

**The Highly Reliable and Load-balancing Routing Protocol for the Wireless  
Sensor Network**

by

Jung Hoon, Lee

A thesis submitted to the Graduate Faculty of  
Auburn University  
in partial fulfillment of the  
requirements for the Degree of  
Master of Science

Auburn, Alabama  
August 9, 2010

Keywords: Wireless sensor network, load-balancing, multipath

Copyright 2010 by Jung Hoon, Lee

Approved by

Alvin Lim, Chair, Associate Professor of Computer Science and Software Engineering  
Drew Hamilton, Professor of Computer Science and Software Engineering  
Wei-Shinn Ku, Assistant Professor of Computer Science and Software Engineering

## Abstract

In wireless sensor networks, the reliability of message delivery is low because its wireless medium has a high packet loss rate and some nodes frequently may not work because each node uses a small amount of battery power. In addition, if single path routing protocol is used, the load is placed on some particular node, causing congestion. As a result, this degrades the overall network performance, as well as increases delays.

In contrast, if the routing protocol provides multiple paths, not only does it guarantee fault tolerance, but also improves reliability by broadcasting messages using multi paths. In addition, by distributing the load that is being focused on a particular node, it reduces incidence of congestion and increases the network's life time and reliability. Therefore, multi-path routing is one of several techniques being studied to solve the issue in the wireless sensor network.

However, the multi-path routing protocols that have been proposed so far either have too much message overhead for finding the multi-path or use a less efficient multi path to reduce this overhead. In addition, as the number of nodes in the network grows, the routing table size is enlarged; making it unsuitable for sensor nodes with a small amount of memory.

This thesis proposes a highly reliable and load-balancing routing protocol for the wireless sensor network through finding multiple node-disjoint paths and through removing the intermediate node's routing table overhead by computing the entire path from the sink to the destination.

## Acknowledgments

The graduate years that had started in front of an empty laboratory desk, with a mixture of fears and flutters, is now drawing to a close. During that period spent within the lab-studying in midst all the crying, laughing and being tired-has slowly accumulated to become, what is now a valuable memory.

I could never have completed this work without supports and assistances from many people. First and foremost, I would like to send my thanks to my family members-to my parents who'd quietly encouraged me through my graduate years and to my wife, EunSuk who'd allowed me to devote myself entirely to my studies for the duration of past two years-because they've been most supportive so that I could become the person I am now.

I would also like to thank my professor, Dr. Alvin Lim who've always, with utmost sincerity, encouraged and guided me through the difficult times, as well as the committee member professors, Dr Hamilton and Dr. Jeff that have provided me with guidance on my thesis.

The end of one road signifies the start of another. No matter what kind of a road lies ahead of me after my graduate school years, I am not afraid because I have individuals that provide me with the strength I need to persevere. I want to use this knowledge I've attained as a stepping stone so I may become the salt of the world.

Lastly, I want to become a guiding light that guides my son, whom is the apple of my eyes, my life's most precious treasure, in his future.

Thank you, to everyone.

*A cold head and a warm heart.*

- James Lee

## Table of Contents

Abstract . . . . .	ii
Acknowledgments . . . . .	iii
List of Figures . . . . .	vi
List of Tables . . . . .	viii
1 Introduction . . . . .	1
1.1 The configuration of wireless sensor networks . . . . .	1
1.2 Characteristics of wireless sensor networks . . . . .	2
2 Motivation and Objective . . . . .	5
2.1 Motivation . . . . .	5
2.2 Problem statement . . . . .	6
2.3 Approach to solve problem . . . . .	7
2.4 Objective . . . . .	7
3 Related Work . . . . .	9
3.1 The difference between Wireless ad-hoc routing protocols and sensor network routing protocols . . . . .	9
3.2 Multipath routing protocol for sensor network . . . . .	10
4 Protocol Design . . . . .	13
4.1 Building network topology considering the hop count to the sink node . . . . .	13
4.2 Route discovery using the priority, shortest hop number and source routing . . . . .	19
4.2.1 Data propagation from the sink node to the sensor node . . . . .	19
4.2.2 Data propagation from the sensor node to the sink node . . . . .	21
4.3 Route Maintenance . . . . .	24
4.3.1 From sink node to destination . . . . .	24

