

Optimizing the CH selection by taking into account the location of the nodes in the cluster improves energy efficiency of WSNs.

The improvement by BRPC becomes more significant when the data aggregation rate or the transmitter-amplifier parameter increases. BRPC is therefore more suitable for the application in which the data that the sensor generated have low correlation or the nodes have high transmitter-amplifier parameter.

4.4 Summary and discussions

4.4.1 Summary

A clustering algorithm, BRPC, for WSNs has been proposed in this chapter. This BRPC contributes to balancing power consumption throughout the network by considering the directional data traffic towards the sink. It also improves energy efficiency by maintaining the CH in the central area of the cluster through re-clustering without the ripple effect.

The simulation results show that BRPC not only extends the lifetime of a WSN, but also improves the network performance. First-node lifetime and network lifetime have been used to evaluate the performance of energy efficiency of the algorithms. BRPC improves first-node lifetime significantly, as it avoids premature node death by balancing power consumption throughout the network. BRPC also improves network lifetime so that the network can run for a longer period. Entire-network data and partial-network data are used to evaluate performance of delivering data. BRPC delivers more entire-network data than do the comparative algorithms. This improvement is significant for realistic applications, which require high quality of service and need the information from the entire network. Besides delivering more entire-network data, BRPC also delivers more network data up to network lifetime. Therefore, it improves both energy efficiency and network performance.

However, similar to the algorithms in [13, 34, 40, 42, 43], the simulation was conducted by implementing BRPC into MATLAB and it only evaluated the energy related issues. Other issues like link quality, quality of service, security were not evaluated. Further research will improve the design of the algorithm and conduct more comprehensive performance evaluation.

4.4.2 Discussions

1. Extension to WSNs with different nodes and uneven node distribution

BRPC assumes a homogeneous WSN with the identical sensors uniformly distributed within the network. During a real-life application, however, the sensor nodes in a network may not be the same and they may not be uniformly distributed in the network.

This is to discuss the extension of the algorithm to WSNs with different sensor nodes and uneven node distribution. When such a network starts to work, the sensors send the location and energy information to the sink, which will then have an energy density map and a node density map of the network. In the assumption of BRPC, both the node density n_d and the energy density n_{energy} are constants so that energy $E_{cluster}$ in a cluster can be expressed by (4.44), in which $S_{cluster}$ is the cluster area.

$$E_{cluster} = S_{cluster} \times n_d \times n_{energy} \tag{4.44}$$

In a network with different nodes and uneven node distribution, the node density $\alpha(n_d)$ and the energy density $\delta(n_{energy})$ become functions of the location. The energy $E_{cluster}$ can be expressed by (4.45).

$$E_{cluster} = \int_S \alpha(n_d) \times \delta(n_{energy}) dS \tag{4.45}$$

The total power consumption of a cluster $P_{cluster}$ in a homogeneous WSN with uniform node distribution can be expressed by (4.46):

$$\begin{aligned} P_{cluster} &= P_{Intra-cluster} + P_{Agg} + P_{Inter-cluster} \\ &= a \times (S_{cluster} \times n_d - 1) \times \left(e_r + e_t + e_{amp} d^n\right) + a \times (S_{cluster} \times n_d) \times e_{Agg} \\ &+ i \times e_r + (\lambda \times a \times (S_{cluster} \times n_d) + i) \times \left(e_t + e_{amp} D^n\right) \end{aligned} \tag{4.46}$$

The meaning of each parameter in (4.46) can be found in Section 4.2.4.

This total power consumption of a cluster $P_{cluster}$ in a WSN with uneven node distribution can be expressed by (4.47):

$$\begin{aligned} P_{cluster} &= P_{Intra-cluster} + P_{Agg} + P_{Inter-cluster} \\ &= a\times(e_r + e_t + e_{amp}d^n)\times(\int_S \alpha(n_d)dS - 1) + a\times\times e_{Agg}\times\int_S \alpha(n_d)dS \\ &+ i\times e_r + (\lambda\times a\times\left(\int_S \alpha(n_d)dS\right) + i)\times\left(e_t + e_{amp}D^n\right) \end{aligned} \tag{4.47}$$

Replacing (4.44) by (4.45) as the energy in a cluster and replacing (4.46) by (4.47) as the total power consumption of the cluster; the proposed clustering algorithm, BRPC, can be extended to WSNs with different nodes and uneven node distribution.

2. Extension to WSNs with high node density

In BRPC, it is assumed that all nodes need to transmit data to the CH during each round. Sometimes, however, the node density in the network is high and one area may be overlapped by several nodes. Therefore, not all nodes need to generate and send data to the CH during each round. A clustering algorithm for such networks with high node density is to be presented in Chapter 5.

5 Balancing power consumption (BPC) by joint clustering and scheduling in high-node-density homogeneous WSNs

5.1 Strategies to improve energy efficiency in high-node-density clustered WSNs

Chapter 4 proposed BRPC to balance power consumption and to improve energy efficiency for clustered WSNs. BRPC requires that each node sends its data to the respective CH during each round of data collection. However, in a WSN with high node density, it is unnecessary to require all nodes to send data to the CH during each round.

In a WSN with high node density, an area that could, almost, be monitored by only one node may in fact have several sensor nodes, as shown in Fig. 5-1. It is therefore unnecessary to require that all nodes in this area be active at any given time [97, 119, 123]. Furthermore, having all these nodes in continuously active could be disadvantageous. First, though the data generated by these nodes can be locally aggregated before being sent to the sink, the source node may still waste significant energy in transmitting these data to the next hop. Secondly, the large number of nodes participating in relaying data makes the network topology more complicated. This complicated topology increases flooding in a network that uses a reactive routing protocol or overhead in a network that uses a reactive routing protocol. Thirdly, the resultant high-density data traffic increases the collisions and congestion, thereby increasing loss of data packets.

For energy efficiency to be improved in such a WSN with high node density, only some nodes are allowed to actively monitor the network and to relay data for others. Others, meanwhile, are put into a sleep state with negligible power consumption [124]. Scheduling schemes have been proposed for this purpose. However, the purpose of most traditional scheduling schemes is to reduce power consumption of the nodes when they are in an idle state or to avoid collisions when some nodes want to use the same channel simultaneously.

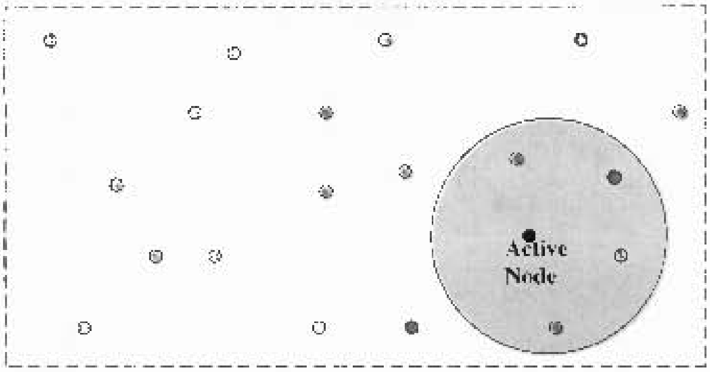


Figure 5-1. The colored cluster (area) is covered by 5 duplicate nodes.

The traditional scheduling algorithms have two disadvantages. First, the active nodes may not cover the entire network, as the nodes usually become active randomly. Secondly, these algorithms usually do not consider the directional data traffic in WSNs, thereby resulting in the nodes nearer the sink dying prematurely as they need to relay more data.

Joint clustering and scheduling in such a network has some advantages. First, in a WSN with high node density, the nodes can be clustered in such a way that one active node in a cluster can almost cover the entire cluster. This means that almost the entire network will be covered by the active nodes. This active node is referred to as the CH of that cluster to make it consistent with Chapter 4. Secondly, in traditional scheduling schemes, each node in the network can be active randomly when it is ready to send data, making balance of power consumption throughout the network infeasible. However, by joint clustering and scheduling, it is possible to allocate the energy to each cluster in accordance with its burden, making the balance of power consumption throughout the entire network feasible.

BRPC in Chapter 4 contributed to balancing power consumption throughout the network and further improving energy efficiency by maintaining the CHs in the central area of the cluster. As stated, however, the active node in a cluster in a high-node-density WSN may not cover the entire cluster. If the cluster lifetime in such a network is also separated into two periods and the CH is maintained in the central area of the cluster as BRPC (refer to Section 4.2.2 and Section 4,2.5), some edge area of the cluster may be out of coverage within the whole *period1*. Thus, to guarantee that the area that is not covered in this round be monitored in the following rounds, the CHs are required to be rotated within the entire cluster.

This chapter therefore proposes an algorithm which joins clustering and scheduling to balance power consumption (BPC) throughout the network for clustered WSNs with high node density. In such a network, only the CH monitors the cluster, while other nodes are in sleep mode. The nodes are clustered in such a way that the energy store in the cluster is proportional to its power consumption. The cluster lifetimes are then equalized. Furthermore, BPC rotates the role of CH frequently according to the residual energy of the node to better balance power consumption among the nodes in the clusters and to make the cluster be better covered.

Following the investigation of the existing scheduling algorithms, the proposed algorithm, BPC, is described. The performance of BPC is then evaluated by comparison with the other existing algorithms.

5.2 Review of the existing scheduling schemes

This subsection first investigates the existing scheduling algorithms. Following this investigation, the analysis of these algorithms is provided.

5.2.1 Investigation into existing scheduling algorithms

Typically, the proposed scheduling schemes can be classified into two major categories: energy saving schemes of the Medium Access Control layer (MAC schemes), and the topology-configuration scheduling schemes [119, 124, 125].

1. MAC schemes

MAC schemes for WSNs can be further divided into contention-based MAC schemes and non-contention-based MAC schemes [125].

Power Aware Multi-Access protocol with Signaling (PAMAS) [126] has a two-channel architecture. It allows a node to sleep to avoid over-hearing a packet intended for a different destination or to avoid interfering with another node's reception as it is not transmitting. However, PAMAS ignores the issue of idle listening.

Sensor-MAC [127, 128] is a single-channel protocol. It applies a simple scheduling scheme to allow neighbors to sleep for long periods and to synchronize activation. However the nodes in Sensor-MAC need sleep schedules to be synchronized with those of surrounding neighbors. This is realized by their schedules being periodically broadcast. This periodic broadcasting increases flooding. Based on the communication of neighbors, T-MAC [129] improves Sensor-MAC by adjusting the length of time that sensors are awake between sleep intervals.

More recently, Chowdhury [130] has proposed a CMAC scheme to allow for listening to and replying controlling messages even when the reception of a data packet is in progress at a sensor node. This saves more energy.

The contention-based MAC schemes improve energy efficiency of WSNs. However, the power consumption of collision, over-hearing, and idle listening in these schemes is still significant. TDMA protocols have been therefore proposed to improve energy efficiency. Preamble-based TDMA protocols are collision-free by scheduling each node when it should be active and sleep.

TRaffic-Adaptive Medium Access protocol (TRAMA) [131] employs a traffic distributed election scheme that selects receivers based on schedules announced by transmitters. The nodes in the network will periodically awake to exchange broadcasts and to learn their two-hop neighborhood. Based on this acquired knowledge, the nodes will periodically reserve future slots for backlogged traffic. A hash-based priority scheme is then used so that only one node in a two-hop neighborhood will transmit in a given slot. Miller [132] improves TRAMA by scheduling long-lived, end-to-end, and periodic flows instead of scheduling recently received packets based on hop by hop. Furthermore, this scheme does not need to maintain consistent two-hop neighborhood information, which therefore results in a simpler scheduling algorithm. Energy and rate based MAC protocol [133] is similar to TRAMA. This protocol divides a network into TDMA groups based on the neighborhood information. Using the critical energy of a node and the traffic flow rates of its neighbors, it can adaptively allocate the appropriate timeslot to each node to balance power consumption.

In Sichitiu's scheme [134], the nodes maintain a table of times when they should wake up to send and to receive data. When a sender intends to schedule a new flow on a link and if no data transmission is sensed in that slot, it will send an RSETUP packet. If the RSETUP packet is received successfully by the receiver, an ACK will be sent back to the sender and the slot is scheduled. If no ACK is sent back to the sender, the sender needs to re-send in a different slot.

2. Topology-configuration scheduling schemes

Geographical Adaptive Fidelity (GAF) algorithm [122] uses geographical information to prolong the lifetime of the network. The main idea of GAF is to partition the network into virtual square clusters. The clusters are structured such that any two nodes within neighboring clusters can communicate with each other. Energy is saved by scheduling only one node to be active in any cluster and by placing the others in its cluster into a sleep state. The algorithm by Koushanfar [135] forms a backbone for the network. Only the nodes in the backbone are active while other nodes are put into a sleep state. The nodes take turns to form part of the backbone to distribute power consumption. Different from GAF and Koushanfar's algorithm, the nodes in Adaptive Self-Configuring WSNs Topologies (ASCENT) [136] decide to become active based on both neighborhood links and the observed data loss rates. This provides the network with the ability to trade energy consumption for communication reliability. Similar to ASCENT, SPAN [137] adaptively elects active nodes according to the number of neighbors, but SPAN also considers the residual energy of the node. The node in SPAN takes the responsibility for making an independent decision as to whether it should become active or not. This decision is based on its residual energy and its benefit to the neighboring nodes of having their communication bridged. The active and the sleeping nodes will periodically check their states to decide whether they should take a new role: sleeping or becoming active.

Xu [138] proposes two distributed algorithms to manage the channels of the nodes: the Basic Energy-Conserving Algorithm (BECA) and the Adaptive Fidelity Energy-

Conserving Algorithm (AFECA). The basic idea of BECA is to put the nodes into sleep state when they are not participating in sending, forwarding, or receiving data. BECA defines three possible states for a node in terms of its radio components: *sleeping*, *listening*, and *active*, as shown in Fig. 5-2. Each node has an initial sleep period of time *Ts*. However, during this sleep period, if the node has data to send, it will become active. If it does not have data to send, it will change to a listening state after the sleep period *Ts*. The listening period will be *Tl*. If it has data to send during this period, it will change to an active state. If not, it will return to a sleep state after *Tl*. The active period is defined as *Ta*. When a node becomes active, if it does not send or receive data after *Ta*, it will return to the sleep state. AFECA is built on BECA but it adjusts the sleep time of a node based on node density.

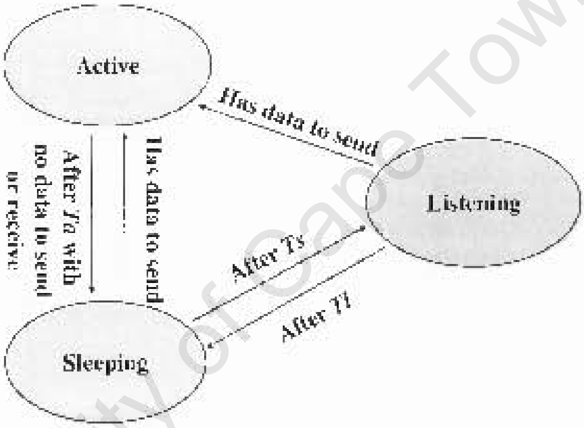


Figure 5-2. Basic energy-conserving algorithm (Adapted from [138]).

Sparse Topology and Energy Management (STEM) [139] has a two-channel architecture. The primary channel is for sending data and the secondary one is for awaking the nodes. Energy saving is achieved by keeping the data radio components sleep until communication is desired. When a node needs to send a packet, it pages the next node in the routing path. This node then turns on its main radio so that it can receive the packet. A node with data to send begins transmitting continuously on the wake up channel long enough to guarantee that all neighbors receive the wake up signal. RATE-EST [140] is similar to STEM, but it achieves greater energy saving by periodically listening on the primary channel and buffering packets.

5.2.2 Analysis of the existing scheduling algorithms

Analysis of the existing algorithms

The existing scheduling algorithms allow the nodes in the network to sleep with negligible power consumption when they do not have data to send or relay. This saves significant energy in the network. As stated in Section 5.1, however, these algorithms may fail to optimize energy efficiency as the directional data traffic is not considered. Furthermore, the active nodes in the network may not cover the entire network. An algorithm that combines clustering and scheduling techniques is expected to addresses these two issues effectively.

Related work

GAF [122] combines the clustering and scheduling techniques to improve energy efficiency. GAF, however, is proposed for MANETs so that the directional data traffic towards the sink is not taken into account. It therefore fails to optimize energy efficiency in WSNs.

Deng [123, 141] proposes an algorithm which applies a scheduling scheme into a clustered WSN. This algorithm organizes the nodes into one-hop clusters. It considers that the nodes that are further away from the CH will have higher burden as they need higher transmission power to reach the CH. By scheduling these nodes into sleep state more frequently than others, this higher burden can be compensated, therefore equalizing the node lifetime in the same cluster. However, Deng's algorithm only balances power consumption among the nodes in the cluster and fails to balance power consumption throughout the network.

5.3 Proposed clustering algorithm-BPC

5.3.1 Network model and energy model

The network model in this chapter is similar to the one in Chapter 4 except that the model in this chapter caters for a high density of nodes. Energy-constrained identical sensors with limited transmission power are uniformly distributed in a WSN. These sensors are aware of their location information, so that they can be geographically grouped into clusters by the sink which has sufficient resources and is located near the network. Each cluster only requires the CH to be active. The role of active CHs is rotated among the nodes in the cluster.

The energy model of BPC is the same as BRPC in Chapter 4 (Section 4.2.1), which is shown in (4.1), and the following parameter values are applied: for distance $d<d_{crossover}$ (where $d_{crossover}$=87m), e_{amp}=10pJ/bit/m^2 and n=2; for distance $d \geq d_{crossover}$, e_{amp}=0.0013pJ/bit/m^4 and n=4; and e_t=e_r=50 nJ/bit.

5.3.2 Cluster organization and contention-based scheduling algorithm

The organization of clusters is described here and summarized in Fig. 5-3. Similar to BRPC in Chapter 4, BPC is distributed yet centralized at start-up to better form the clusters. The clusters are fixed, yet re-clustering (with resultant ripple effect) will take place when original nodes leave or new nodes join the network. The network lifetime is divided into rounds and each round is divided into a set-up phase and a data communication phase.

When the network commences, the sink collects the location information of the nodes and then divides the network into geographically-based clusters using a centralized scheme. The clusters are organized in such a way that the burden of each cluster is relative to its energy store to make the clusters have similar lifetimes. This is analyzed in detail in Section 5.3.4. After determining the cluster divisions, the sink broadcasts the division information, which includes location information, cluster ID and the number of nodes in each cluster, to each node.

On reception of the division information from the sink, a node can determine which cluster it belongs to. Through exchange of the information of the cluster ID that each node receives from the sink, the relationship of cluster-mates of the nodes in the same cluster is developed. Thus, each node can maintain the information of all nodes in the same cluster. The next step is to elect a CH in each cluster to monitor that cluster and to relay data from other clusters. The selection of the CH is based on the residual node energy. However, should the network commence when all nodes have the same residual energy, the node at the centre of the cluster is selected to be CH.