4.3.2	From sensor to the sink . . . . .	27
5	Performance Evaluation . . . . .	31
5.1	The goal of performance evaluation . . . . .	31
5.2	Simulation methodology . . . . .	31
5.3	Performance evaluation metric . . . . .	32
5.4	Performance comparison . . . . .	34
5.4.1	Throughput . . . . .	34
5.4.2	End to end delay . . . . .	35
5.4.3	The number of node-disjoint multipath and Multipath ratio . . . . .	36
5.4.4	Routing table size . . . . .	39
5.4.5	The effect of the uplink neighbor number . . . . .	40
5.4.6	The effect of path number . . . . .	41
6	Conclusion . . . . .	43
	Bibliography . . . . .	46

## List of Figures

1.1	Wireless sensor network . . . . .	2
4.1	BNTQ process . . . . .	14
4.2	BNTQ broadcasting using hop count . . . . .	15
4.3	BNTQ error process . . . . .	18
4.4	Example of multipath selection . . . . .	21
4.5	Uplink path node selection process . . . . .	23
4.6	Downlink route maintenance . . . . .	26
4.7	uplink route maintenance . . . . .	29
5.1	Throughput of EMRP, DCHT, and Directed Diffusion with different network sizes	34
5.2	End-to-end delay of EMRP, DCHT, and Directed Diffusion with different network sizes . . . . .	35
5.3	Multipath number from sink node to source node with different network sizes . . . . .	36
5.4	Count example 1 of multipath number from sink node to each source node . . . . .	37
5.5	Count example 2 of multipath number from sink node to each source node2 . . . . .	37
5.6	Ratio of Multipath number to network sizes . . . . .	38

5.7 Routing table size of sink and each sensor nodes with different network sizes . . . 39

5.8 Multipath number with different uplink neighbor node nubmer that sensor node  
can maintain . . . . . 40

5.9 Packet delivery ratio of EMRP with different path number. The network size is  
116 nodes . . . . . 41

5.10 Packet loss rate of EMRP with with different path number. The network size is  
116 nodes. Data rate is 50 packets/sec . . . . . 42

## List of Tables

3.1	Comparison of routing protocol . . . . .	9
4.1	The Hop count of each node . . . . .	15
4.2	Sink's routing table based on Figure 4.2 . . . . .	16
4.3	Sink's routing table based on Figure 4.3 . . . . .	17
4.4	The order of queue's extension . . . . .	21
5.1	The simulation methodology . . . . .	32
5.2	The performance evaluation metrics . . . . .	33



## Chapter 1

### Introduction

With recent developments in wireless technology and hardware technology, the development of small, low-power sensor nodes have increased. In addition, a wireless sensor network research using these sensor nodes is actively being conducted. The advancements in wireless sensor network technologies introduced networking capabilities to all areas of human life, beyond just the scope of the computer network; and is becoming a key technology to realize a ubiquitous network era.

#### 1.1 The configuration of wireless sensor networks

A wireless sensor network is a set of wireless sensor nodes organizing a network by itself without centralized a communication infrastructure. Each sensor node in a wireless sensor network has several functions, which are information collection, information processing, storage, and wireless communication. A wireless sensor network is made up of multiple sensor nodes and a few sink nodes. The sensor nodes collect data and then transmit that data in the sensor field. Sink nodes process the information that they receive from sensor nodes as well as make the information accessible to users[1].

Figure 1.1 shows the wireless sensor network [2]. A sensor node is usually deployed in difficult or dangerous places where humans are active and collects interest data. The collected data will be sent to the sink node. With these structures, the user of the wireless sensor network can pass a query to the sensor nodes through the sink, or receive the collected data from sensor nodes.