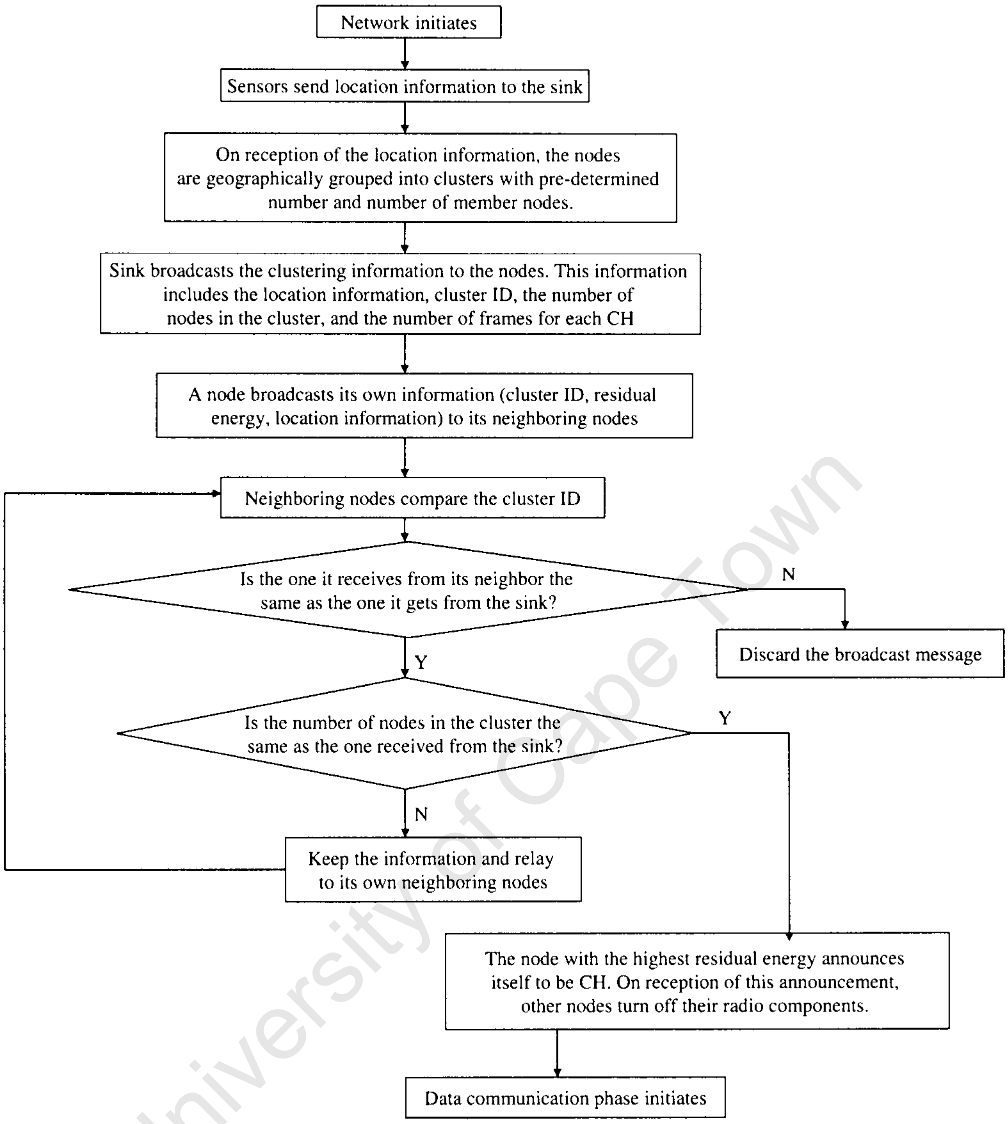
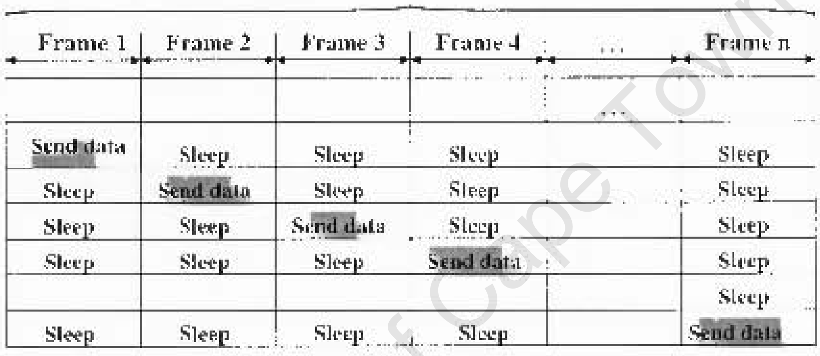


Figure 5-3. Flow chart of cluster organization of BPC.

The scheduling scheme used for the competition of being a CH is contention-based. To reduce the flooding caused by exchanging residual energy information, only the CH will broadcast its residual energy information to the member nodes when each round expires. This is because, only the CH consumes energy during each round (other nodes are in sleep state with negligible power consumption) and thus only its residual energy information needs to be updated. Each time residual energy information is updated; the node with the highest residual energy becomes a CH for the next round. The sink assigns the CHs a number of frames. This number is pre-set according to the requirement of the network. If a network needs high quality of service and needs all information carried by the entire network, the active CH needs to be rotated frequently. This frequent rotation, however, may consume some energy, increases topology overhead and flooding in routing discovery. Adjusting the rotation period, the tradeoff between the QoS and the power consumption can be achieved.

The TDMA scheduling scheme in BRPC (proposed in Chapter 4) for each CH, as shown in Fig. 5-4, allows each member nodes to send data to the CH in their own frames (refer to Section 4.2.3).

TDMA scheme for each CH to allow the member nodes to send data in BRPC



Frame 1	Frame 2	Frame 3	Frame 4	...	Frame n
				...	
Send data	Sleep	Sleep	Sleep		Sleep
Sleep	Send data	Sleep	Sleep		Sleep
Sleep	Sleep	Send data	Sleep		Sleep
Sleep	Sleep	Sleep	Send data		Sleep
					Sleep
Sleep	Sleep	Sleep	Sleep		Send data

Figure 5-4. TDMA scheme for the CHs in BRPC (Proposed in Chapter 4).

The scheduling scheme of BPC is shown in Fig. 5-5. At each round, the node with the highest residual energy in the cluster becomes the CH. After the CH is determined, the number of frames is assigned to each CH. A CH is allowed to send a data packet within each frame.

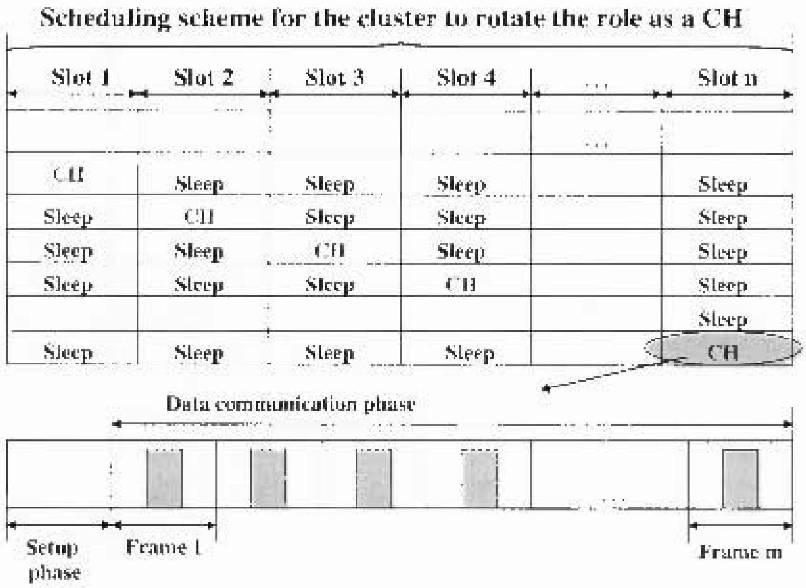


Figure 5-5. Scheduling scheme for BPC.

5.3.3 Data communication phase

The whole process of the data communication phase of BPC is shown in Fig. 5-6. When the number of frames is assigned to each CH, the CH can send data to the sink. BRPC requires a CH to aggregate the data from the member nodes before it sends the data to the sink (refer to Section 4.2.3). This is because the data from the member nodes correlate and thus the data aggregation process can greatly reduce the amount of data and thus power consumption of relaying them. The data generated by each CH in BPC, however, are from different clusters, which have virtually small/no correlation [35]. The data aggregation process cannot significantly reduce the data amount. BPC therefore does not require the CHs to aggregate data over hop by hop as done in previous research [35]. The routing discovery process in BPC is similar to that in BRPC, when a CH acquires the routing information, it selects the routes based on the consideration of residual node energy.

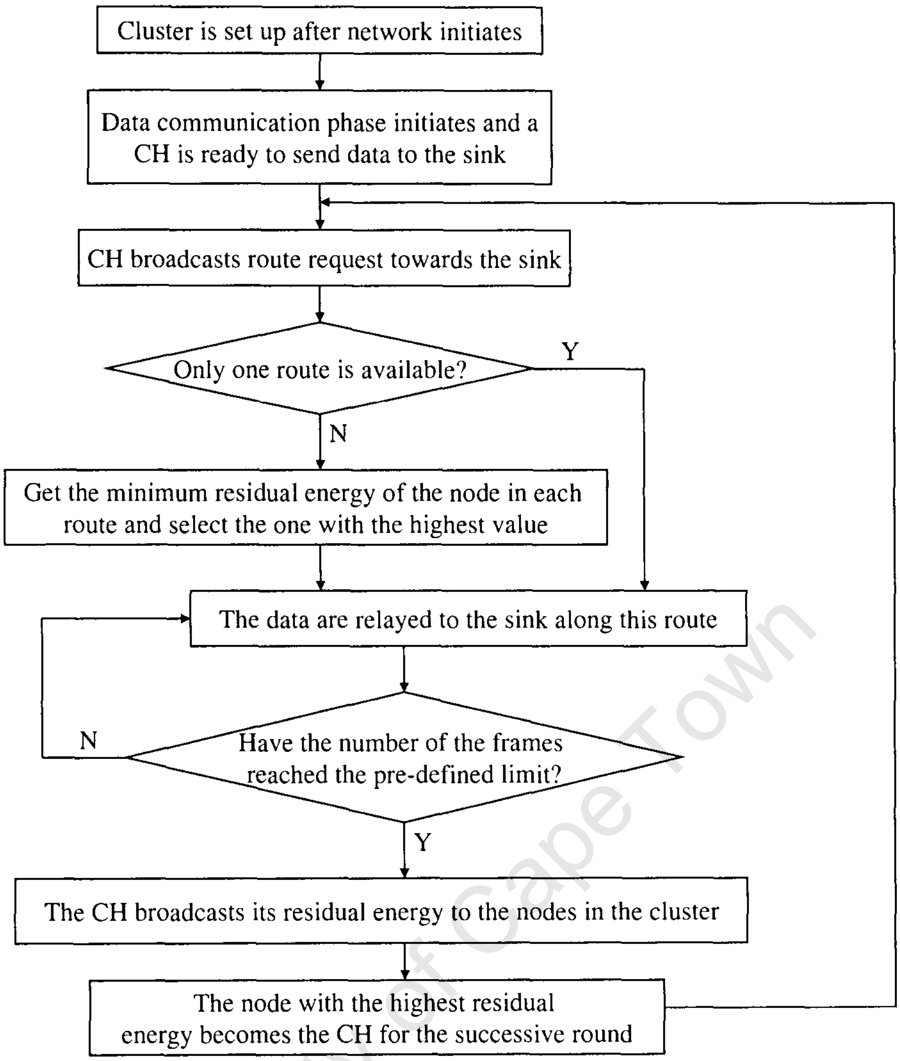


Figure 5-6. Flow chart of data communication phase of BPC.

5.3.4 Equalizing cluster lifetime

This subsection discusses equalizing cluster lifetime using the model shown in Fig. 5-7. This model has been used to explain equalization of cluster lifetimes in BRPC. The explanation of the parameters can be found in Section 4.2.4.

The clusters are named according to their geographic positions. The identical sensors with the initial energy of $E_{initial}$ are uniformly distributed in the network which has a node density of n_d. To simplify the analysis, it is assumed that each cluster selects the node at the centre of the cluster to be CH. For cluster a_ib_j, the transmission range of d_{ij} is assigned.

This transmission range, however, is adjustable in both realistic applications and simulations.

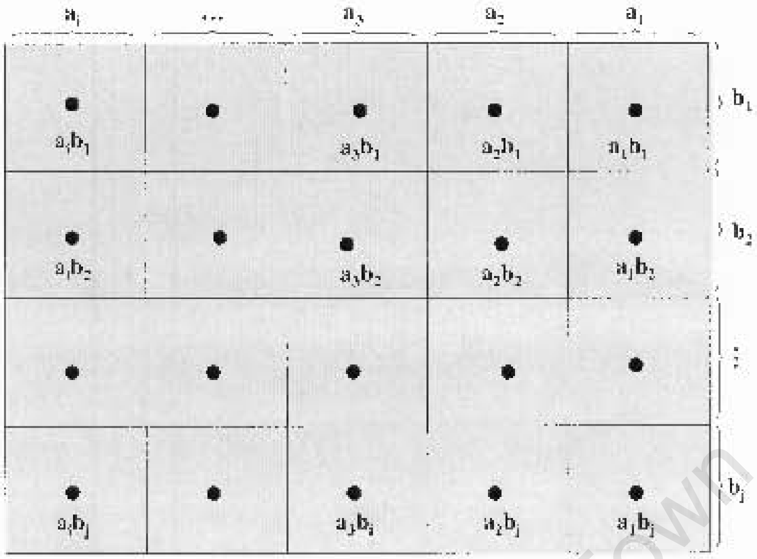


Figure 5-7. WSN to explain cluster-lifetime equalization in BPC.

To equalize the cluster lifetimes, (5.1) must be observed.

$$\frac{E_{total}(i)}{\sum P_{re}(i)+\sum P_{tr}(i)} \approx const \tag{5.1}$$

where $E_{total}(i)$ is the total energy store in cluster a_ib_j, and $\sum P_{re}(i)$ and $\sum P_{tr}(i)$ are respectively the total CH power consumption of receiving and transmitting data in cluster a_ib_j.

There are four neighboring clusters as shown in Fig. 5-8. Each CH generates an α-bit data message. It is assumed that the probability of a neighboring CH nearer the sink of cluster $a_{i-1}b_{j-1}$ to relay the route request is β. The total data of the CH in cluster a_ib_j ($Data_{a_ib_j}$) that need to be relayed is separated into the data that the CH receives from others ($R(a_ib_j)$) and the data that the CH transmits to the next hop ($T(a_ib_j)$), as shown in (5.2). The explanation of the parameters can be found in Section 4.2.4.

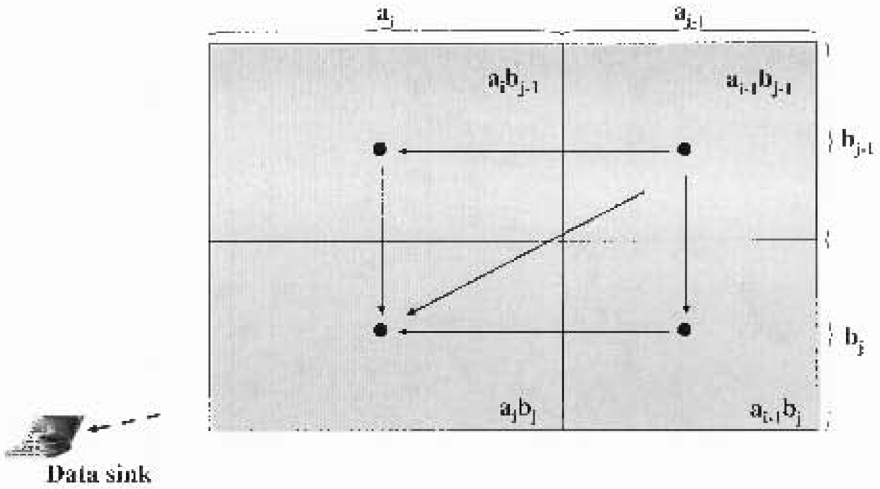


Figure 5-8. Different data traffic in neighboring clusters in BPC.

$$Data_{a_ib_j} = T(a_ib_j) + R(a_ib_j) \tag{5.2}$$

in which

$$\begin{aligned} R(a_ib_j) &= T(a_{i-1}b_j)\times\beta + T(a_ib_{j-1})\times\beta + T(a_{i-1}b_{j-1})\times\beta + 2\times T(a_{i-1}b_{j-1})\times\beta^2 \\ &= \sum_{(m>i-1)}\sum_{(n>j-1)}(i+j-m-n)\times a_m a_n \times\beta^{i+j-m-n} \end{aligned} \tag{5.3}$$

$$T(a_ib_j) = a + \sum_{(m>i-1)}\sum_{(n>j-1)}(i+j-m-n)\times a_m a_n \times\beta^{i+j-m-n} \tag{5.4}$$

The power consumption of the CH in cluster a_ib_j is therefore determined in (5.5).

$$P_{(a_ib_j)} = R(a_ib_j)\times e_r + T(a_ib_j)\times\left(e_t + e_{amp}\times R_{ij}^{\,2}\right) \tag{5.5}$$

The total initial energy store of cluster a_ib_j is (5.6):

$$E_{cluster} = a_i \times b_j \times n_d \times E_{initial} \tag{5.6}$$

By acquiring (5.7), the cluster lifetimes are equalized.

$$\frac{E_{cluster}(a_i b_j)}{P_{(a_i b_j)}} \approx const \tag{5.7}$$

5.4 Performance evaluation

To evaluate the performance of BPC for WSNs with high node density, a traditional clustering algorithm, which organizes clusters with similar sizes, such as in [14, 34, 36, 42], and sends data to the sink over multi-hop routes, such as in [34, 36], is used for comparison. To render these two algorithms comparable, the scheduling algorithm being implemented makes only the CH in the selected traditional clustering algorithm active. This algorithm, which is now similar to GAF, is referred to as Same Size with only CH Active (SSCA).

5.4.1 Simulation scenario

Homogeneous sensor nodes, 400 in total, are uniformly distributed within an area of 100m×50m, with the data sink located 10m away from the network, as shown in Fig. 5-9. Each node has 2J initial energy with a maximum transmission range of 70m. The operation of both algorithms is broken up into *rounds*. Each round assigns 20 frames to a CH and a CH sends a 500 bytes packet to the sink over a multi-hop route during each frame. The nodes are organized into 8 clusters.

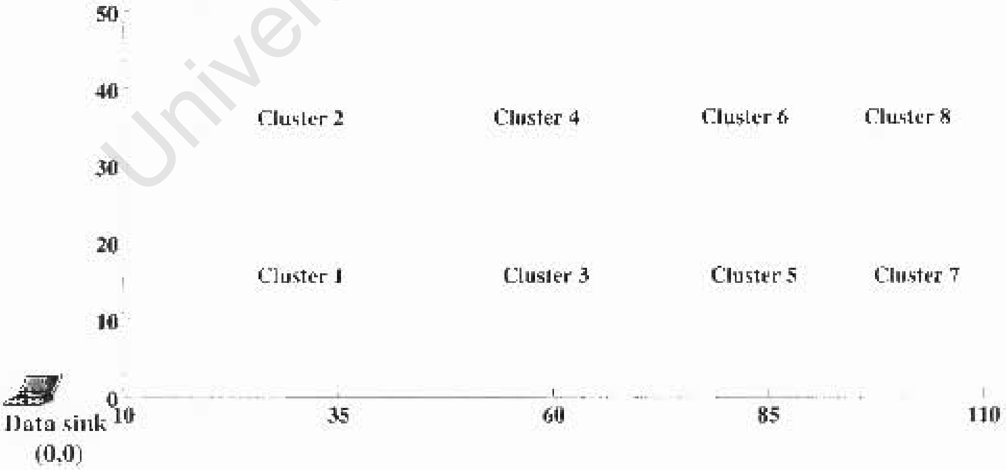


Figure 5-9. Simulation Scenario of BPC.

5.4.2 Assessments requirement

The performance evaluation for BRPC (refer to Section 4.3) only measured the first-node lifetime and the network lifetime. Besides measuring these two lifetimes, the performance evaluation of BPC provides more details about the residual node energy at different stages, as shown in Fig. 5-10.

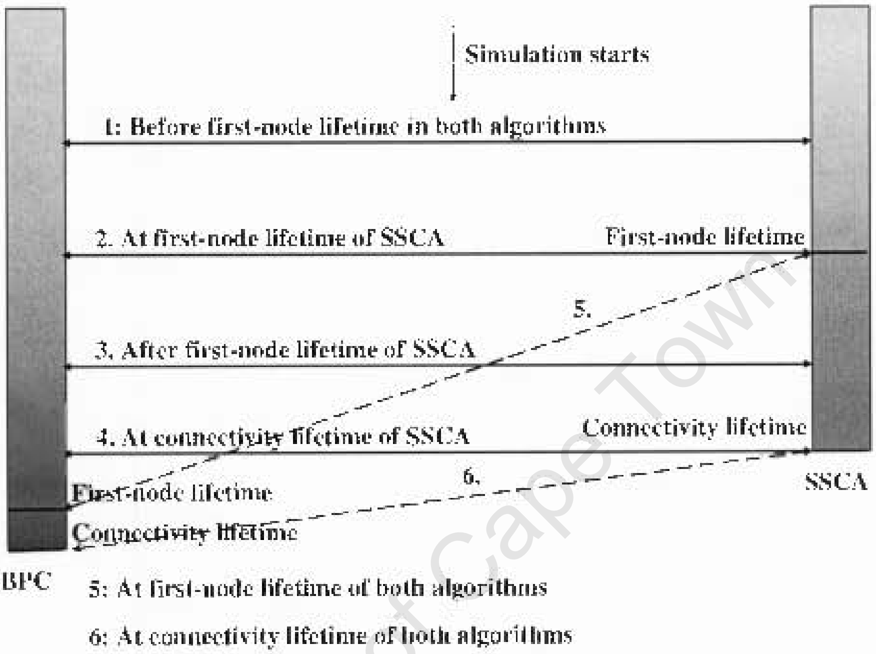


Figure 5-10. Comparisons to be made at different stages.

The network lifetime is separated into four stages:

i. *before* the first-node lifetime (the definition of first-node lifetime can be found in Section 4.3.2) of *both* algorithms,

ii. *at* the first first-node lifetime encountered in either algorithm (possibly SSCA),

iii. *after* this first-node lifetime and,

iv. *at* the first connectivity lifetime (the definition of connectivity lifetime can be found in Section 4.3.2) encountered in either algorithm (possibly SSCA).

The first stage (i) shows how much average energy remains in every node in different clusters. A similar average energy means a good balance of power consumption. The second stage (ii) is expected at the first-lifetime of SSCA. BPC balances power consumption throughout the network so that it is reasonable to expect that BPC has longer first-node lifetime than that of SSCA. After the first-node lifetime, the nodes with residual energy in SSCA keep relaying data to the sink. The third stage (iii) compares the average residual node energy to show how energy is consumed in the clusters that still have remaining energy. The fourth stage (iv) is expected at the connectivity lifetime of SSCA, as BPC should have a longer connectivity lifetime than does the SSCA.

The above four stages compare the average residual energy at the same number of rounds. The first-node lifetime and connectivity lifetime in BPC and SSCA, however, are different. It is therefore necessary to compare the respective lifetimes and the average residual node energy at their respective lifetimes in BPC and SSCA.

Besides the comparison of the residual energy at different stages, the following four components are also evaluated to more comprehensively assess the algorithms: (1) the number of live nodes up to connectivity lifetime, (2) the distribution of residual node energy at the connectivity lifetime, (3) the amount of energy remaining in the network up to connectivity lifetime, and (4) the data packets delivered to the sink up to connectivity lifetime.

5.4.3 Simulation results

Fig. 5-11 shows the average residual node energy at 900 rounds. Up to 900 rounds, SSCA consumes 157J energy whereas BPC consumes 153J energy. The consumed energy is similar. However, as SSCA assigns similar number of nodes to each cluster without the consideration of the directional data traffic, the cluster that is nearest the sink needs to relay more data from others and will drain energy at a more rapid rate. BPC, on the other hand, allocates the number of the nodes to each cluster according to the burden of the respective cluster. The clusters in BPC therefore drain energy at a similar rate, which is obvious from Fig. 5-11.

Fig. 5-12 shows the comparison at 1072 rounds, which is the first-node lifetime of SSCA. Up to 1072 rounds, SSCA consumes 187J energy whereas BPC consumes 181J energy. When any node dies in cluster 1 (refer to Fig. 5-9) in SSCA, other clusters still have much energy left. The directional data traffic in the WSN makes cluster 1 die much earlier than the others. However, the average residual energy in BPC is almost the same. BPC thus presents a far better balance in power consumption than does SSCA.

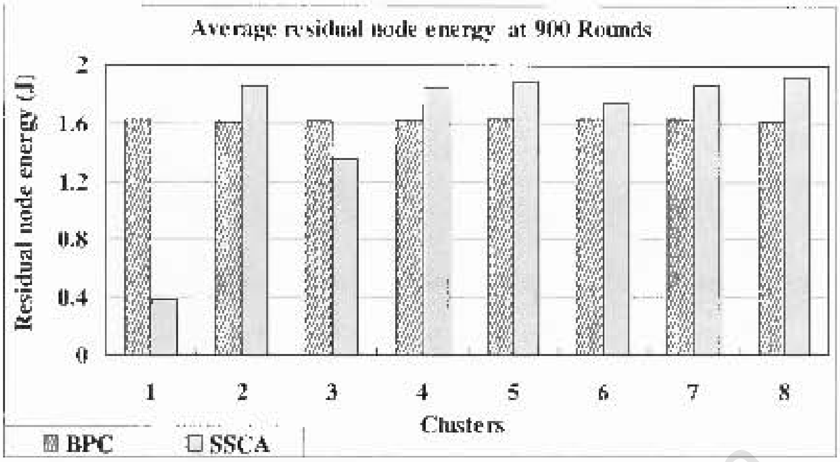


Figure 5-11. Average residual node energy at 900 Rounds in BPC and SSCA.

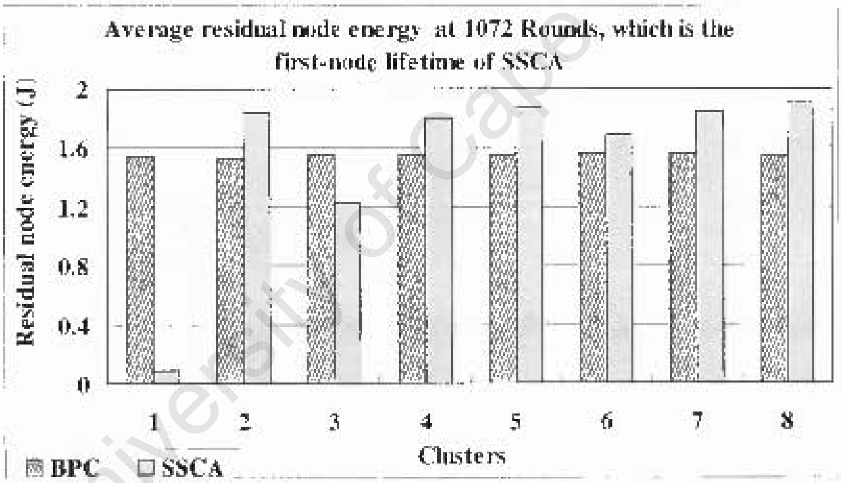


Figure 5-12. Average residual node energy at 1072 Rounds in BPC and SSCA.

After cluster 1 in SSCA dies, the CH in cluster 2 or cluster 3 (refer to Fig. 5-9) needs to increase transmission power to reach the sink, resulting in higher burden carried by these clusters, which makes them drain energy more rapidly. Fig. 5-13 shows the average residual node energy at 1750 rounds. Up to 1750 rounds, SSCA consumes 317J energy whereas BPC consumes 292J energy. The difference in consumed energy between SSCA and BPC is 25J whereas this difference is only 6J at 1072 rounds. Though no energy is consumed in cluster 1 after it dies prematurely in SSCA, the increase of power consumption caused by the higher transmission power in cluster 2 or cluster 3 is more significant. SSCA therefore presents a higher rate of consuming energy after cluster 1 dies.

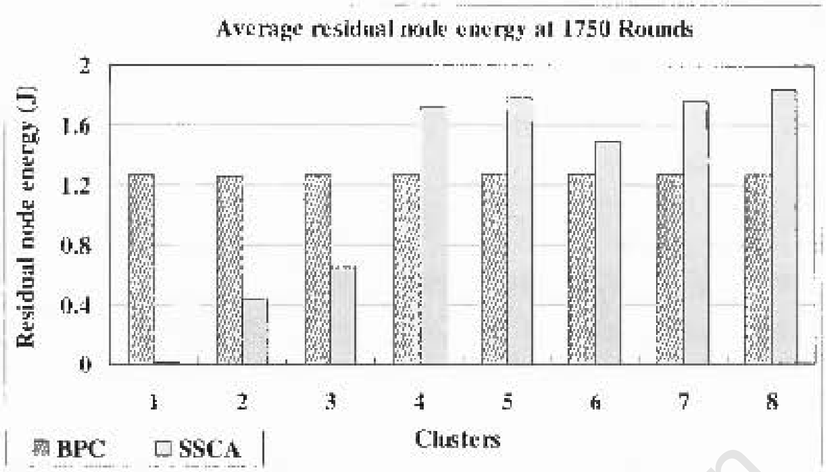


Figure 5-13. Average residual node energy at 1750 Rounds in BPC and SSCA.

Fig. 5-14 shows the average residual node energy at 2298 rounds, which is the connectivity lifetime of SSCA. Up to 2298 rounds, SSCA consumes 389J energy whereas BPC consumes 383J energy. The difference in consumed energy between SSCA and BPC, which is 6J at 2298 rounds, becomes smaller than the one at 1750 rounds, which is 25J. This is because as more clusters die, the decrease of power consumption due to the premature cluster death is more significant than the increase caused by the higher power needed to reach the sink. The directional data traffic makes SSCA waste more than 50% energy. At the connectivity lifetime of SSCA, BPC still has 1J energy left in each node.

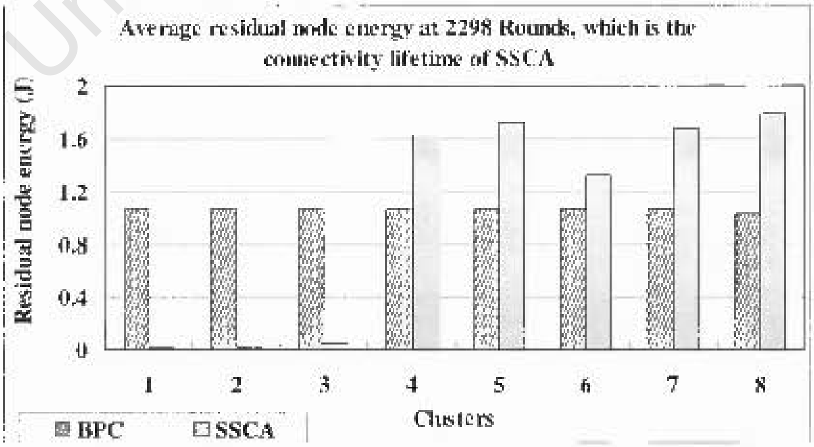


Figure 5-14. Average residual node energy at 2298 Rounds in BPC and SSCA.

Fig. 5-15 compares the residual node energy in BPC and SSCA at their respective first-node lifetime. The first-node lifetime in SSCA is only 1072 rounds whereas it is 4682 rounds in BPC, which is a 4.37 times improvement. At the first-node lifetime, SSCA consumes only 187J energy whereas BPC consumes 780J energy. The low total residual energy of the network means the power consumption is well balanced. BPC performs far better in balancing power consumption throughout the network that does SSCA.

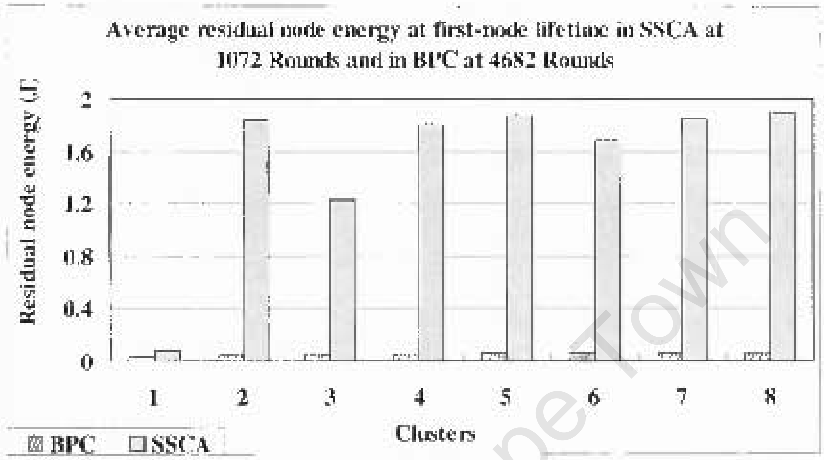


Figure 5-15. Average residual node energy at first-node lifetime of BPC and SSCA.

Fig. 5-16 compares the residual node energy in BPC and SSCA at their respective connectivity lifetimes. The connectivity lifetime in SSCA is only 2298 rounds whereas it is 4736 rounds in BPC. Up to connectivity lifetime, SSCA consumes 389J energy whereas BPC consumes 789J energy. The wasted energy in SSCA and BPC is therefore 411J and 11J respectively. BPC performs far better in using the energy than does SSCA.

The difference between the first-node lifetime and the connectivity lifetime in SSCA is 1226 rounds, which is even longer than its first-node lifetime (1072 rounds). This means that, though the connectivity lifetime is as long as 2298 rounds in SSCA, some clusters are out of coverage after 1072 rounds (first-node lifetime). It has been stated that each cluster is supposed to be (almost) covered by only one active node. When any node dies in the network, almost all nodes in that cluster are dying as the power consumption of the active node (CH) is well balanced. This means that the entire cluster will not be monitored any longer, thus degrading the network performance. This difference in BPC is only 54 rounds. Compared to the first-node lifetime (4682 rounds), this difference is negligible. BPC therefore not only balances power consumption of the network but also improves network performance as well by preventing some areas from being prematurely out of coverage.

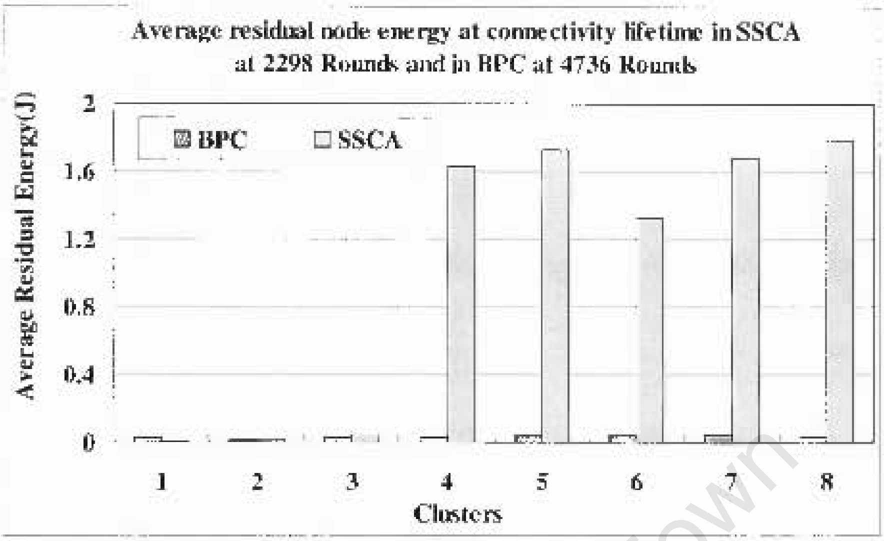


Figure 5-16. Average residual node energy at connectivity lifetime of BPC and SSCA.

Fig. 5-17 shows the total number of live nodes up to connectivity lifetime. In SSCA, the nodes in the same cluster die almost simultaneously. This is because the CH role is rotated frequently and the power consumption of the node is well balanced within the clusters. This balance however is only acquired within the cluster, not the entire network, as it does not consider the directional data traffic. After the cluster nearest the sink dies, the connectivity between the network and the sink is still maintained as other clusters can still relay data to the sink using a longer transmission range within the maximum limit. The connectivity between the sink and the network is lost when the distance between the sink and its nearest cluster which still has energy exceeds the maximum transmission range of the nodes. In BPC, however, the clusters die almost simultaneously because the power consumption is well balanced throughout the network.

The distribution of the residual node energy at connectivity lifetime in SSCA and BPC is presented in Fig. 5-18. All nodes in BPC have less than 5% energy left. However, in SSCA, many nodes have a good proportion of residual energy: about 13% of nodes have 50% energy left and about 50% of nodes have more than 75% energy left. The wasted residual energy in SSCA is therefore quite significant. This means that the balance of power consumption among the nodes in typical WSNs is actually important for increasing WSN lifetime [37].

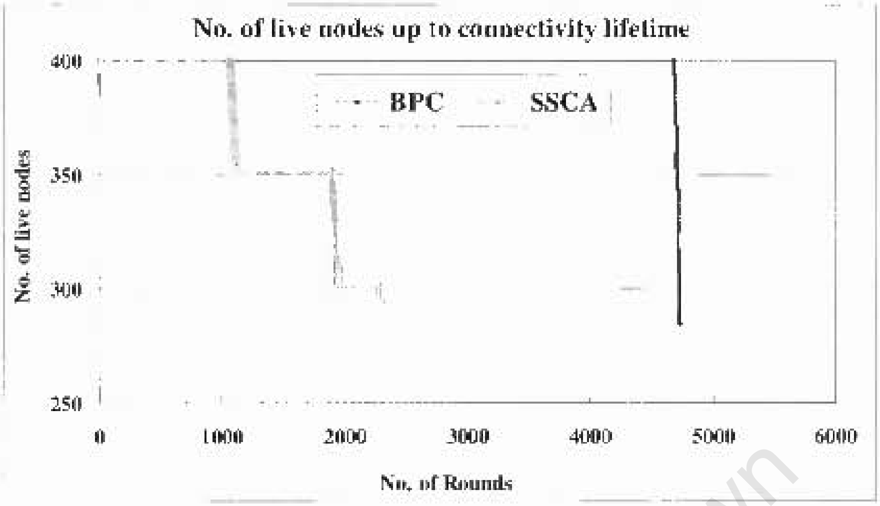


Figure 5-17. Number of live nodes up to the connectivity lifetime.

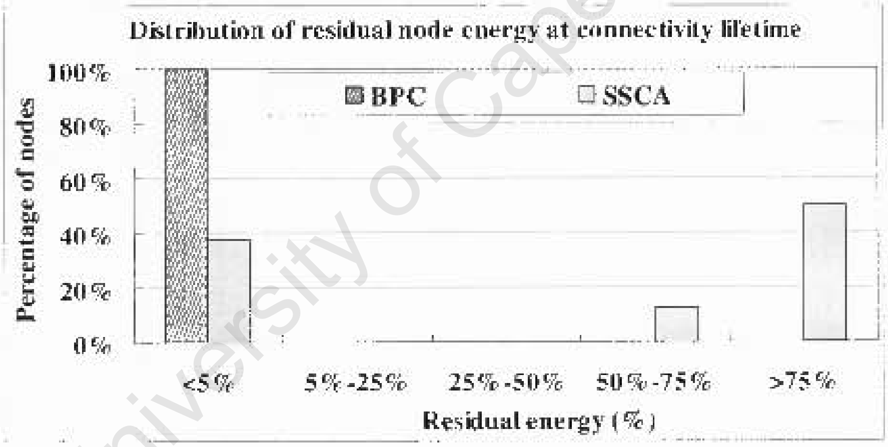


Figure 5-18. Distribution of residual node energy at connectivity lifetime.

Fig. 5-19 shows the energy not yet used up to connectivity lifetime. This plot illustrates that before the first-node lifetime, the energy efficiency is quite similar in SSCA and BPC as they almost consume the same amount of energy at the same number of rounds. After its first-node lifetime, BPC consumes energy more slowly than does SSCA as some nodes in SSCA need higher power to reach the sink. As more clusters die in SSCA, however, BPC consumes energy more quickly than does SSCA. This is because that the decrease in power consumption of SSCA caused by the smaller number of nodes that are consuming energy becomes more significant than the increase in power consumption cased by the higher power needed to reach the sink, thus decreasing the rate of consuming energy. Fig. 5-19 also shows that, at connectivity lifetime, SSCA consumes less than 50% energy whereas BPC consumes almost all energy in the network. This remaining energy at connectivity lifetime cannot be used by the network any longer and is wasted. BPC therefore presents an obvious improvement compared to SSCA.

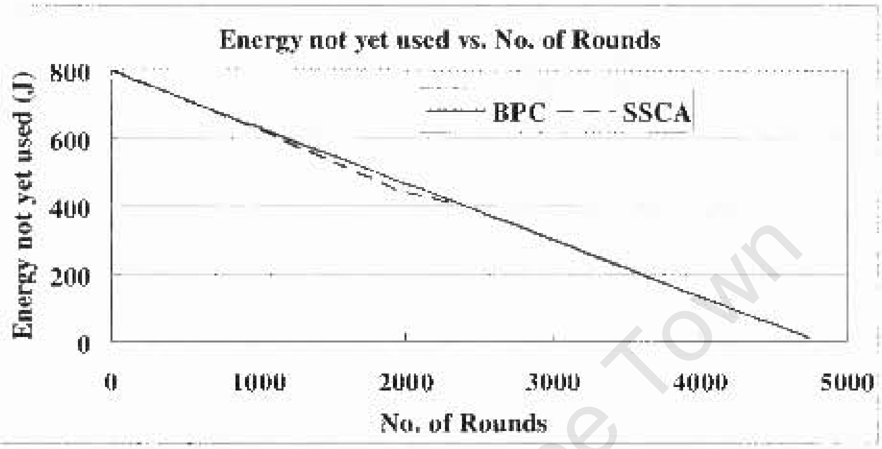


Figure 5-19. Energy not yet used up to connectivity lifetime.

Fig. 5-20 shows the total number of data packets delivered to the sink up to connectivity lifetime. This illustrates the effectiveness of BPC in delivering data. Up to connectivity lifetime, SSCA only delivers 337240 packets whereas BPC delivers 757040 packets to the sink. BPC offers improvements in data delivery by 124.5% over SSCA. Among these network data, 99.0% and 50.9% packets are delivered to the sink before the first-node lifetime, which are from the entire network, in BPC and SSCA respectively. BPC not only improves the energy efficiency but also the network performance.

5.4.4 Inferences from the result

From the simulation results, the following inferences can be achieved:

The characteristic of directional data traffic towards the data sink in WSNs with high node density degrades the energy efficiency greatly, as well as the performance because it results in premature node death. By taking into account this characteristic when designing algorithm which combining clustering and scheduling, the energy efficiency of the network can be dramatically improved, as well as the network performance because it both prevents the premature node death and makes the coverage of the entire network more reliable.