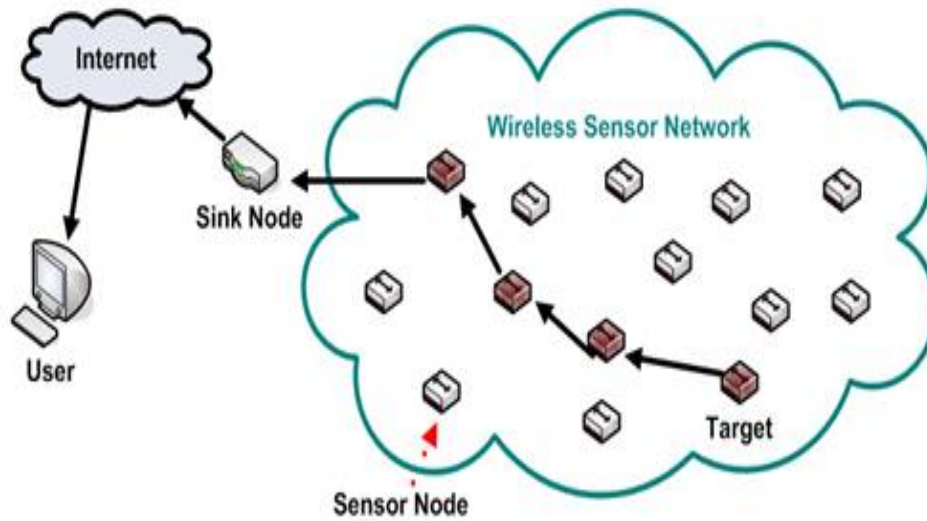


Figure 1.1: Wireless sensor network

## 1.2 Characteristics of wireless sensor networks

A wireless sensor network is a self-deployable network of many sensor nodes in an environment that has no existing communication infrastructure. Even if low-power sensor nodes were developed, each sensor node has limited information processing and storage capabilities, providing a relatively short distance for communication. On the other hand, the sink node is free from such restrictions. When compared to the sensor nodes, the sink node has a longer central processing unit (CPU), memory devices, and larger capacity, as well as a power supply that can be sustained. In addition, depending on user's needs, a sink node can connect to another sink node, or to a wired connection that has the infrastructure network (such as the internet).

Therefore, the deployment of the sink node is limited; and most of the sensor nodes have no sink node within their transmission range. A wireless sensor network that consists of these nodes has the following characteristics:

- 1) Consists of compact, low-cost sensor nodes, and uses a limited computing power and battery-based energy

2) Can deploy a number of nodes without a predetermined configuration in the region of interest

3) Precision deployment of the nodes; so the overall operation of the network is not affected because the neighbor sensor nodes detect similar information (even if the behavior of any sensor node fails or is destroyed)

4) Problems such as late communication speed of the wireless medium, heavy transmission error characteristics, a limited power supply, and almost impossible replacement of the sensor nodes due to random deployment

Based on these characteristics, the following considerations are important when a wireless sensor network protocol is designed.

1) Increase the overall life of the network by distributing energy consumption to the entire network

2) Quick response to sudden environmental changes

3) Self-organizing and wireless multi-hop routing through cooperation between the nodes

To meet these requirements, many sensor network routing protocols were proposed; and these protocols will be introduced in the related work section.

If the routing protocol provides multi-paths, a high level of fault tolerance can be guaranteed. However, the more control message overhead needed to find multiple paths, the more energy is consumed. Also, routing table size increases in proportion to the number of nodes in most of the multi-path routing protocol.

Therefore, this thesis proposes an efficient routing protocol for the wireless sensor network through finding multiple paths and by removing the intermediate node's routing table overhead by calculating the entire path from the sink to the destination.

This thesis is organized as follows: In section 2, the motivations and objectives for the proposed protocol for wireless sensor networking are discussed. In section 3, related work is discussed. In section 4, an efficient multi-path routing protocol which reduces routing

overhead and improves packet delivery ratios is proposed. In section 5, the performance of the proposed protocol is described and the results using an ns-2 simulator are evaluated [3]. Finally, a conclusion to this thesis comes in section 6.

## Chapter 2

### Motivation and Objective

The motivations and objectives for the proposed protocol for wireless sensor networking are discussed in this chapter.

#### 2.1 Motivation

In wireless sensor networks, the reliability of message delivery is low because the wireless medium used in the wireless sensor network has a high packet loss rate; and the possibility that node will not work occurs frequently because each node uses a small amount of battery power. In addition, if single path routing protocol is used, the load is weighted in a particular path, causing congestion. As a result, this degrades the overall network performance, as well as the increasing delays.