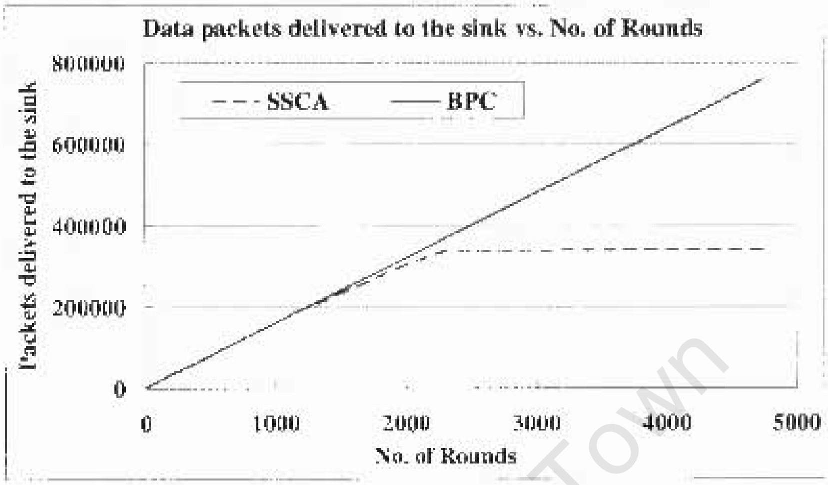


Figure 5-20. Number of packets delivered to the sink.

5.5 Summary and discussions

5.5.1 Summary

An algorithm, which combines clustering and scheduling to balance power consumption throughout a homogeneous WSN with high node density, has been proposed in this chapter. This algorithm only requires the CH in each cluster to be active to monitor that cluster and to relay data for others. By frequently rotating the CH role, the burden of monitoring the cluster and relaying data for others is balanced and the cluster coverage is guaranteed. By designing the clusters in such a way that the energy store in the cluster is proportional to its power consumption, cluster lifetimes are equalized. The simulation results show that this algorithm improves both energy efficiency and network performance.

5.5.2 Discussions

The design of the algorithm and the conducted performance evaluation only focus on energy efficiency. Other issues like link failure, data re-transmission, and network security are not taken into account.

Furthermore, similar to some clustering algorithms [30, 122], the data packets are forced to be relayed by the neighboring CHs during the simulation. The selection of the route, though having better balanced power consumption throughout the network, may not be the most energy efficient one.

Further work will improve BPC design by considering more issues as motioned. It will also consider further improvement in energy efficiency by selecting backbone routes based on the consideration of total needed energy of the routes.

6 Reducing unnecessary broadcasting and balancing power consumption (RBBPC) in clustered MANETs

6.1 Strategies to improve energy efficiency in clustered MANETs

Chapter 3 analyzed existing energy-efficient clustering algorithms for MANETs and concluded that a clustering algorithm which reduces unnecessary broadcasting and balances power consumption throughout the network is necessary.

Unlike WSNs which function to collect data for certain special tasks, a typical MANET is usually applied to enable communications between the nodes in the network. Therefore, in a MANET, no data sink is necessary and thus no data traffic is directed towards a sink. This means that the data traffic in a MANET is usually more uniform than that in a WSN. Equalizing cluster sizes in a MANET can actually balance power consumption among the clusters. Unfortunately, the free and arbitrary movement of the nodes brings more challenges to maintenance of the clusters and cluster sizes, and associated energy efficiency of the network.

In a clustered MANET, the clusters may frequently be changed by the movement of the nodes. It is therefore necessary to maintain the clusters. A typical traditional clustering algorithm broadcasts control messages to periodically update the links or the node states. Each CH maintains a list of IDs of the member nodes in the cluster. This periodic updating detects the changes in the cluster by receiving the responses of the broadcast control messages from the member nodes. The cluster information can therefore be updated. Typically, in a MANET with higher node speed, the clusters are changed more frequently. Thus, to maintain these clusters, more frequent broadcasting is required. Periodic broadcasting therefore needs to meet the requirement of the potentially highest node speed in a MANET. The negative outcome of this is that when the nodes are moving with lower speed, the unnecessary broadcasting wastes network energy. Adaptive broadcasting according to the requirement of the node speed helps to reduce unnecessary broadcasting when the node speed is lower. To broadcast control messages adaptively, the speeds of the nodes need to be determined. Most clustering algorithms use GPS technology or calculations to predict the movement or to acquire the speed information. These algorithms thus draw on extra support or more powerful nodes in the network [142-146].

This chapter proposes an efficient method to assess the node speed and the broadcasting frequency that this speed needs. Gavalas also proposes an algorithm to broadcast control messages adaptively [72]. The adaptive broadcasting frequency in his algorithm, however, is determined arbitrarily. Therefore, this adaptive broadcasting frequency may exceed or may not satisfy the required frequency of the node speed. The proposed algorithm RBBPC builds the relationship between the speed and the rate that this speed changes the topology. This rate can be assessed by measuring the number of re-clusterings in a clustered MANET within a limited interval when some nodes are moving at this speed. The necessary broadcasting frequency of each speed can then be determined and the control messages can be broadcast adaptively according to the requirement of the speed.

Different to a WSN, a typical MANET usually has relatively lower node density [11]. A CH in a clustered MANET is more likely to die. This occurs especially in algorithms that extend the service time of CHs to improve cluster stability [72]. The lack of mobile nodes due to energy depletion may cause network partition and communication interruption [11, 63]. Hence, it is important to balance power consumption among mobile nodes to avoid premature node failure, especially in networks with low node density [103].

Similar to the CHs in a clustered WSN, rotating the CH role in a clustered MANET also distributes the higher burden of the CHs among the mobile nodes. Besides the rotation of CH, equalizing cluster sizes in a MANET further balances power consumption among the CHs and the clusters. As the nodes move freely and arbitrarily, however, this size balance has more challenges. When the speed is low, maintaining the cluster sizes within a limited range is feasible. When the speed is high, the movement of nodes may change the clusters and cluster sizes more frequently. A limited range of cluster sizes will lead to frequent re-clustering, thereby increasing the power consumption of maintaining the clusters and cluster sizes. RBBPC, therefore, adaptively balances the cluster sizes according to the node speed. When the speed is higher, the clusters are allowed to have a greater difference in sizes. Otherwise, a more limited range of cluster sizes will be applied to better balance power consumption throughout the network [147].

6.2 Proposed clustering algorithm-RBBPC

6.2.1 Network model and energy model

Energy-limited identical mobile nodes are distributed randomly in the network. Each node has a unique ID number. The nodes can move freely in a linear pattern. Each node is aware of its residual energy information yet is not aware of the location information. Each node has two levels of transmission ranges. The shorter one is for intra-cluster communication whereas the greater one is for inter-cluster communication.

The nodes are grouped into multi-hop clusters with a CH in each cluster acting as co-ordinator. Intra-cluster communication applies a proactive routing protocol whereas inter-cluster communication uses a reactive protocol. The clusters and the intra-cluster routing information are maintained by control messages being broadcast adaptively.

Each node can be either a member node or a CH. It is assumed that each node has two levels of power consumption. One is P_{member} for a member node and the other is P_{CH} for a CH when it has only one member node. The power consumption of a member node is linear with time, whereas the power consumption of a CH within a certain interval is relative to (i) the time it serves the cluster, (ii) the ratio between the transmission range of intra-cluster communication and the transmission range of inter-cluster communication and (iii) the number of the member nodes that it serves. Within a limited interval T, the total consumed energy E of a node is determined by (6.1), the definition of each parameter is shown in Table 6-1.

$$E = T_{member} \times P_{member} \times k_a + \sum_i (P_{CH} \times n_i \times k_b \times T_i) \tag{6.1}$$

in which:

$$T = T_{member} + \sum_{i} T_i \tag{6.2}$$

Table 6-1. Definitions of each parameter of power consumption.

Parameters	**Definitions**
T_{member}	The total time being a member node
k_a	The coefficient of member node relative to the node speed
n_i	The number of the member nodes in the cluster at i^{th} CH period
k_b	The coefficient of CH relative to the node speed
T_i	The length of the service time of the CH at i^{th} CH period

6.2.2 Size-balancing cluster organization

RBBPC aims at improving energy efficiency and balancing power consumption of MANETs by maintaining an effective network topology, which adapts to node mobility. This distributed RBBPC uses a mobility-sensitive model to dynamically organize the nodes into clusters in such a way that (i) the topology can be more responsive and optimal when speeds are low and more efficient when they are high and (ii) the cluster sizes can be balanced adaptively according to the node speed.

When a network starts to work, one of the nodes has a certain probability of becoming a CH. This probability is a pre-determined parameter depending on factors such as the size of the network, the extent of spacing between the nodes in the network and the required backbone complexity. The self-selected CHs broadcast their CH announcements to others immediately.

A node that receives a CH announcement forwards it to all its one-hop away neighbors. The announcement has a field called the cluster radius field, which specifies the number of hops to which the announcement packet can be forwarded. On reception of this packet, a node that is not a self-selected CH responds this announcement to confirm that it is joining this CH as a member node. This node will then not respond to any subsequent CH announcement. The node then decreases the radius field by one and forwards the CH announcement to all its one-hop away neighbors if the value in the radius field is not zero. The radius value is another pre-determined parameter that specifies the maximum number of hops a node in the network can have from its CH. This process will be repeated until all nodes are clustered. It can be seen that the communication complexity of clustering in RBBPC is quite low.

The formed clusters, however, may have different sizes. The next stage is to balance the sizes of the clusters. It is assumed that U is the upper bound and L is the lower bound of the cluster size in RBBPC. These U and L are adaptive in accordance with the node speed. When the speed is low, they will be similar. Otherwise, more difference is allowed. The CH of each cluster needs to determine whether the number of member nodes in the cluster is in the L and U limitations. If it is, no further process will be performed. If it is not, the process of size maintenance will be invoked.

Ohta's algorithm [74] proposes an algorithm which also limits the cluster sizes within the U and L bounds to balance loads among the clusters. This algorithm performs a merge mechanism if the cluster size is smaller than L or a division mechanism if the cluster size is greater than U. L and U are not adaptive. However, with these two size maintenance mechanisms, the clusters in this algorithm still experience significant size difference. RBBPC not only proposes to balance cluster sizes adaptively, but also proposes a simpler mechanism.

When a CH, named *S1*, finds that it does not have enough nodes, it will announce this information to its neighboring CHs. When a neighboring CH receives this information, it will reply to *S1* with the number of nodes in its own cluster. In case *S1* receives more than one response, it will pick the one that has the most number of member nodes and send a message requiring this CH to forward it a member node. On reception of this request, the CH allows the member nodes to receive the announcement from *S1*. The member node that receives this announcement and needs the highest number of hops to access its current CH is allowed to join *S1*. This process will be repeated if necessary until the cluster size is within the limited range. However, the repetition usually only happens during the cluster formation phase when the network starts to work. Because the control messages are broadcast adaptively according to the requirement of each speed, any change in the cluster will be detected immediately.

When a CH, named *S2*, finds that it has too many nodes in the cluster, it will not announce this information to other CHs. Yet it allows the member node that needs the highest number of hops to access it to leave this cluster. This member node will then receive the control message from other CHs and join the CH that has the fewest number of nodes as a new member node. This process will be repeated if necessary until the cluster size is in the limited range. However, as with the process of size maintenance when the CH has too few nodes, this repetition also usually only happens during the cluster formation phase when the network starts to work.

It can be seen that the re-clustering triggered by size maintenance does not have a resultant ripple effect, thus further improving energy efficiency.

6.2.3 Data communication

During the data communication phase, when a source node is ready to send data to a destination node, two scenarios may occur depending on whether the destination is in the same cluster as the source node or not.

If the destination node is in the same cluster as the source node, intra-cluster communication occurs. As intra-cluster communication applies proactive routing protocol and each node holds the routing information of the whole cluster, a node can explore the routing information to the destination from its routing table. RBBPC considers balancing power consumption throughout the network so that a route formed by the nodes with higher residual energy is selected to relay the data, the process of routing selection is similar to that in BRPC (refer to Section 4.2.3).

If the destination node is not in the same cluster as the source node, inter-cluster communication occurs. Inter-cluster communication uses a reactive routing protocol. Should the source CH find more than one valid route, it will then select the route based on the consideration of residual node energy, which is the same as the routing selection in the intra-cluster communication.

During the inter-cluster communication, once a source CH determines the route to the destination CH, it will cache this routing information. This cached routing information is to be valid for a limited interval. This limited interval is referred to as a *timer*. Most traditional algorithms set this timer to be a constant. To make sure that the cached routing information is always valid; the timer needs to be short. This means that each time inter-cluster communication occurs, the cached routing information is usually supposed to be no longer available (though in fact it may still be available, especially when the nodes are moving at low speed), thereby requiring the CHs to frequently re-initiate the routing discovery process. This repeated routing discovery process causes considerable flooding which consumes significant energy.

In a realistic application, the node speed may change from time to time. When the speed is low, the cached routing information will be valid for a longer time, making repeated routing discovery unnecessary. RBBPC therefore, again, builds an adaptive timer to reduce the unnecessary flooding caused by backbone routing discovery. When the speed is high, the timer is short; otherwise it will be longer. Similar to the adaptive broadcasting frequency, the timer is also relative to the node speed. The relationship between the timer T_{timer} and the broadcasting period $T_{broadcasting}$ is built as (6.3), in which K is the ratio between the transmission ranges of inter- and intra-cluster communication.

$$T_{timer} = K \times T_{broadcasting} \tag{6.3}$$

6.2.4 Cluster maintenance

In a clustered MANET, the CH is responsible for maintaining local membership and global topology information [11, 148]. During the cluster maintenance, the following issues need to be considered: intra-cluster routing information and cluster topology.

The nodes in a MANET can move freely and arbitrarily. When they leave the original cluster or join a new cluster, the intra-cluster routing information and the topology of the original cluster are changed. The proactive routing protocol for intra-cluster communication requires updating of the link to guarantee the availability of the tabled routing information in the nodes. This link is updated by broadcasting control messages and by receiving the responses of these messages from the member nodes. Unlike the periodic broadcasting in the traditional clustering algorithms, the broadcasting in RBBPC adapts to meet the minimum requirement of each node speed. (The issue of how to broadcast control messages adaptively is to be discussed in Section 6.2.5.). When the corresponding replies arrive at the initial CH, the CH updates the intra-cluster routing information and broadcasts it to its member nodes.

If a CH does not receive a reply from the member node the ID of which is previously in its list, it assumes that this member node has left the cluster and will remove its ID from the list. If a member node does not receive the control message from its CH within a limited interval, it assumes that it has already left its original cluster. It will then receive the control messages from other CHs and join the CH which has the fewest number of member nodes. If a CH can not get any response of the control messages from its member nodes, it can then determine that it (the CH) has left the cluster. To avoid the disappearance of the cluster, the CH will keep its role and to request the neighboring CHs to donate their member nodes.

6.2.5 Adaptively broadcasting control messages

Adaptive broadcasting according to the requirement of the node speed reduces unnecessary broadcasting. This adaptive broadcasting, however, needs the speed information of the nodes. A simple yet effective method to evaluate the node speed is necessary.

1. Node speed and the rate this speed changes the cluster topology

RBBPC proposes a method, which does not need location information or powerful computational capability of nodes to assess the speeds of the nodes. It first builds the relationship between the speed and the rate that each speed changes the cluster topology. The rate can be determined by measuring the number of re-clusterings that occur within a limiter interval when a number of nodes are moving at a specific speed in the network. According to this relationship, the required broadcasting frequency of each speed can be determined. Using this relationship, the node speed can also be determined by accounting the number of re-clustering within a limited interval.

Gavalas also [72] proposes a similar method to assess the node speed using the information of topology changes. His method, however, did not establish the relationship between the speed and the rate that this speed changes the topology of the cluster. Without this relationship, the control messages can still be broadcast adaptively according to the cluster-topology changes. However, the broadcasting frequency may exceed or may not satisfy the required frequency for maintaining the clusters. If the broadcasting frequency exceeds the required one, the broadcasting will waste energy. Otherwise, it cannot maintain the clusters or update intra-cluster information efficiently, thereby increasing the rate of data loss when the member nodes use this stale routing information.

The CH in each cluster holds the ID information of all nodes in the cluster. Each time the CH broadcasts control messages, it may detect ID changes in the cluster from the responses it received from the member nodes. Each time ID changes are detected, re-clustering occurs. If the broadcasting period is a constant, these changes may become more significant when the node speed increases. Therefore, changing the speeds of the nodes and then measuring the number of re-clusterings when some nodes are moving within a limited interval can build the relationship between the speed and the rate this speed changes the cluster topology.

It is assumed that when the nodes are moving at different speeds within the same limited interval T, the expected number of re-clusterings within each unit of time $E\left(\overline{Times_{re-c}}\right)$ are linear with the speeds S_i and the number of nodes that are moving at this speed N_i, as shown in (6.4), where k_c is the coefficient of this linear relationship.

$$E\left(\overline{Times_{re-c}}\right) = k_c \times \sum_i (N_i S_i) \tag{6.4}$$

For example, the $E\left(\overline{Times_{re-c}}\right)$ when 4 nodes are moving at 6m/s within 8s is the same as the $E\left(\overline{Times_{re-c}}\right)$ when 3 nodes are moving at 8m/s within 8s.

2. Broadcasting control messages adaptively

After the relationship between the node speed and rate that this speed changes the cluster topology is established, the next step is to determine the broadcasting period for each node speed. This is important in order to meet the requirement of each node speed, as stated that both too high frequency and too low frequency of the broadcasting have disadvantages.

From (6.2), the expected number of re-clusterings $E(Times_{re-c})$ within T, when N_i nodes are moving at S_i within T, can be described as (6.5):

$$E(Times_{re-c}) = k_c \times T \times \sum_i N_i S_i \tag{6.5}$$

Expected average node speed $E(S)$ can be determined by (6.6), in which $\overline{N_i}$ is the average number of nodes that are moving at different speeds and k_d is the coefficient of the relative node speed:

$$E(S) = \frac{E(Times_{re-c})}{k_d \times \overline{N_i}} \tag{6.6}$$

Expected broadcasting period $E(T_{bd})$ can then be determined by (6.7):

$$E(T_{bd}) = \frac{T}{\dfrac{E(Times_{re-c})}{2 \times N_{Cluster} \times \overline{N_i} \times \dfrac{1}{N_{hop}}}} \tag{6.7}$$

where 2 means each node leaving the original cluster and joining another one triggers two re-clustering processes, $N_{Cluster}$ is the number of the clusters, $\overline{N_i}$ is the average number of nodes that are moving in each cluster. Re-clustering only occurs when a node leaves or joins a cluster, however, for a multi-hop cluster, the intra-cluster routing information may be changed even when the nodes are moving within the cluster. Therefore, a coefficient of $1/N_{hop}$ is needed for the efficient updating of the intra-cluster routing information.

An example of how to determine the broadcasting period according to the speeds is shown in Table 6-2. Assuming that there are two two-hop clusters in a clustered MANET and 10 nodes are moving at the same speed which increases from 2m/s to 20m/s. During the phase of setting the relationship between the speed and the rate that this speed changes the cluster topology, the number of re-clusterings is to be detected of each speed within 30 successive broadcasts. Each of which is sent out every second. As assumed, the expected number of re-clusterings is linear with the time interval, the node speed and the number of nodes that are moving (refer to Section 6.5). The expected time when the first re-clustering occurs can then be obtained. Effective broadcasting is expected to detect some changes in the cluster. However, these changes should be minimized to guarantee that the intra-cluster routing information is updated on time to avoid any data loss. The control message therefore needs to be sent out when the first re-clustering is expected to occur, as shown in the final row of the table.

Table 6-2. Example of determining the broadcasting Period.

Node speeds (m/s)	2	4	6	8	10	12	14	16	18	20
Re-clustering (times)	30	60	90	120	150	180	210	240	270	300
Number of nodes that are	5	5	5	5	5	5	5	5	5	5
Time interval (s)	30	30	30	30	30	30	30	30	30	30
Number of hops of the cluster	2	2	2	2	2	2	2	2	2	2
Expected first re-clustering (s)	10	5	5/2	10/3	2	5/3	10/7	5/4	10/9	1
Expected broadcasting period	10	5	5/2	10/3	2	5/3	10/7	5/4	10/9	1

6.2.6 Balancing power consumption throughout the network

Many algorithms have been proposed to improve energy efficiency for MANETs by reducing energy consumed in routing data. To maximize lifetime of the network, however, the power consumption needs to be balanced throughout the network, thereby preventing premature node death and thus the premature network disconnection [37]. RBBPC applies two methods to balance power consumption throughout the network. The first one distributes the higher burden of the CH among the nodes by rotating the CH role: a method that has been used in many existing clustering algorithms. The second balances power consumption among the clusters by balancing cluster sizes adaptively [120, 147].

1. Distributing the higher burden of CH among nodes

The CH needs to be rotated as it has a higher burden than member nodes. When a CH decides to give up its role, it will broadcast a *dismissal control message* to its member nodes. On reception of this announcement, the member nodes within the cluster exchange the residual energy information. The one with the highest residual energy becomes a CH and announces itself as a CH for next CH period together with the ID list of the member nodes that it received from the previous CH. Only the nodes in the ID list will respond to this announcement and the routing information is updated when the CH receives all responses from its member nodes. Thus the higher burdens of CHs are distributed among the nodes. However, to reduce the destruction of stability caused by CH rotation; the service time of the CH is suitably extended. This suitable extension means that, before any node dies, each node in the network can get at least one chance to become a CH. It can be seen that, the re-clustering triggered by CH rotation in RBBPC is completed without resultant ripple effect, thereby saving more energy.

During the cluster organization (Section 6.2.2), the CH is in the central area of the cluster, thus power consumption of intra-cluster communication is reduced. However, during the rotation of the CH, any node in the cluster has the chance of becoming CH according to its residual energy. This means that the node that becomes the CH in the consequent periods is not necessary in the central area of the cluster. Maintaining the CH at the central area by re-clustering with ripple effect improves energy efficiency of intra-cluster communication. However, the energy consumed by the flooding caused by re-clustering with ripple effect is quite considerable [11, 14]. Furthermore, re-clustering with ripple effect will further reduce the stability of the network. RBBPC thus rotates the CH role according to residual node energy without considering the node location in the cluster.

2. Adaptively balancing cluster sizes

Rotation of the CH role distributes the higher burden of CH among the nodes. Balance between cluster sizes further reduces the difference in burdens between the CHs and between the clusters. To prevent premature energy depletion of the cluster with higher burden and in turn maximize its lifetime, the difference in size of clusters needs to be minimized. The existing methods, either by limiting the maximum cluster size or by strictly limiting the maximum and minimum cluster sizes, have limitations (refer to Section 3.2.2). Adaptively maintaining the cluster size better balances power consumption, thus further improves energy efficiency of the network.

It is assumed that the difference in U (upper bound) and L (lower bound) is η. Equation (6.8) explains the difference in cluster sizes S_{diff} according to the node speed M_{speed}. When the speed is low, the cluster size may not be changed frequently and therefore it is worth equalizing the cluster size, thus η will be a lower value. When the speed is high, to maintain the cluster sizes within a strict range needs frequent re-clustering. This frequent re-clustering will further reduce the stability of the cluster and consume more energy. Therefore, when the node speed increases, η will be greater, thereby liberating the network from frequent re-clustering.

$$S_{diff} = U - L = \eta \propto M_{speed} \tag{6.8}$$

6.3 Performance evaluation

This subsection evaluates the performance of RBBPC, which broadcasts control messages and balances cluster sizes adaptively. Two existing clustering algorithms are used for comparison. The first limits the maximum cluster sizes and broadcasts control messages periodically, which is similar to the highest degree algorithm and the lowest ID algorithm in [71]. This algorithm is referred to as Maximum Sizes with Periodic Broadcasting (MSPB). The other algorithm limits maximum cluster sizes, maximizes service time of the CH and broadcasts control messages adaptively, which is similar to the algorithm in [72]. This algorithm is referred to as Maximum Sizes and maximum CH Service time with Adaptive Broadcasting (MSSAB).

In a WSN, the nodes have the same destination and are stable, thereby making it feasible to evaluate the first-node lifetime and the network lifetime. The measurement of the lifetimes in a MANET, however, differs from those in a WSN. First, as the data sink is the same destination of all sensor nodes, it is therefore quite clear where all data need to be relayed, making it possible to measure the power consumption of relaying data. In a MANET, however, the nodes can communicate with each other freely, thereby making it difficult to determine the destination. The selection of the destination nodes greatly affects the power consumption of the communication. The power consumption of the communication between the nodes in two clusters which are far away from each other will be considerably higher than the power consumption of the communication between the nodes in the same cluster. It is therefore meaningless to measure the power consumption of communication in such a scenario. Secondly, in a clustered WSN in which the CHs send data to the sink using multi-hop routes, the loss of connectivity between the network and the sink is usually caused by the (premature) node death (refer to Section 4.3.2 and Section 5.4.2). In a MANET, the free and arbitrary movement of the nodes may also cause the nodes to become disconnected from each other. It is therefore difficult to determine whether this disconnection is caused by the movement of the nodes or by the premature energy failure, thereby making the measurement of the connectivity lifetime meaningless. This performance evaluation therefore only assesses the first-node lifetime of the network.

The main application of a WSN is usually to collect data from the network [14]. It is therefore reasonable to request the network to send the necessary data, which cover (almost) all the information about the network to the sink, during each round. The main application of a MANET, however, is to enable communication between the nodes [3]. Besides the challenge in determining the destination, it is also difficult to decide how many nodes are sending data. The issue becomes that in a MANET it is almost impossible to determine how many nodes are sending data and to which nodes.

The performance evaluation is therefore conducted by combining the simulation and mathematical analysis, as shown in Fig 6-1. It first uses simulation to build the relationship between the speed and the rate this speed changes the cluster topology (Step 1). Using this relationship, the necessary control messages can be broadcast adaptively according to each speed (Step 2). The next step builds the relationship of the power consumptions between the algorithms by mathematical analysis, which uses the results from Steps 1 and 2 (Step 3). The first node lifetime is then determined (Step 4) based on the relationship of the power consumption from Step 3 and the energy model of the network (refer to Section 6.2.1).

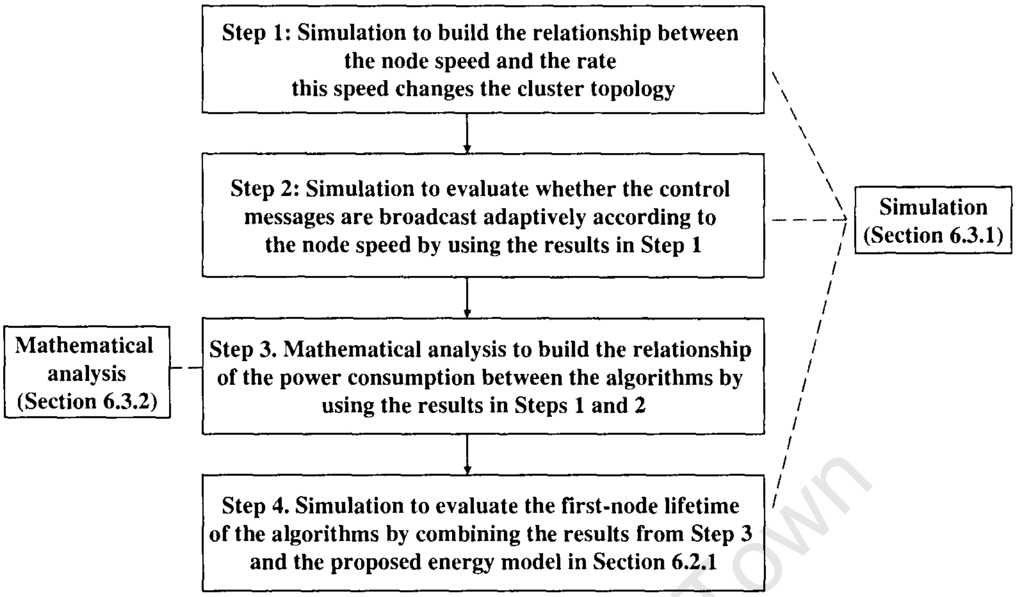


Figure 6-1. Steps of the conducted performance evaluation.

6.3.1 Simulation

1. Simulation scenario

Nodes, 100 in total, are uniformly distributed within an area of 1000m×1000m. Each node has 100 units of initial energy and has two transmission ranges: one is 200m for intra-cluster communication and the other is 400m for inter-cluster communication. The nodes, which are moving from 2-20m/s, are expected to be organized into 10 clusters. The control messages are broadcast according to the speeds of the nodes adaptively in RBBPC and MSSAB. Whilst this broadcasting in MSPB is periodic and is in accordance with the requirement of 20m/s, as has been stated that the periodic broadcasting needs to meet the requirement of the potentially maximum node speed (refer to Section 6.1)

2. Assessment requirement

Rate that each speed changes the cluster topology

The measurement of re-clustering builds the relationship between the node speed and the rate this speed changes the cluster topology. The number of re-clusterings within a limited interval when some nodes are moving at different speed will be measured.

Adaptive broadcasting

With the relationship built between the node speed and the rate that this speed changes the cluster topology, the broadcasting period of each node speed can be determined. This measurement of the broadcasting is to evaluate whether the control messages are actually broadcast adaptively or not in RBBPC and MSSAB. The maximum broadcasting frequency in RBBPC and MSSAB is to be the same as the periodic broadcasting in MSPB when the nodes are moving at 20m/s.

First-node lifetime

The measurement of the first-node lifetime compares the energy efficiency of the algorithms. RBBPC improves energy efficiency and further balances power consumption throughout the network so that it is expected to have longer first-node lifetime than that of MSPB and MSSAB. Though it broadcasts control messages adaptively, MSSAB does not distribute the higher burden of the CH among the nodes. The CHs in MSSAB will therefore die quickly.

3. Simulation results

Rate that each speed changes the cluster topology

Fig. 6-2 shows the results of the rate that each speed changes the cluster topology when 20 nodes are moving at the same speed, which increases from 2m/s to 20m/s. It shows that the total number of re-clusterings within 80 seconds is almost linear with the node speed.

In a realistic application, the nodes may move at different speeds, the expected total number of re-clusterings within a limited interval can be assessed according to (6.4). For example, when 10 nodes are moving at 6m/s, 15 nodes are moving at 8m/s and 6 nodes are moving 10m/s respectively within 40 seconds, the total number of re-clusterings ((10×6+15×8+6×10)×40) /(20×10×80)) ×235=141 (times) can be expected.

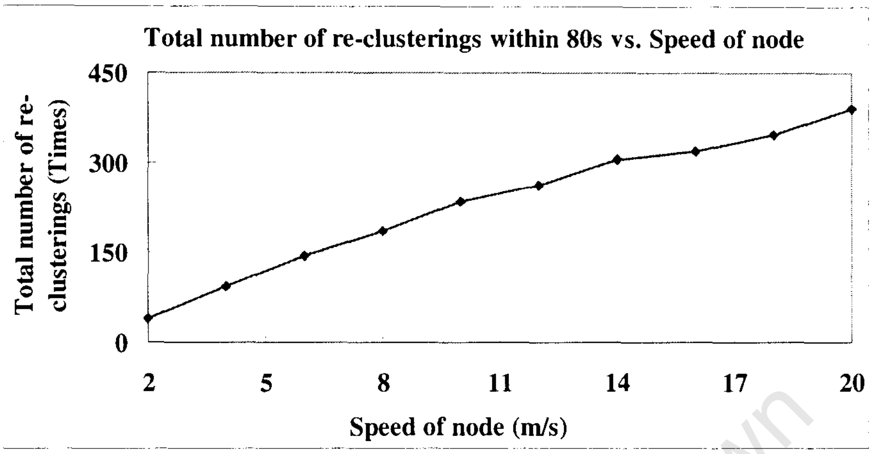


Figure 6-2. Rate that each respective node speed changes the cluster topology.

Adaptive broadcasting

When the relationship between the speed and the rate that this speed changes the cluster topology is built, the broadcasting frequency of each speed can be determined. The control messages can then be broadcast adaptively. The broadcasting frequency when all nodes are moving at the same speed is shown in Fig. 6-3 and when only 60 nodes are moving at the same speed is shown in Fig. 6-4.

It can be seen that, as expected, the control messages are broadcast adaptively. The frequency of the broadcasting increases linearly with the node speed, which is consistent with the rate that each speed changes the cluster topology (refer to Fig. 6-2). When 100 nodes are moving at 2m/s, according to (6.7), the $E(T_{bd})$ is about 8s and thus the frequency is about 0.125 (times/second), which is quite similar to the result in Fig. 6-4. However, traditional periodic broadcasting needs to meet the requirement of maintaining the clusters when the nodes are moving at 20m/s. The unnecessary broadcasting, when the node speed is lower, is significantly reduced in RBBPC. RBBPC, therefore, improves energy efficiency compared to MSPB.

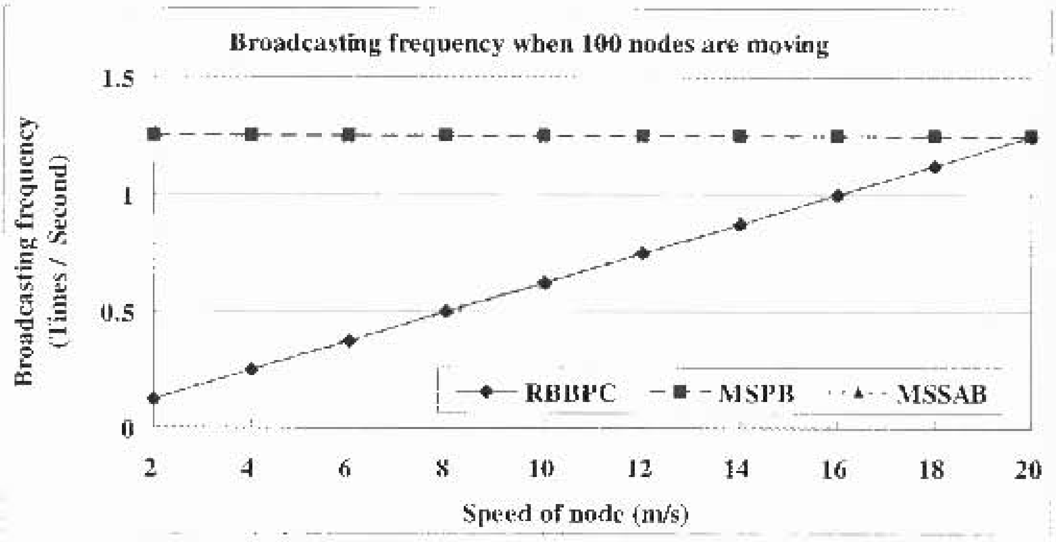


Figure 6-3. Broadcasting frequency when 100 nodes are moving.

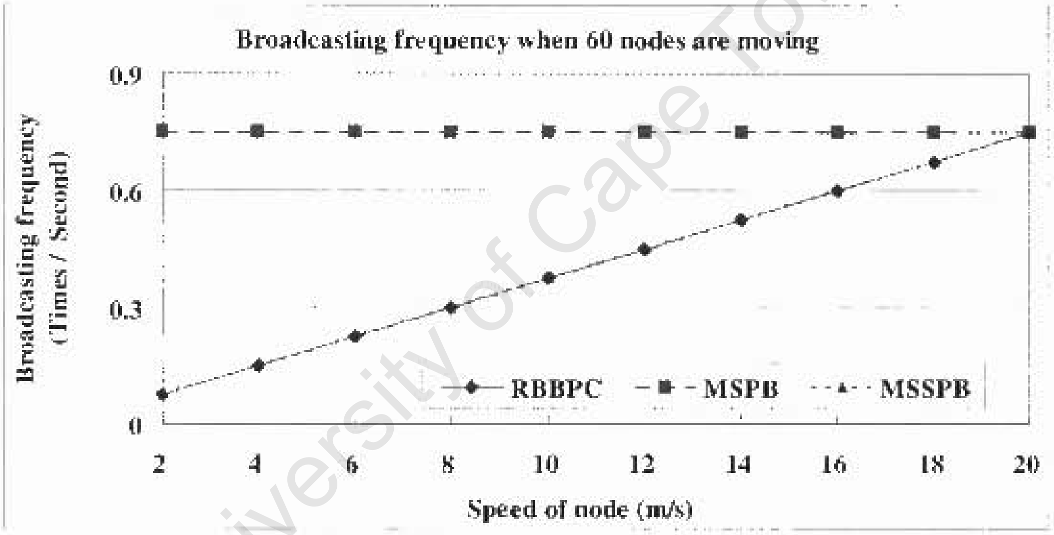


Figure 6-4. Broadcasting frequency when 60 nodes are moving.

Similar to the broadcasting frequency when 100 nodes are moving, the control messages are also broadcast adaptively according to the requirement of each speed when 60 nodes are moving at the same speed, as shown in Fig. 6-4.

Different broadcasting frequencies have been adopted in some references. Gavalas set the frequency of the periodic broadcasting of the algorithm, that is used for comparison with his adaptive broadcasting algorithm [72], as 200 times/second when nodes are moving with speeds of 0-15m/s. From this simulation, this high frequency of broadcasting is unnecessary.

First-node lifetime

The first-node lifetime of RBBPC, MSPB and MSSAB are shown in Fig. 6-5 with low traffic density and Fig. 6-6 with high traffic density.

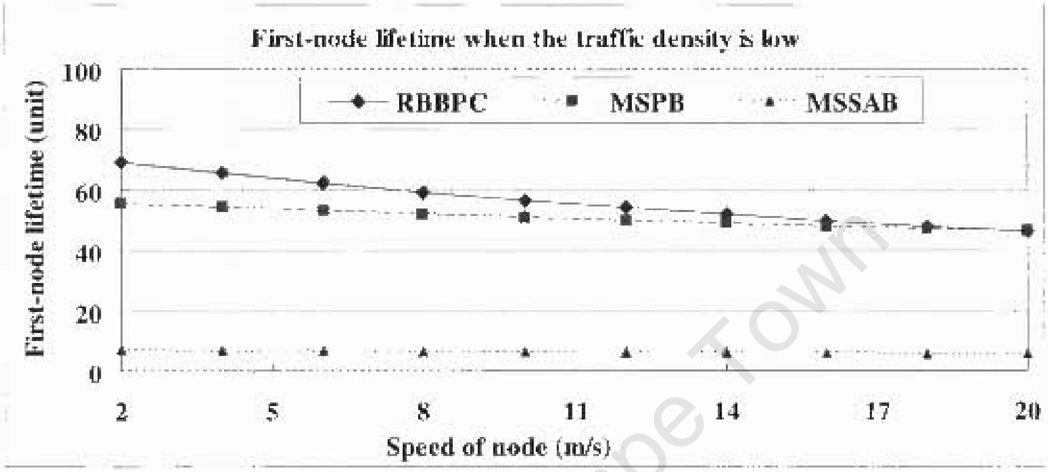


Figure 6-5. First-node lifetime of clustered MANET with low traffic density.

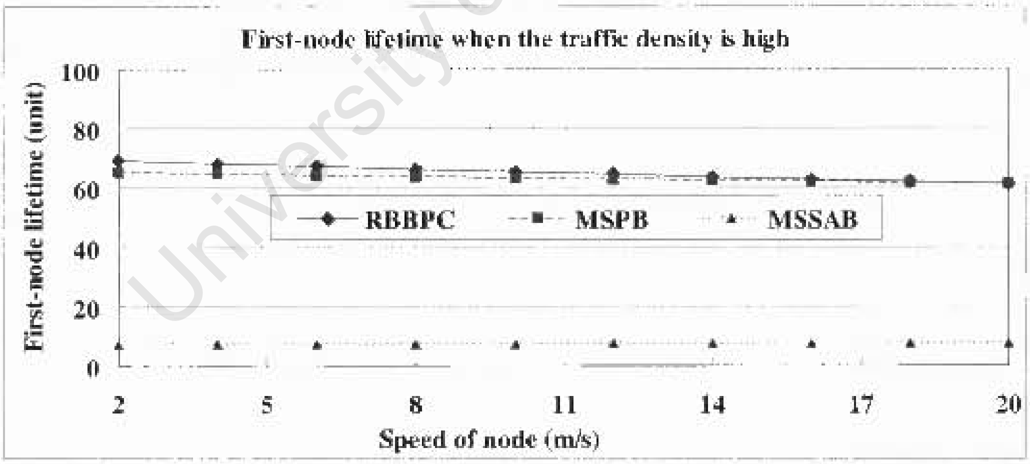


Figure 6-6. First-node lifetime of clustered MANET with high traffic density.

Fig. 6-5 shows the first-node lifetime when the traffic density is low (refer to Section 6.3.2). Low traffic density is defined as when each node only sends out one packet each 100s. When the nodes are moving at 2m/s in RBBPC, it is assumed that the power consumption of a member node is 0.1 units and of a CH is one unit (when the CH has one member node). When the nodes are moving faster, the power consumption of broadcasting, cluster maintenance, cluster size maintenance and inter-cluster routing discovery in RBBPC increases, the lifetime of the network therefore decreases. [Note that the specific increase in the power consumption when the nodes move faster can be found in Section 6.3.2 (Fig. 6-8)].

RBBPC has a longer first node lifetime than MSPB and MSSAB. When the speed is 2m/s, the ratio of lifetime between RBBPC, MSPB and MSSAB is 1: 0.1: 0.80. This improvement becomes less significant when the speed increases. When the speed of the nodes is 10m/s, this ratio becomes 1:0.11: 0.90 and when the speed of the nodes is 20m/s, RBBPC presents similar lifetime as MSPB. This is because the reduction in unnecessary broadcasting becomes less significant when the nodes are moving faster.

Though the control messages in MSPB are broadcast periodically, its lifetime also decreases according to the node speed. This is because when the node speed increases, the power consumption of updating intra-cluster routing information in MSPB increases. Thus, first-node lifetime decreases accordingly.