In contrast, if the routing protocol provides multiple paths, it not only guarantees fault tolerance, but also improves reliability by broadcasting messages using multiple paths. In addition, by distributing the load that is being focused on a particular path, it reduces incidence of congestion and increases the network life time and reliability. Therefore, multi-path routing is the one of several techniques being studied to solve the issue in the wireless sensor network.

However, the multi-path routing protocols that have been proposed so far either have too much message overhead for finding the multi-path or use a less efficient multi path to reduce this overhead. In addition, as the number of nodes in the network grows, the routing table size is enlarged; leaving it unsuitable for sensor nodes with a small amount of memory.

The efficient multi-path routing protocol proposed in this thesis is based on AODV [5] and DD [4] mechanisms; and each node decides whether to broadcast a RREQ(Route

REQuest) by comparing the hop count of sender and receiver during the initial topology formation. Each node that receives RREQ(Route REQuest) sends RREP with an uplink neighbor list to the sink using a unicast method. By doing this, the sink node can select multiple routes based on the routing table. In addition, each intermediate sensor node does not need to maintain its own routing table since the sink node sends all source routes to their destination.

## 2.2 Problem statement

Each node in the WSN is very small (about the size of a coin), and larger numbers of nodes create a more dense network than the existing ad-hoc network. In addition, WMSN(Wireless Multimedia Sensor Network)[6], using WSN, has gained attention due to the performance improvement of the node attached to the camera's image sensor. However, the existing routing protocols in ad-hoc environments generally find a single optimized path for data transfer. Subsequently, the energy consumption of sensor nodes located on the path is concentrated. In other words, the network capability might be degraded because some nodes in the sensor network consume all the energy. In addition, if a large amount of multimedia data coming into multiple paths is concentrated on a small number of sensor nodes, a situation that leads to discarding packets due to overflow will occur because the sensor nodes have a reception queue with small memory capacity.

Although some multipath routing protocols[7][8][9] have been proposed for improving reliability and throughput, such multipaths share a path with some nodes, and are not node-disjoint[10].

Therefore, for reliable multi-media data transfer in wireless sensor networks, a node-disjoint multipath routing protocol for transferring large amounts of data in the wireless sensor network is needed; not the routing protocol used in existing ad-hoc environment.

### 2.3 Approach to solve problem

As an approach to solve the above problems, this thesis proposes a routing protocol that supports reliable multimedia data transmission as well as load-balancing.

First, the data transmission in DD selects the path with the least delay, and transfers data to the destination node after a single path is reinforced. However, as already mentioned, sensor nodes located in the path consume large amounts of energy based on the single path system. As an alternative to this, the proposed protocol uses multiple paths in order to distribute the load concentrated on some nodes located on a single path. In addition, we propose the node-disjoint path selection algorithm.

Second, the hop number from each sensor node to sink node will be considered while setting up the gradient between a sink node and each sensor node. This can minimize the delay because the shortest path can be used.

Third, in the route maintenance phase, the routing overhead will be reduced by broadcasting RREQ to a limited range and not the overall network.

Finally, in AODV, the routing table size is large because each sensor node maintains the cache about the next hop information related to the destination node. This causes large memory consumption. However, in the proposed protocol, each sensor node maintains only an uplink with the neighbor node's information (which are node address and priority). In this way the memory consumption of sensor node can be minimized.

### 2.4 Objective

In this thesis, the routing protocol considered to achieve the following goals is proposed.

First, to compare the well-known DD with the proposed routing protocol to verify its performance.

Second, to minimize the control message overhead needed for the path, and implement memory-efficient routing protocol by minimizing the memory consumption of sensor nodes.

Third, to distribute the load on the sensor nodes located in the path by finding multiple node-disjoint paths.

Fourth, to obtain a multiple node-disjoint path that can be updated easily in case of path error.

Finally, to find the optimal simulation parameters for applying the proposed routing protocol to the wireless sensor network.



## Chapter 3

### Related Work

Routing protocols will now be discussed for sensor networks and wireless ad hoc networks devoid of predetermined backbone infrastructure[11].