The first-node lifetime of MSSAB is quite interesting. It is expected to decrease according to the node speed, because when the nodes are moving faster, the power consumption of broadcasting, cluster maintenance and inter-cluster routing discovery increases. However, this phenomenon is barely presented in the results. This is because as the nodes move faster, they leave the original cluster and join other clusters more frequently. The greater speed can cause a cluster of greater size to reduce and a cluster of smaller size to increase in size in the next period. This improved balance in power consumption to a certain extent compensates the increase in power consumption caused by higher speed. MSSAB shows a far shorter first-node lifetime than do RBBPC and MSPB. This is because it maximizes the service time of the selected CHs. The CH rotation only occurs when any selected CH exhausts the energy.

Fig 6-6 shows the first-node lifetime when the traffic density is high (refer to Section 6.3.2). Low traffic density is defined as when each node sends out four packets each 100s.

When the nodes are moving at 2m/s in RBBPC, it is assumed that the power consumption of a member node is 0.1 units and of a CH is one unit (when the CH has one member node). Similar to the scenario of low traffic density, when the nodes move faster, the power consumption of the network increases. [Note that the specific increase in the power consumption when the nodes move faster can be found in Section 6.3.2 (Fig. 6-9)]. The improvement of RBBPC is less significant than that when the network has low traffic density. This is because, in a network with high traffic density, the power consumption of sending data packets becomes more significant and plays a major role in consuming energy, making the energy saved by reducing unnecessary broadcasting less significant.

The result of MSSAB is also quite interesting. It even increases slightly as nodes move faster. This is because the better balance in power consumption caused by nodes moving with higher speeds as stated becomes more significant than the increase in power consumption caused by higher speed.

In a realistic application, the nodes in a MANET usually do not consistently send data and usually move at a low speed. Therefore, the improvement of about 25% by RBBPC can be expected. If the data traffic becomes lower, more significant improvement is possible.

6.3.2 Mathematical analysis of power consumption

As stated, it is difficult to determine the power consumption of the nodes and of the network. This section therefore builds the relationship of power consumption between the algorithms by using mathematical analysis. This analysis uses the results from the simulations in Steps 1 and 2. The analysis results will be used to evaluate the first-node lifetime in Step 4 (refer to Fig. 6-1).

It is assumed that the total consumed energy E_{total} in a clustered MANET within a limited interval T is separated into five parts, as shown in (6.9) with the definition of each part described in Table 6-3.

$$E_{total} = E_{reclu-re} + E_{broadcasting} + E_{updaterouting} + E_{sizebalance} + E_{r-dis} + E_{d-comm} \tag{6.9}$$

The expected average power consumption $\overline{P}$ is then (6.10):

$$\overline{P} = E_{total} / T \tag{6.10}$$

Table 6-3. Definitions of the items in total power consumption.

Items	Definitions
$E_{reclu\text{-}re}$	Energy consumed by clustering organization with ripple effect when the network starts to work
$E_{broadcasting}$	Energy consumed by broadcasting to detect the changes of the nodes and the routes in the clusters
$E_{updaterouting}$	Energy consumed by broadcasting the updated intra-cluster routing information to the member nodes when re-clustering occurs
$E_{sizebalance}$	Energy consumed by balancing cluster sizes when the cluster sizes are not in the limited range
$E_{r\text{-}dis}$	Energy consumed by searching a backbone route for inter-cluster communication
$E_{d\text{-}comm}$	Energy consumed by delivering data packets from the source nodes to the destination nodes

The model shown in Fig. 6-7 is used for analysis. It is assumed that there are M nodes in the network, which are organized into N clusters. The size of the control message that is used for maintaining the clusters or inter-cluster routing discovery is a. The size of the data packet that a source node sends to a destination node is A.

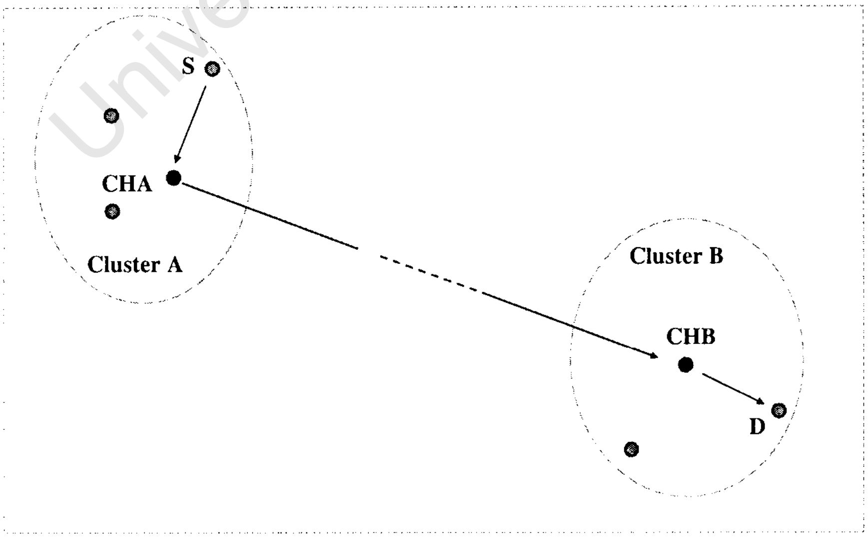


Figure 6-7. Model to analyze the power consumption of the network.

When the network starts to work, it needs to organize the nodes into clusters. It is assumed that the energy consumed by broadcasting each control message is $E_{conmess}$ (including the CH sends out the message and the member node responds to this message). Then the energy consumed by this clustering phase with resultant ripple effect is (6.11), in which *k1* is the coefficient of clustering communication complexity - how many rounds of message exchanges that the network needs to organize the clusters - of RBBPC:

$$E_{reclu-re} = k1 \times M \times E_{conmess} \tag{6.11}$$

After the clusters are formed, the network needs to maintain the clusters by broadcasting control message adaptively. The broadcasting frequency $f_{broadcasting}$ in this clustered MANET is (6.12), in which *k2* is a coefficient according to the node speed, which increases when the nodes are moving faster, and $f_{lowestspeed}$ is the broadcasting frequency of the lowest node speed.

$$f_{broadcasting} = k2 \times f_{lowestspeed} \tag{6.12}$$

Within interval *T*, the energy consumed by broadcasting control messages is then (6.13):

$$E_{broadcasting} = T \times k2 \times f_{lowestspeed} \times E_{conmess} \times M \tag{6.13}$$

Nodes leaving or joining the cluster are detected by the broadcasting control messages and the responses of these messages from the member nodes. When the CH finds that re-clustering occurs, it needs to broadcast the updated intra-cluster routing information to the member nodes. The size of this routing information is $k3 \times a$ (*k3* is a coefficient, which is relative to the number of nodes that the CH serves), because this message includes the routing information of the whole cluster so that its size is greater than that of the control message. The member nodes will confirm this reception. It is supposed that re-clustering occurs *P* times when the nodes are moving at the lowest speed. The energy consumed by updating intra-cluster routing information within interval *T* is (6.14):

$$E_{updaterouting} = k2 \times P \times E_{conmess} \times k3 \tag{6.14}$$

Sometimes, the cluster size will not be in the limited range which is pre-determined according to the node speed. The cluster size then needs to be adjusted. In RBBPC, the control messages are sent out adaptively to actually meet the requirements of different node speeds. Any changes of the cluster topology will be found out immediately, and only those that have sizes larger or smaller than the limitation will send node to or request node from other clusters. The communication complexity is therefore much less than that of the cluster organization phase when the network just starts to work. Thus no ripple effect occurs during the cluster maintenance phase. It is assumed that *L* times of size balance occur within this interval *T*. Then the energy consumed by balancing cluster size is (6.15), in which *k4* is the coefficient of the communication complexity – how many information exchanges occur when size balance happens.

$$E_{sizebalance} = L \times k4 \times E_{conmess} \tag{6.15}$$

RBBPC sets the relationship between the timer of the cached inter-cluster routing information and the period of broadcasting as (6.3). Wang assumes a relationship between broadcasting and timer as $T_{timer}=3\times T_{broadcasting}$ [149]. RBBPC considers a more realistic application and determines the coefficient to be the ratio between the transmission range of intra-cluster communication $R_{membernode}$ and of inter-cluster communication R_{CH} as shown in (6.16), in which f_{timer} is the frequency that the cached backbone routing information needs to be updated.

$$T_{timer} = \frac{R_{CH}}{R_{membernode}} \times T_{broadcasting} \Rightarrow f_{timer} = \frac{R_{membernode}}{R_{CH}} \times f_{broadcasting} \tag{6.16}$$

The energy of sending routing request E_{route} to the next hop is (6.17), in which $k5$ is a coefficient of power attenuation (refer to (4.1) in Section 4.2.1) in the energy model, of the ratio between the intra-cluster and inter-cluster communication ranges.

$$E_{roure} = \left(\frac{R_{CH}}{R_{membernode}} \right)^{k5} \times E_{conmess} \tag{6.17}$$

It is assumed that the total number of hops of routing discovery is *k6*. The energy consumed by inter-cluster routing discovery E_{r-dis} can be described as (6.18):

$$E_{r-dis} = T \times \frac{R_{membernode}}{R_{CH}} \times k2 \times f_{lowestspeed} \times k6 \times \left(\frac{R_{CH}}{R_{membernode}} \right)^{k5} \times E_{conmess} \tag{6.18}$$

During interval *T*, each node in the network sends *q* packets to a destination node, the size of each packet is *A*. The distance between the CH of the source node and the CH of the destination node is *k7* hops. The member node needs *k8* hops to access the CH. Then the energy consumed in delivering these packets is (6.19):

$$E_{d-comm} = M \times q \times \left(2 \times k8 \times \frac{A}{a} E_{conmess} + k7 \times \left(\frac{R_{CH}}{R_{membernode}} \right)^{k5} \times \frac{A}{a} \times E_{conmess} \right) \tag{6.19}$$

Therefore, the total consumed energy within interval *T* can be determined (6.20):

$$\begin{aligned} & E_{total} = E_{reclu-re} + E_{broadcasting} + E_{updaterouting} + E_{sizebalance} + E_{r-dis} + E_{d-comm} = \\ & k1 \times M \times E_{conmess} + T \times k2 \times f_{lowestspeed} \times E_{conmess} \times M + \\ & k2 \times P \times E_{conmess} \times k3 + L \times k4 \times E_{conmess} + \\ & T \times \frac{R_{membernode}}{R_{CH}} \times k2 \times f_{lowestspeed} \times k6 \times \left(\frac{R_{CH}}{R_{membernode}} \right)^{k5} \times E_{conmess} + \\ & M \times q \times \left(2 \times k8 \times \frac{A}{a} E_{conmess} + k7 \times \left(\frac{R_{CH}}{R_{membernode}} \right)^{k5} \times \frac{A}{a} \times E_{conmess} \right) \end{aligned} \tag{6.20}$$

The values assigned to the parameters are shown in Table 6-4, in which *P* and $f_{broadcasting}$ are from Steps 1 and 2.

Fig. 6-8 shows the results of power consumption of these three algorithms when each node sends out one packet within 100s. When the node speed is low, the improvement of RBBPC is more obvious. This is because that the unnecessary broadcasting to maintain the clusters and the unnecessary flooding to discover the inter-cluster route are reduced significantly. At 20m/s, however, the power consumption in RBBPC is slightly higher than that of the MSPB. This is because at 20m/s, the broadcasting frequencies of RBBPC and MSPB are the same. However, RBBPC needs extra power consumption to balance the cluster size and thus more energy is needed.

Table 6-4. Values assigned to the parameters.

T	M	N	$E_{conmess}$	a	A	$R_{membernodes}$
100 (s)	100	10	0.005 (unit)	25 bytes	250 bytes	200 (m)
R_{CH}	$k1$	$k2$	$k3$	$k4$	$k5$	$k6$
400 (m)	2	1-10	5	15	3	25
$k7$	$k8$	P	$f_{lowestspeed}$	q	L	
5	1.5	250	0.125 (at 2m/s)	1, 4	10	

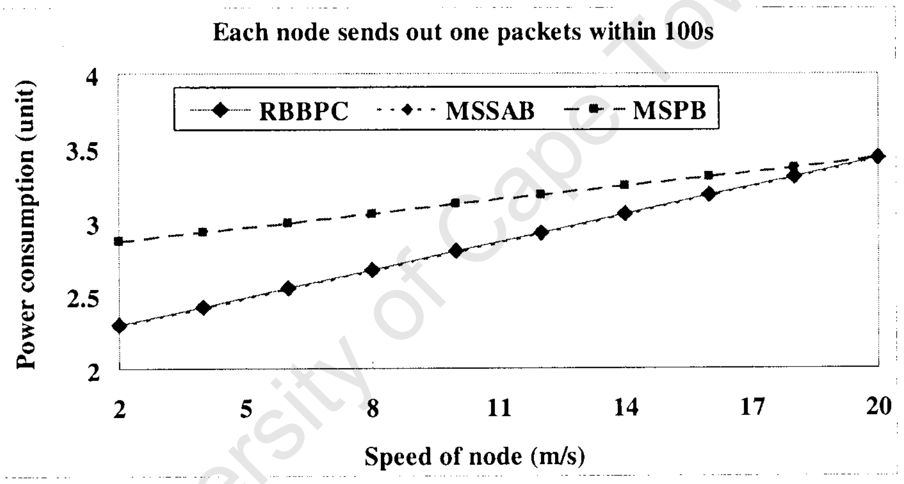


Figure 6-8. Power consumption when each node sends out one packet.

Fig. 6-9 shows the power consumption of these three algorithms when each node sends out four packets within 100s. It can be seen that the improvement of RBBPC is less significant compared to that when each node sends out one packet. This is because the power consumption of data communication plays a major role so that the improvement achieved by reducing unnecessary broadcasting becomes less significant.

Both Fig. 6-8 and Fig. 6-9 show that RBBPC and MSSAB have similar power consumption. This is because, similar to RBBPC, MSSAB also broadcasts control message adaptively. However, this similar result does not necessarily illustrate that

RBBPC and MSSAB have similar first-node lifetimes. Unlike MSSAB, RBBPC distributes the higher burden of the CH among the nodes and furthermore balances power consumption throughout the network. RBBPC thus has a much longer lifetime than does MSSAB (refer to Section 6.3.1).

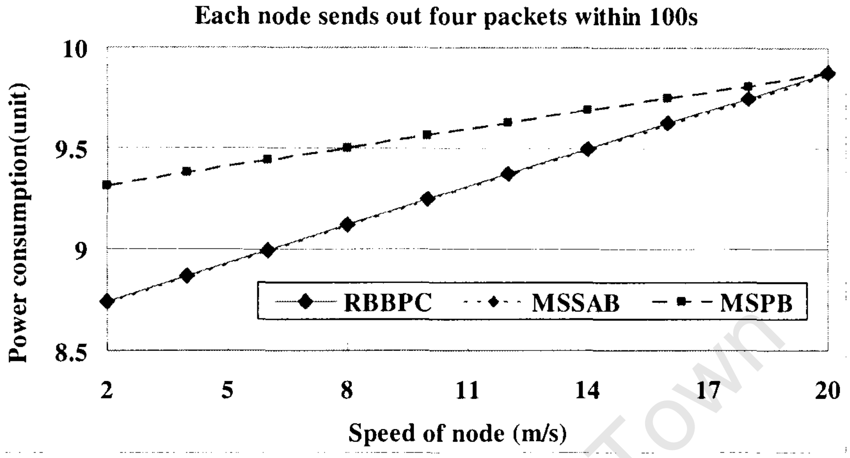


Figure 6-9.Power consumption when each node sends out four packets.

6.3.3 Inferences from the result

From the simulation results, the following inferences can be achieved:

The periodic broadcasting to maintain the network topology wastes energy of clustered MANETs. Broadcasting the control messages according to the node speeds reduces unnecessary broadcasting, thus improving energy efficiency. The improvement achieved becomes more significant when the nodes are moving at a low speed.

The movement of the nodes changes the cluster sizes from time to time, which alleviates the difference in cluster size, especially when the nodes are moving at a high speed.

6.4 Summary and discussions

6.4.1 Summary

An algorithm named RBBPC has been proposed in this chapter. This RBBPC broadcasts control messages adaptively according to various speeds, thereby reducing unnecessary broadcasting and improving energy efficiency of the network. Furthermore, it achieves the balance of cluster sizes adaptively according to the speed, thereby further balancing power consumption throughout the network. The performance evaluation by NS2 shows that (i) the control messages are really broadcast adaptively, (ii) energy efficiency is improved and (iii) the first-node lifetime is extended.

However, the proposed algorithm RBBPC and the performance evaluation only focus on energy related issues. Other questions like QoS, security, and data loss rate are beyond this research. Further work in improving the design can be undertaken by considering more related questions and by conducting more comprehensive performance evaluations.

6.4.2 Discussions

RBBPC has exempted energy-hungry CHs from relaying data. However, during the simulation, it is assumed that the selection of backbone route in RBBPC, MSPB and MSSAB, when inter-cluster communication occurs, is the same. Further research can be conducted to evaluate the routing performance of the inter-cluster communication.

7 Conclusion and future work

The emergence of ad hoc networks enables communication between wireless nodes by multi-hop routes more conveniently without any infrastructure. This technique, however, also has many challenges due to the infrastructureless topology structure, such as routing protocols, security, QoS, network scalability, energy efficiency and real-world implementation [150-158].

This thesis investigates energy efficiency in ad hoc networks. More specifically, it contributes to improving energy efficiency in clustered ad hoc networks, which is potentially one of the most promising topology structures for further application.

7.1 Summary of contributions

This dissertation addresses energy efficiency as one of the primary challenges in ad hoc networks. As WSNs and MANETs have different characteristics and different nodes, this dissertation treats these two types of networks separately. Based on the investigation of existing clustering algorithms for energy saving, different clustering algorithms have been proposed for WSNs and MANETs. The following summarizes the key contributions of this dissertation.

7.1.1 Clustering algorithm for WSNs (Chapters 4 and 5)

1. Balancing power consumption throughout the network in BRPC (Chapter 4)

The main contribution of the proposed clustering algorithm is to balance power consumption throughout the network. This is empirical work in the improvement of the energy efficiency by consideration of the directional data traffic towards the sink in a clustered homogeneous WSN in which the CHs send data to the sink over a multi-hop route. It builds the relationship between the data traffic and the power consumption so that the clusters are organized in such a way that their energy store is proportional to their respective power consumption. The cluster lifetimes are therefore equalized, thereby preventing premature node death caused by directional traffic. The simulation results show that this equalization not only improves the energy efficiency but also improves the performance of delivering data to the sink significantly.

2. Maintaining the CH in the central area of the cluster in BRPC (Chapter 4)

Some traditional clustering algorithms consider reducing power consumption of intra-cluster communication by locating a CH in the central area of the cluster. This localization is usually realized by re-clustering the nodes with ripple effect, but this fails to optimize energy efficiency. By contrast, the proposed algorithm realizes this localization by re-clustering without ripple effect. Furthermore, it maximizes the service time of the CH in the central area of the cluster and thus further improves energy efficiency.

3. Joint clustering and scheduling to balance power consumption in BPC (Chapter 5)

This dissertation also presents an algorithm to improve power consumption for WSNs with high node density. By joint clustering and scheduling, the power consumption is balanced throughout the network and the network is better covered. The simulation results show that both the energy efficiency and the performance of delivering data to the sink are significantly improved with the proposed BPC.

4. Centralized and distributed clustering algorithms (Chapters 4 and 5)

This dissertation considers a more realistic scenario in which the sink usually has sufficient resources and is the actual destination for all nodes in the network. The proposed algorithms therefore make better use the resources of the sink and combine the merits of centralized and distributed features. The proposed algorithms are distributed yet they are centralized to start cluster organization. This combined action makes equalizing cluster lifetime feasible and thus improves energy efficiency.

5. New methods to evaluate clustering algorithms (Chapters 4 and 5)

Most existing algorithms are evaluated by the use of a lifetime definition of (i) when any node dies (first-node lifetime) and/or (ii) when all nodes die (last-node lifetime). However, although the connectivity between the network and the data sink is usually not always lost after the first-node lifetime, it usually has been lost long before all nodes have died. This dissertation considers a more realistic scenario. First, it uses the first-node lifetime to measure the effective balance of power consumption throughout the network. Secondly, it uses a new method referred to as the connectivity lifetime to measure the period of the network life.

Besides the measurement of the lifetimes, it also defines the entire-network data (which are delivered to the sink before the first-node lifetime) and the partial-network data (which are delivered to the sink after the first-node lifetime). This is done to more reasonably evaluate the performance of the clustering algorithms on delivering data to the sink.

7.1.2 Clustering algorithm for MANETs (Chapter 6)

1. Adaptive broadcasting according to speeds

In a MANET, the traditional clustering algorithms broadcast control messages periodically to maintain the cluster. Typically, a higher node speed needs a higher broadcasting frequency. Therefore, this periodic broadcasting needs to meet the requirement of the potentially highest node speed to efficiently maintain the clusters. When the speed is low, the unnecessary broadcasting wastes network energy. The proposed algorithm RBBPC therefore broadcasts the control messages adaptively according to the node speeds. This reduces unnecessary broadcasting and results in higher energy efficiency. Furthermore, it optimizes the broadcasting frequency to make it actually meet the requirement of each speed. This optimized broadcasting frequency both improves energy efficiency and reduces the data loss rate.

2. Adaptive cluster-size balance according to speeds

Power consumption of the cluster, especially of the CH, is affected by the cluster size. Balancing cluster sizes equalizes power consumption among the clusters and CHs. However, if the cluster sizes are limited within a strict range, the frequent re-clustering triggered by the requirement of size maintenance will consume significant energy, thus defeating the advantage of balance in sizes. The proposed algorithm balances cluster sizes adaptively according to the node speed. When the speed is high, a more relaxed size range is applied whereas when it is low, a stricter size range becomes effective. This adaptive limitation makes the algorithm more suitable for the mobile environment and is more energy efficient.

3. New method to evaluate the clustering algorithm

This dissertation also proposes a new method to evaluate the first-node lifetime of a clustered MANET. There appears to be no literature or record of the first-node lifetime being evaluated in such a way. A WSN usually has stationary nodes and has a data sink. The first-node lifetime of a WSN can therefore be evaluated by all nodes sending data to the sink. However, a MANET does not have a sink and is formed by mobile nodes. Each node in a MANET can communicate with others freely. It is therefore difficult to determine how many nodes are sending data and to which destination nodes. This dissertation builds the relationship of the power consumption between the algorithms by mathematical analysis to make measurement of first-node lifetime easier.

7.2 Discussion and foreseen challenges

Although the simulation results show that all proposed clustering algorithm have (significantly) improved the energy efficiency of the network and thus extended lifetime, some future work is still necessary. This subsection summarizes some of the challenges foreseen.

7.2.1 Clustering algorithm for WSNs

The proposed BRPC in Chapter 4 assumes homogeneous WSNs with identical sensor nodes and with uniform node distribution. Sometimes, however, different nodes are uniformly distributed in a WSN. Though the discussion in Section 4.4.2 indicates that BRPC is also applicable in such a network, more detailed analysis and the extension of the algorithm can be conducted in future. Furthermore, some recent research shows the interest in WSNs in which the nodes can move freely [29-31]. Such a network combines the key challenges in WSNs and MANETs, which makes balancing power consumption and maintaining CH in the central area more challenging. Future extension of the algorithm may also consider such networks.

The proposed BPC in Chapter 5 is for homogeneous WSNs with higher node density. In this algorithm, as concluded in Section 5.5.2, the data are forced to be relayed by neighboring CHs nearer the sink. This forced relaying makes that the selected route may not be the most energy efficient one, thought the main objective of BPC to balance power consumption throughout the network is realized (refer to Section 4.3.3), future work can focus on achieving further improvement in energy efficiency by optimizing the route selection.

In a WSN, there are numerous issues that need to be taken into account. The proposed BRPC and BPC, however, only focus on energy efficiency. The conducted simulations also only evaluate the performance of energy related issues, such as different lifetimes and different data packets that the network delivers to the sink. Other challenges of sudden link failure, data re-transmission, QoS and security need to be further explored to make the proposed algorithm be more suitable for realistic application.

7.2.2 Clustering algorithm for MANETs

The proposed RBBPC for MANETs in Chapter 6 shows higher energy efficiency than those of some traditional algorithms from the simulation results and the mathematical analysis. The improvement of RBBPC for MANETs with low data traffic is significant. Though the low data traffic is the more realistic outcome as the nodes in the network usually do not keep sending data to others, some MANETs may experience busy data traffic. The simulation results and analysis show that the improvement of RBBPC for such a network is not significant. In a MANET with busy data traffic, most energy is consumed in routing data. Further work can be therefore conducted to optimize the route that is selected to relay data by considering the total energy of the route.

Similar to the proposed algorithms BRPC and BPC for WSNs, the proposed RBBPC for MANETs is based on energy efficiency consideration. The conducted simulations and analysis also only evaluate the performance of energy related issues, such as the power consumption in networks with different traffic density and the first-node lifetime. The issues like security, delay, network scalability and hidden-terminal are beyond the scope of this dissertation and will be taken into account in further research to make the proposed algorithm be more suitable for realistic application.

7.2.3 Implementation of the proposed algorithms

Only the simulations have been conducted so far during this research. The implementation of the proposed algorithms for WSNs and MANETs will be considered in future.

The algorithms BRPC and BPC for WSNs require the nodes to be aware of their location information. The implementation of these algorithms requires that the node to be supported by GPS technology or that each node can be carefully arranged before they are deployed into the network. The latter means that each node can get the relative location information so that the distance of the neighboring nodes can be easily acquired. The methods to acquire node location information in [98, 99], however, will be considered to incorporated into the proposed algorithms. Furthermore, the assumption made for BRPC and BPC that each node can adjust its transmission power in accordance with the necessary power for each hop may somehow be ahead of the realistic situation. However, many sensor nodes today may have several power levels. The power level which is most similar to yet higher than the necessary one can be selected in that case. This selection not only improves energy efficiency, but also avoids data loss caused by short of transmission range.

The algorithm RBBPC for MANETs only requires the nodes to have two power levels. Many mobile terminals today have more than one power level, making the implementation of RBBPC into MANETs quite convenient.

Besides the requirements of the hardware, other issues such as how to optimize the number of clusters in the network also need to be considered when such proposed algorithms are being implemented. This optimization is relative to both the transmission power of the nodes and the energy efficiency of the networks in WSNs or MANETs.

Bibliography

[1]. Fotini-Niovi Pavlidou, Mario Gerla and Werner Mohr, "Guest Editorial: Ad Hoc Networking," *International Journal of Wireless Information Networks*, 9(2): 71-73, April 2002.

[2]. Andrew T. Campbell, Marco Conti and Silvia Giordano, "Editorial: Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 8, pp. 483-484, 2003.

[3]. Imrich Chlamtac, Marco Conti and Jennifer J.-N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challenges," *Ad Hoc Networks*, vol. 1, pp.13-64, 2003.

[4]. Dali Wei and H Anthony Chan, "Analysis of the Applications and Characteristics of Ad Hoc Networks," *Proc. of IEEE ICCT 2006*, Guilin, China, Nov. 2006, pp. 881-884.

[5]. Jamal N. Al-Karaki and Ahmed E. Kamai, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Communications*, pp. 6-28, December 2004.

[6]. Jun Zheng, Pascal Lorenz and Petre Dini, "Guest Editorial: Wireless Sensor Networking," *IEEE Network*, pp.4-5, May 2006.

[7]. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey," *Computer Networks*, vol. 38, pp. 393-422, 2002.

[8]. Kemal Akkaya and Mohamed Younis, "A Survey on Routing Protocols for Wireless Sensor Networks," *Ad Hoc Networks*, vol. 3, pp. 325-349, 2005.

[9]. Mehmet Ulema, Jose Marcos Nogueira and Barcin Kozbe, "Guest Editorial: Management of Wireless Ad Hoc Networks and Wireless Sensor Networks," *Journal of Network and System Management*, 14(3): 327-333, September 2006.

[10]. Rüdiger Schollmeier, Ingo Gruber and Michael Finkenzeller, "Routing in Mobile Ad Hoc and Peer-to-peer Networks: A Comparison," *Lecture Notes in Computer Science*, vol. 2376, pp. 172-186, 2002.

[11]. J. Yu and P. Chong, "A Survey of Clustering Schemes for Mobile Ad Hoc Networks," *IEEE Communication Surveys*, 7(1): 32-48, March 2005.

[12]. Dali Wei and Anthony Chan, "Clustering Ad Hoc Networks: Schemes and Classifications," *Proc. of IEEE SECON'06* under the track of IEEE IWWAN 2006, New York, USA, June 2006, pp. 920-926.

[13]. W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan, "Energy-efficient Communication Protocol for Wireless Microsensor Networks," *Proc. of the 33rd IEEE Hawaii International Conference on System Science*, Jan. 2000.

[14]. Wendi Beth Heinzelman, *Application-Specific Protocol Architectures for Wireless Networks*, PhD Thesis, Massachusetts Institute of Technology, June 2000.

[15]. U. C. Kozat, George Kondylis, Bo Ryu and Mahesh K. Marina, “Virtual Dynamic Backbone for Mobile Ad Hoc Networks,” *Proc. of IEEE ICC’01*, Helsinki, Finland, June 2001, pp. 250-55.

[16]. A. B. MaDonald and T. F. Znati, “A Mobility-based Framework for Adaptive Clustering in Wireless Ad Hoc Networks,” *IEEE JSAC*, vol. 17, pp. 1466-1487, Aug. 1999.

[17]. A. Iwata, C. C. Chiang, G. Pei, M. Gerla and T. W. Cen, “Scalable Routing Strategies for Ad Hoc Wireless Networks,” *IEEE JSAC*, vol. 17, pp. 1369-1379, Aug. 1999.

[18]. A. Ozgur, O. Leveque and D. Tse., "Hierarchical Cooperation Achieves Linear Capacity Scaling in Ad Hoc Networks", *Proc. of IEEE INFOCOM 2007*, Anchorage, USA, May 2007, pp.382-390.

[19]. T.-C. Hou and T. J. Tsai, “An Access-based Clustering Protocol for Multihop Wireless Ad Hoc Networks,” *IEEE JSAC*, 19(7): 1201-10, July, 2001.

[20]. Mehran Abolhasan, Tadeusz Wysocki and Eryk Dutkiewicz, “A Review of Routing Protocols for Mobile Ad Hoc Networks,” *Ad Hoc Networks*, vol. 2, pp.1-22, 2004.

[21]. C.E. Perkins and T.J. Watson, “Highly Dynamic Destination Sequenced Distance Vector (DSDV) Routing for Mobile Computers,” *Proc. of ACM SIGCOMM Conference on Communications Architectures*, London, UK, Sep., 1994, pp. 234-244.

[22]. P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, “Optimized Link State routing Protocol for Ad Hoc Networks,” *Proc. of IEEE INMIC*, Pakistan, Dec. 2001, pp. 62-68.

[23]. D. Johnson, D. Maltz and J. Jetcheva, “The Dynamic Source Routing (DSR) Protocol for Mobile Ad Hoc Networks,” *IETF Internet Draft*, draft-ietf-manet-dsr-07.txt, work in progress, 2002.

[24]. S. Das, C. Perkins and E. Royer, “Ad Hoc on Demand Distance Vector (AODV) Routing,” *IETF Internet Draft*, draft-ietf-manetaodv-11.txt, work in progress, 2002.

[25]. Dali Wei, H Anthony Chan and Mohammed Bashier Rawoot, “Hybrid Routing Protocol to Decrease Delay and to Extend Lifetime for Mesh Networks,” *Information Technology Journal*, 6 (4): 518-525, 2007.

[26]. M. Dohler, D. Barthel, F. Maraninchi, L. Mounier, S. Aubert, C. Dugas, A. Buhrig, F. Paugnat, M. Renaudin, A. Duda, M. Heusse and F. Valois, “The ARESA Project: Facilitating Research, Development and Commercialization of WSNs,” *Proc. of IEEE SECON 2007*, San Diego, USA, June 2007, pp. 590-599.

[27]. N. Vlajic and D. Xia, “Wireless Sensor Networks: To Cluster or Not to Cluster?” *Proc. of WoWMoM’06*, Buffalo, USA, June 2006.

[28]. J. E. Wieselthier, G. D. Nguyen and A. Ephremides, “Algorithms for Energy-

efficient Multicasting in Static Ad Hoc Wireless Networks," *Mobile Networks and Applications*, vol. 6, pp. 251-263, 2001.

[29]. Nojeong Heo and Pramod K. Varshney, "Energy-efficient Deployment of Intelligent Mobile Sensor Networks," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 35(1): 78-92, Jan. 2005.

[30]. Xin Huang and Asuman Ozdaglar, "Power Control and Network Design in Mobile Sensor Networks," *Proc. of WIOPT 2006*, Boston, USA, April 2006.

[31]. Joengmin Hwang, H. C. Du and E. Kusmierek, "Energy Efficient Organization of Mobile Sensor Networks," *Pro. of ICPP 04*, Montreal, Canada, Aug. 2004, pp. 84-91.

[32]. Paolo Santi, "Topology Control in Wireless Ad Hoc and Sensor Networks," *ACM Computing Surveys*, 37(2): 164-194, June 2005.

[33]. Y. Thomas Hou, Yi Shi, Jianping Pan and Scott F. Midkiff, "Maximizing the Lifetime of Wireless Sensor Networks Through Optimal Single-session Flow Routing," *IEEE Transactions on mobile computing*, 5(9): 1255-1266, Sep. 2006.

[34]. S. D. Muruganathan, D. C. F. Ma, R. I. Bhasin and A. O. Fapojuwo, "A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks," *IEEE Radio Communications*, 43(3): S8-S13, March 2005.

[35]. Hao Chen and Seapahn Megerian, "Cluster Sizing and Head Selection for Efficient Data Aggregation and Routing in Sensor Networks," *Proc. of IEEE WCNC 2006,* Las Vegas, USA, March 2006, pp. 2318-2323.

[36]. J. S. Liu and C. H. Richard Lin, "Energy-Efficiency Clustering Protocol in Wireless Sensor Ntworks," *Ad Hoc Networks*, vol. 3, pp. 371-388, May 2005.

[37]. X. Du, Y. Xiao and F. Dai, "Increasing Network Lifetime by Balancing Node Energy Consumption in Heterogeneous Sensor Networks," Wiley Journal of *Wireless Communications and Mobile Computing (WCMC)*, 2006, DOI: 0.1002/wcm.452, available online (Sep. 2006).

[38]. Koustuv Dasgupta, Konstantinos Kalpakis and Parag Namjoshi, "An Efficient Clustering-Based Heuristic for Data Gathering and Aggregation in Sensor Networks," *Proc. of IEEE WCNC*, New Orleans, USA, March 2003, pp. 1948-1953.

[39]. B. Krishnamachari, D. Estrin and S. Wicker, "The Impact of Data Aggregation in Wireless Sensor Networks," *Proc. of Intl. Workshop on Distributed Event-Based Systems*, Vienna, Austria, July 2002.

[40]. Mao Ye, Chengfa Li, Guihai Chen and J. Wu, "EECS: An Energy Efficient Clustering Scheme in Wireless Sensor Networks," *Proc. of IPCCC 2005*, Phoenix, USA, April 2005, pp. 535-540.

[41]. Yuning He and Yongbing Zhang, "A New Energy Efficient Approach by Separating Data Collection and Data Report in Wireless Sensor Networks," *Proc. of*

ACM IWCMC'06, Vancouver, Canada, July 2006, pp. 1165-1170.

[42]. O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Transactions on Mobile Computing*, 3(4): 366-379, 2004.

[43]. O. Younis and S. Fahmy, "Distributed Clustering in Ad Hoc Sensor Networks: A hybrid, Energy-Efficient Approach," *Proc. of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[44]. G. Pei and C. Chien, "Low Power TDMA in Large Wireless Sensor Networks," *Proc. of IEEE MILCOM*, McLean, USA, October 2001, pp. 347-351.

[45]. Weixing Wang and Axel Jantsch, "An Algorithm for Electing Cluster Heads Based on Maximum Residual Energy," *Proc. of ACM IWCMC'06*, Vancouver, Canada, July 2006, pp. 1465-1470.

[46]. Zouhair El-Bazzal, Michel Kadoch, Asile L. Agba, Francois Gagnon and Maria Bennani, "An Efficient Management Algorithm for Clustering in Mobile Ad Hoc Network," *Proc. of PM^2HW^2N'06*, Malaga, Espagne, Oct. 2006, pp. 25-31.

[47]. S. Ghiasi, A. Srivastava, X. Yang and M. Sarrafzadeh, "Optimal Energy Aware Clustering in Sensor Networks," *MDPI Sensors*, pp.(2): 258-269, Feb. 2002.

[48]. S. Bandyopadhyay and E. J. Coyle, "An Energy Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks," *Proc. of IEEE INFOCOM*, Hyatt Regency, USA, March 2003, pp. 1713-1723.

[49]. R. Krishnan and D. Starobinski, "Efficient Clustering Algorithms for Self-Organizing Wireless Sensor Networks," *Ad Hoc Networks*, 4(1): 36-59, Jan. 2006.

[50]. L. Ramachandran, M. Kapoor, A. Sarkar and A. Aggarwal, "Clustering Algorithms for Wireless Ad Hoc Networks," *Proc. of the 4th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Boston, USA, August 2000, pp. 54-63.

[51]. Gaurav Gupta and Mohamed Younis, "Load-Balanced Clustering of Wireless Sensor Networks," *Proc. of IEEE ICC*, Anchorage, USA, May 2003, pp. 1848-1852.

[52]. K. S. Manousakis and J. S. Baras, "Clustering for Transmission Range Control and Connectivity Assurance for Self Configured Ad Hoc Networks," *Proc. of IEEE MILCOM*, Boston, USA, Oct. 2003, pp. 1042-1047.

[53]. V. Kawadia and P. R. Kumar, "Power Control and Clustering in Ad Hoc Networks," *Proc. of IEEE INFOCOM*, Hyatt Regency, USA, March 2003, pp. 459-469.

[54]. Dali Wei and H Anthony Chan, "Equalizing Cluster Lifetime for Sensor Networks with Directional Data Traffic to Improve Energy Efficiency," *Proc. of IEEE CCNC 2007*, Jan. 2007.

[55]. Dali Wei and H Anthony Chan, "Mesh Networking Backbone with Equalized Power Consumption Cells to Prolong the Lifetime of Large Sensor Networks," *Proc. of IEEE ICCT 2006*, Guilin, China, Nov. 2006, pp. 889-892.

[56]. Dali Wei, H Anthony Chan and Kevin V.N Kameri, "Circular-Layer Algorithm for Ad Hoc Sensor Networks to Balance Power Consumption" *Proc. of IEEE SECON'06* under the track of IEEE IWWAN 2006, New York, USA, June 2006, pp. 945-950.

[57]. Dali Wei and H. Anthony Chan, "Energy Efficient Dynamic Cluster Algorithm for Ad Hoc Sensor Networks," *Proc. of IEEE International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2006)*, Sydney, Australia, March 2006.

[58]. Dali Wei, H Anthony Chan and Benjamin Silombela, "A Low-Power Consumption Design of Ad Hoc Wireless Sensor Networks," *Proc. of the 12th International Conference on Telecommunications (ICT 2005)*, Cape Town, South Africa, May 2005.

[59]. Ming Ma and Yuanyuan Yang, "Clustering and Load Balancing in Hybrid Sensor Networks with Mobile Cluster Heads," *Proc. of QShine'06*, Waterloo, Canada, Aug. 2006.

[60]. Ching-Ju Lin, Po-Lin Chou and Cheng-Fu Chou, "HCDD: Hierarchical Cluster-based Data Dissemination in Wireless Sensor Networks with Mobile Sink," *Proc. of ACM IWCMC'06*, Vancouver, Canada, July 2006.

[61]. R. Zheng and R. Kravets, "On-demand Power Management for Ad Hoc Networks," *Proc. of IEEE INFOCOM'03*, , Hyatt Regency, USA, March 2003, vol. 1, pp. 481-491.

[62]. J. Gomez and A. T Campbell, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," *Wireless Networks*, vol. 9, pp. 443-60, 2003.

[63]. C.-K. Toh, "Maximum Battery Life Routing to Support Ubiquitous Mobile Computing in Wireless Ad Hoc Networks," *IEEE Communication Magazine*, 39(6): 138-147, June 2001.

[64]. Yi Xu and Wenye Wang, "MEACA: Mobility and Energy Aware Clustering Algorithm for Constructing Stable MANETs," *Proc. of IEEE MILCOM 2006,* Washington, USA, Oct. 2006.

[65]. Mainak Chatterjee, Sajal K. Das and Damla Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad Hoc Networks," *Cluster Computing*, vol. 5, pp. 193-204, 2002.

[66]. Alan D. Amis and Ravi Prakash, "Load-Balancing Clusters in Wireless Ad Hoc Networks," *Proc. of the 3rd IEEE Symposium on Application-Specific Systems and Software Engineering Technology (ASSET'00)*, Richardson, USA, March 2000, pp. 25-32.

[67]. F. Li, S. Z., X. Wang, X. Xue and H. Shen, "Vote-Based Clustering Algorithm in Mobile Ad Hoc Networks," *Proc. of International Conference on Networking Technologies for Broadband and Mobile Networks (ICOIN'2004)*, LNCS, Busan, Korea, Feb. 2004, vol. 3090, pp. 13-23.

[68]. D. Gavalas, G. Pantziou, C. Konstantopoulos and B. Mamalis, "Lowest-ID with Adaptive ID Reassignment: A Novel Mobile Ad-Hoc Network Clustering Algorithm," *Proc. of the 1st IEEE International Symposium on Wireless Pervasive Computing (ISWPC'2006)*, Phuket, Thailand, Jan. 2006.

[69]. S. Sivavakeesar, G. Pavlou and A. Liotta, "Stable Clustering Through Mobility Prediction for Large-Scale Multihop Ad Hoc Networks," *Proc. of IEEE WCNC 2004*, Atlanta, USA, March 2004, pp. 1488-1493.

[70]. C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks," *IEEE Journal of Selected Areas Communication*, 15(7): 1265-1275, Sep. 1997.

[71]. Y. Yi, M. Gerla and T. Kwon, "Efficient Flooding in Ad hoc Networks: a Comparative Performance Study," *Proc. of IEEE ICC 2003*, Anchorage, USA, May 2003, pp. 1059-1063.

[72]. Damianos Gavalas, Grammati Pantziou, Charalampos Konstantopoulos and Basilis Mamalis, "Clustering of Mobile Ad Hoc Networks: An Adaptive Broadcast Period Approach," *Prof. of IEEE ICC 2006,* Istanbul, Turkey, Jan. 2006, pp. 4034-4039.

[73]. T. Ohta, S. Inoue and Y. Kakuda, "An Adaptive Multihop Clustering Scheme for Highly Mobile Ad Hoc Networks," *Proc. of 6th ISADS'03*, Pisa, Italy, Apr. 2003, pp. 293-300.

[74]. J. Wu and H. L. Li, "On Calculating Connected Dominating Set for Efficient Routing in Ad Hoc Wireless Networks," *Proc. of 3rd International Workshop of Discrete Algorithms and Methods for Mobile Computation and Communication*, Washington, USA, Aug. 1999, pp. 7-14.

[75]. Marco Aurelio Spohn and J. J. Garcia-Luna-Aceves, "Bounded-distance Multi-clusterhead Formation in Wireless Ad Hoc Networks," *Ad Hoc Networks,* vol. 5, pp. 504-530, 2007.

[76]. Gayathri Venkataraman, Sabu Emmanuel and Srikanthan Thambipillai, "Size-restricted Cluster Formation and Cluster Maintenance Technique for Mobile Ad Hoc Networks," *International Journal of Network Management*, vol. 17, pp.171-194, 2007.

[77]. C. F. Chiasserini, I. Chlamtac, P. Monti and A. Nucci, "An Energy-Efficient Method for Nodes Assignment in Cluster-Based Ad Hoc Networks," *Wireless Networks,* vol. 10, pp. 223–231, 2004.

[78]. Jie Wu, Fei Dai, Ming Gao and Ivan Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks," *Journal of Communications and Networks*, 4(1): 1-12, March 2002.

[79]. Kai Liu, Jie Su, Jun Zhang, Feng Liu and Cheng Gong, "A Novel Stable Cluster Protocol for Mobile Ad Hoc Networks," *Proc. of 2005 IEEE International Symposium on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, Beijing, China, Aug. 2005, pp. 1328-1332.

[80]. P. Basu, N. Khan and T. D. C. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," *Proc. of IEEE ICDCSW'01*, Phoenix, USA, Apr. 2001, pp. 413-418.

[81]. S. Sivavakeesar, G. Pavlou, C. Bohoris and A. Liotta, "Effective Management through Prediction-Based Clustering Approach in the Next-generation Ad Hoc Networks," *Proc. of IEEE ICC*, Paris, France, June 2004, pp. 4326-4330.

[82]. T. J. Kwon and M. Gerla, "Efficient Flooding with Passive Clustering (PC) in Ad Hoc Networks," *Proc. of ACM SIGCOMM*, Pittsburgh, USA, Aug. 2002, pp. 44-56.

[83]. M. Gerla, T. J. Kwon and G. Pei, "On Demand Routing in Large Ad Hoc Wireless Network with Passive Clustering," *Proc. of IEEE WCNC*, Chicago, USA, Sep. 2000, pp. 23-28.

[84]. John Sucec and Ivan Marsic, "Clustering Overhead for Hierarchical Routing in Mobile Ad Hoc Networks," *Proc. of IEEE INFOCOM*, New York, June 2002, pp. 1698-1706.

[85]. A. B. McDonald and T. F. Znati, "Design and Performance of a Distributed Dynamic Clustering Algorithm for Ad-Hoc Networks," *Proc. of 34th Annual Simulation Symposium*, Los Angeles, USA, Apr. 2001, pp. 27-35.

[86]. Beongku An and Symeon Papavassiliou, "A Mobility-based Clustering Approach to Support Mobility Management and Multicast Routing in Mobile Ad-hoc Wireless Networks," *International Journal of Network Management*, pp 387-395, Nov. 2001.

[87]. Lyes Dekar and Hamamache Kheddouci, "A Cluster Based Mobility Prediction Scheme for Ad Hoc Networks," *Ad Hoc Network*, 6(2): 168-194, 2008.

[88]. Jung-hee Ryu, Sanghwa Song and Dong-Ho Cho, "New Clustering Schemes for Energy Conservation in Two-Tiered Mobile Ad-Hoc Networks," *Proc. of IEEE ICC*, Helsinki, Finland, June 2001, pp. 862-866.

[89]. Yaacov Fernandess and Dahlia Malkhi, "K-Clustering in Wireless Ad Hoc Networks," *Proc. of POMC'02*, Toulouse, France, Oct. 2002, pp. 31-37.

[90]. Supeng Leng, Liren Zhang and Yifan Chen, "K-hop Compound Metric Based Clustering Scheme for Ad Hoc Networks," *Proc. of IEEE ICC*, Seoul, Korea, May 2005, pp. 3396-3400.

[91]. C. Jayakumar, C. Chellappan, Gautam Das and Ved P. Gulati, "Associativity Based Mobility-adaptive K-clustering in Mobile Ad-hoc Networks," *Lecture Notes in Computer Science (LNCS)*, vol. 3356, pp. 160-168, 2004.

[92]. S. Yang, J. Wu and Jiannong Cao, "Connected K-hop Clustering in Ad Hoc

Networks," *Proc. of International Conference on Parallel Processing*, Univ. of Oslo, Norway, June 2005, pp. 373-380.

[93]. Giacomo Angione, Paolo Bellavista, Antonio Corradi and Eugenio Magistretti, "A K-hop Clustering Protocol for Dense Mobile Ad-hoc Networks," *Proc. of 26th IEEE International Conference on Distributed Computing Systems Workshops*, Lisboa, Portugal, July 2006.

[94]. Tsung-Chuan Huang, Liang-Chen Shiu and Han-Chun Ke, "A Double-Manager K-hop Clustering Algorithm in Mobile Ad Hoc Networks," *Proc. of the 4th International Conference on Computer and Information Technology*, Wuhan, China, Sep. 2004, pp. 640-645.

[95]. Wei Lou and Jie Wu, "A K-hop Zone-based Broadcast Protocol in Mobile Ad Hoc Networks," *Proc. of IEEE GLOBECOM*, Dallas, USA, Dec. 2004, pp. 1665-1669.

[96]. Dali Wei and H Anthony Chan, "Clustering Algorithm to Balance and to Reduce Power Consumptions for Homogeneous Sensor Networks," *Proc. of IEEE WiCom 2007*, Shanghai, China, Sep. 2007, pp. 2723-2726.

[97]. Dali Wei and H Anthony Chan, "Joint Scheduling and Clustering to Balance Power Consumption for High-Node-Density Sensor Networks with Directional Data Traffic," *Proc. of IEEE WiCom 2007*, Shanghai, China, Sep. 2007, pp. 2727-2730.

[98]. T. Watteyne, I. Augé-Blum, M. Dohler and D. Barthel, "Geographic Forwarding in Wireless Sensor Networks with Loose Position-Awareness," *Proc. of IEEE PIMRC 2007*, Athens, Greece, Sep. 2007.

[99]. T. Watteyne, D. Simplot-Ryl, I. Augé-Blum and M. Dohler, "On Using Virtual Coordinates for Routing in the Context of Wireless Sensor Networks," *Proc. of IEEE PIMRC 2007*, Athens, Greece, Sep. 2007.

[100]. C. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," *Proc. of IEEE WMCSA'99*, New Orleans, USA, Feb. 1999, pp. 90-100.

[101]. D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," T. Imielinski and H. Korth, Editors, *Mobile Computing*, Kluwer Academic Publishers, ch. 5, pp. 153-181, 1996.

[102]. Zhao Cheng and Wendi B. Heinzelman, "Exploring long lifetime routing (LLR) in ad hoc networks," *Proc. of MSWiM 04*, Venice, Italy, Oct. 2004, pp. 203-210.

[103]. S. M. Senouci and G. Pujolle, "Energy Efficient Routing in Wireless Ad Hoc Networks," *Proc. of IEEE ICC*, Paris, France, June 2004, pp. 4057 - 4061.

[104]. Danjie Chen and Philip J. Lin, "Minimum Energy Path Planning for Ad Hoc Networks," *Proc. of IEEE WCNC*, Las Vegas, USA, March 2006, pp. 288-293.

[105]. S. C. Wang, D. S. L. Wei and S. Y. Kuo, "SPT-based Power-efficient Topology Control for Wireless Ad Hoc Networks," *Proc. of IEEE MILCOM*, Monterey, USA, Nov. 2004, pp. 1483-1490.

[106]. Li (Erran) Li and Joseph Y. Halpern, "A Minimum-energy Path-Preserving Topology-Control Algorithm," *IEEE Transactions on wireless communications*, 3(3): 910-921, May 2004.

[107]. S. Doshi, S. Bhandare and T. X. Brown, "An on-demand Minimum Energy Routing Protocol for a Wireless Ad Hoc Network," *Mobile Commuting and Communications Reviews*, 6(3): 50-66, March 2002.

[108]. Rajit Manohar and Anna Scaglione, "Power Optimal Routing in Wireless Networks," *Proc. of IEEE ICC*, Seattle, USA, May 2003, pp. 2979 - 2984.

[109]. J. Ye, K. C. Chua and W. C. Wong, "Power Conservation and Path Efficiency for Multicast Ad Hoc Networks," *Proc. of IEEE ICC*, Paris, France, June 2004, pp. 3623 - 3627.

[110]. R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks Using Transmit Power Adjustment," *Proc. of IEEE INFOCOM*, Tel-Aviv, Israel, March 2000, pp. 404-413.

[111]. M. Kubisch, H. Kark, A. Wolisz, L. C. Zhong and J. Rabaey, "Distributed Algorithms for Transmission Power Control in Wireless Sensor Networks," *Proc. of IEEE WCNC*, New Orleans, USA, March 2003, pp. 558 - 563.

[112]. P. P. Bergamo, A. Giovanardi, A. Travasoni, D. Maniezzo, G. Mazzini and M. Zorzi, "Distributed Power Control for Energy Efficient Routing in Ad Hoc Networks," *Wireless Networks 10*, pp. 29-42, 2004.

[113]. J. H. Chang and L. Tassiulas, "Maximum Lifetime Routing in Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, 12(4): 609-619, August 2004.

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

[115]. Q. Gao, K. J. Blow, D. J. Holding, I. W. Marshall and X. H. Peng, "Radio Range Adjustment for Energy Efficient Wireless Sensor Networks," *Ad Hoc Networks*, 4(1): 75-82January 2006.

[116]. J. Gomez, A. T. Campbell, M. Naghshineh and C. Bisdikian, "PARO: Supporting Dynamic Power Controlled Routing in Wireless Ad Hoc Networks," *Wireless Networks.* vol. 9, pp 443-460, 2003.

[117]. Eun-Sun Jung and Nitin H. Vaidya, "Power Aware Routing Using Power Control in Ad Hoc Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, 9(3): 7-18 , July 2005.

[118]. Dali Wei and H Anthony Chan, "Route Optimization for Energy Conservation in Ad Hoc Networks," *Proc. of IEEE ICCT 2006*, Guilin, China, Nov. 2006, pp. 885-888.

[119]. Dali Wei, H Anthony Chan and Benjamin Silombela, "Rectangular Grids Design to Balance Power Consumption for Homogeneous Sensor Networks with High Node

Density," *Information Technology Journal,* 6(6): 827-834, 2007.

[120]. Dali Wei and H Anthony Chan, "An Efficient Clustering Algorithm for Topology Maintenance and Energy Saving in MANETs," *Proc. of South Africa Telecommunications and Networks Application (SATNAC 2006),* Cape Town, South Africa, Sep. 2006.

[121]. A. Bachir and D. Barthel, "Localized max-min remaining energy routing for WSN using delay control," *Proc. of IEEE ICC 2005*, Seoul, Korea, May 2005, pp. 3302-3306.

[122]. Y. Xu, J. Heidemann and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing," *Proc. of ACMMobile'01,* Rome, Italy, July 2001, pp 70-84.

[123]. J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney, "Scheduling Sleeping Nodes in High Density Cluster-based Sensor Networks," *ACM/Kluwer Mobile Networks and Applications*, 10(6): 825-835, Dec. 2005.

[124]. Lan Wang and Yang Xiao, "A Survey of Energy-Efficient Scheduling Mechanisms in Sensor Networks," *Mobile Networks and Applications,* vol. 11, pp. 723-740, 2006.

[125]. Sunil Kumar, Vineet S. Raghavan and Jing Deng, "Medium Access Control protocols for ad hoc wireless networks: A survey," *Ad Hoc Networks,* vol. 4, pp. 326-358, 2006.

[126]. S. Singh and C.S. Raghavendra, "PAMAS-Power Aware Multi-Access Protocol with Signalling for Ad Hoc Networks," *ACM Computer Communication Review,* 28(3): 5-26, 1998.

[127]. W. Ye, J. Heidemann and D. Estrin, "An Energy-efficient MAC Protocol for Wireless Sensor Networks," *Proc. of IEEE INFOCOM*, New York, USA, June 2002, vol. 3, pp.1567-1576.

[128]. Wei Ye, John Heidemann and Deborah Estrin, "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks," *IEEE/ACM Transactions on Networking*, 12(3): 493-506, June 2004.

[129]. T. Van Dam and K. Langendoen, "An Adaptive Energy-efficient MAC Protocol for Wireless Sensor Networks," *Proc. of the 1st International Conference on Embedded Networked Sensor Systems (SenSys '03),* Los Angles, USA, Nov. 2003, pp.171-180.

[130]. K. R. Chowdhury, N. Nandiraju, D. Cavalcanti and D. P. Agrawal, "CMAC-A Multi-Channel Energy Efficient MAC for Wireless Sensor Networks," *Proc. of IEEE WCNC*, Las Vegas, USA, March 2006, pp.1172-1177.

[131]. V. Rajendran, K. Obraczka and J. J. Garcia-Luna-Aceves, "Energy-efficient, Collision-free Medium Access Control for Wireless Sensor Networks," *Proc. of the 1st International Conference on embedded Networked Sensor Systems (ACM SenSys '03),* Los Angeles, USA, Nov. 2003, pp. 181-192.

[132]. M. J. Miller and N. H. Vaidya, "On-Demand TDMA Scheduling for Energy Conservation in Sensor Networks," *Technology report of University of Illinois at Urbana-Champaign*, 2004.

[133]. Rajgopal Kannan, Ram Kalidindi and S. S. Iyengar, "Energy and Rate based MAC Protocol for Wireless Sensor Networks," *SIGMOD Record*, 32(4): 60-65, 2003.

[134]. M. L. Sichitiu, "Cross-Layer Scheduling for Power Efficiency in Wireless Sensor Networks," *Proc. of IEEE INFOCOM*, Hong Kong, China, March 2004, pp. 1740-1750.

[135]. F. Koushanfar, A. Davare, D. T. Nguyen, M. Potkonjak and A. S. Vincentelli, "Low Power Coordination in Wireless Ad-hoc Networks," *Proc. of International Symposium on Low Power Electronics and Design*, Seoul, Korea, Aug. 2003, pp. 475-480.

[136]. A. Cerpa and D. Estrin, "ASCENT: Adaptive Self- configuring Sensor Networks Topologies," *Proc. of IEEE INFOCOM*, New York, USA, June 2002, pp. 1278-1287.

[137]. B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "SPAN: An Energy-Efficient Coordination Algorithm for Topology Maintenance in ad hoc Wireless Networks," *Wireless Networks*, vol. 8, pp. 481-494, 2004.

[138]. Y. Xu, J. Heidemann and D. Estrin, "Adaptive Energy-conserving Routing for Multihop Ad Hoc Networks," *Technical report 527*, USC/ISI, 2000.

[139]. C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastave, "STEM: Topology Management for Sensor Networks," *Proc. of ACM MOBIHOC*, Lausanne, Switzerland, June 2002, pp. 135-145.

[140]. M. J. Miller and N. H. Vaidya, "A MAC Protocol to Reduce Sensor Network Energy Consumption Using a Wakeup Radio," *IEEE Transactions on Mobile Computing*, 4(3): 228-242, May 2005.

[141]. J. Deng, Y. S. Han, W. B. Heinzelman and P. K. Varshney, "Balanced-energy Sleep Scheduling Scheme for High Density Cluster-based Sensor Networks," *Elsevier Computer Communications Journal, Special Issue on ASWN '04*, 28(14): 1631-1642, Sep. 2005.

[142]. T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Networks Research," Wireless Communication and *Mobile Computing (WCMC)*: special issue in Mobile Ad Hoc Networking: Research, Trends and Applications, 2(5): 483-502, 2002.

[143]. W. Su, S.-J. Lee and M. Gerla, "Mobility Prediction and Routing in Ad Hoc Wireless Networks," *International Journal of Network Management*, 11 (1): 3-30, 2001.

[144]. Werner Creixell and Kaoru Sezaki, "Routing Protocol for Ad Hoc Mobile Networks Using Mobility Prediction," *International Journal of Ad Hoc and Ubiquitous*

Computing, 2(3): 149-156, 2007.

[145]. R. Chellappa Doss, A. Jennings and N. Shenoy, "A Review of Current Mobility Prediction Techniques for Ad Hoc Networks," *Proc. of 4th IASTED International Multi-conference*, Banff, Canada, July 2004, pp 536-542.

[146]. K. Wang and B. Li, "Group Mobility and Partition Prediction in Wireless Ad-Hoc Networks," *Proc. of IEEE ICC 2002*, New York, April 2002, pp. 1017-1021.

[147]. Dali Wei, H Anthony Chan, Engelbert Linus Chuwa and Batebe Laura Majugo "Mobility-Sensitive Clustering Algorithm to Balance Power Consumptions in Mobile Ad Hoc Networks," *Proc. of IEEE WiCom 2007*, Shanghai, China, Sep. 2007, pp. 1645-1648.

[148]. Xiaoguang Niu, Zhihua Tao, Gongyi Wu, Changcheng Huang and Li Cui, "Hybrid Cluster Routing: An Efficient Routing Protocol for Mobile Ad Hoc Networks," *Proc. of IEEE ICC 2006*, Istanbul, Turkey, June 2006, pp. 3554-3559.

[149]. Hong-bo Wand, Yao-xue Zhang, Xiao-hui Wang and Guo-qiang Guo, "A Suitable Size Clustering Algorithm for Ad Hoc Wireless Networks," *Journal of Software*, 13(9): 1741-1756, 2002.

[150]. D. Djenouri, L. Khelladi and N. Badache, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks," *IEEE Communication Surveys*, 7(4): 2-28, 2005.

[151]. Wolfgang Kiess and Martin Mauve, "A Survey on Real-world Implementations of Mobile Ad-hoc Networks," *Ad Hoc Networks*, 5(3): 324-339, April 2007.

[152]. Baoxian Zhang and Hussein T. Mouftah, "QoS Routing for Wireless Ad Hoc Networks: Problems, Algorithms, and Protocol," *IEEE Communication Magazine*, pp. 110-117, Oct. 2006.

[153]. T. Bheemarjuna Reddy, I Karthigeyan, B. S. Manoj and C. Siva Ram Murthy, "Quality of Service Provisioning in Ad Hoc Wireless Networks: A Survey of Issues and Solutions," *Ad Hoc Networks*, 4(1): 83-124, Jan. 2006.

[154]. Patroklos G. Argyroudis and Donal O'Mahony, "Secure Routing for Mobile Ad Hoc Networks," *IEEE Communication Survey*, 7(3): 2-21, 2005.

[155]. Siddhartha Gupte and Mukesh Singhal, "Secure Routing in Mobile Ad Hoc Networks," *AD Hoc Networks*, vol. 1, pp. 151-174, 2003.

[156]. Fei Hu and Neeraj K. Sharma, "Security Considerations in Ad Hoc Sensor Networks," *Ad Hoc Networks*, vol. 3, pp. 69-89, 2005.

[157]. Hao Yang and Haiyun Luo, Fan Ye, Songwu Lu and Lixia Zhang, "Security in Mobile Ad Hoc Networks Challenges and Solutions," *IEEE Wireless Communication*, pp. 38-47, Feb. 2004.

[158]. Ian F. Akyildiz and Ismail H. Kasimoglu, "Wireless Sensor and Actor Networks: Research Challenges," *Ad Hoc Networks*, vol. 2, pp. 351-367, 2005.