#### 3.1 The difference between Wireless ad-hoc routing protocols and sensor network routing protocols

Although some of the wireless ad hoc networking techniques are applicable to wireless sensor networks, a sensor network differs from an ad hoc network in many aspects [12][13].

As we see in table 3.1, the existing wireless ad-hoc network routing protocol is not suitable for application to wireless sensor networks because of the given reasons.

	Sensor network routing	Wireless ad-hoc routing
Node number	Large	Small
Deployment	Dense	Loose
Topology change	Dynamic	Static
Communication method	broadcast	point to point
Capacity	limited power, CPU, and memory	better than sensor network

Table 3.1: Comparison of routing protocol

Therefore, the routing protocol for a wireless sensor network must be able to provide reliable communication with minimal use of energy under dynamic topology. If the existing wireless Ad-hoc routing protocol such as DSDV[14], DSR[15], etc. were applied to the wireless sensor network, the following problems would occur due to the characteristics of the sensor network mentioned above.

1) On demand routing scheme: In the case that many sensor nodes were distributed with high density, each node sends a RREQ(Route REQuest) message to find the best path, thereby dramatically increasing the energy consumption and communication delay.

2) Table-driven routing scheme: Sensor nodes have very limited data storage space, so it is difficult to maintain routing tables for all nodes.

### **3.2 Multipath routing protocol for sensor network**

There are several criteria in classification of routing protocols in a wireless sensor network, but the thing that can be differentiated from other wireless networks is the criteria based on how routing is conducted. According to this criterion, the routing in the sensor network can be classified into address-centric routing or data-centric routing.

Address-centric routing protocol [16] uses multi-path routing to increase the duration of the network by minimizing the energy consumption. This routing scheme proposes to select a multipath based on the probability of remaining amounts of energy in the nodes, and insists that it would improve the network's life time to use the node that has more remaining energy even if that node is not on the optimal path. However, this routing protocol uses on-demand route discovery, so it has to use control message overhead because the destination node broadcasts route request messages over the entire network whenever a path is needed.

In data-centric routing protocol[17], the sink broadcasts interest over the entire network. The sensor node that received the interest sends the information that it collected back to the sink through the reverse path configured while the sink broadcasted interest. There are several reverse paths that are set up to the sink because they were created by the sink's broadcast. The sink reduces the multipath into one of the multiple paths and sends a path reinforcement message to one of the neighbor nodes that have sent interest. The intermediate sensor node that received the path reinforcement message also sends the reinforcement

message to one of its neighbor nodes that have sent interest. When the reinforcement message arrives at the node that has the exploratory data, it sends the data to the sink using the single-path configured by the reinforcement message.

Another data-centric routing protocol [18], modified the Directed Diffusion protocol to ensure network fault tolerance. In this protocol, in order to configure the multipath, the sink node sends the path reinforcement message to several neighbor nodes instead of one of the neighbor nodes as in Directed Diffusion. Result [18] show that if a longer path in terms of hop number is found, it has poorer energy efficiency; even if finding the path consisting of all other nodes is better in terms of fault tolerance. Therefore, even if the same node is shared, the braided path having smaller hop numbers is more efficient.

Split Multipath Routing(SMR)[19] is a protocol in which paths share nodes if a disjoint path can not be found, and uses a per-packet allocation scheme. The per-packet allocation scheme enables packets to be sent over various paths, meaning they are not limited to only being split at the source[20].

Various routing protocols[21][22][23] that use multiple path have been proposed for WSNs with network reliability as their design priority. In addition, the data transmission relies mostly on the optimal path, and the alternative path is used in case the nodes on the primary route fail[1].

The proposed protocol is similar to [17][15][5].

- 1) The sink broadcasts interest over entire network and set up the gradient for forwarding exploratory data as in [17].

- 2) The sink computes all paths from itself to the source node, and intermediate nodes will forward the message to its neighbor node as in [15].

- 3) The sink use the sequence number for route maintenance, and request ID for route discovery. In addition nodes that are not on the path do not maintain the routing information, and exchange the routing table with the neighbor node [5]. This prevents loops from

occurring and allows path information to be kept up to date.

However, there are three significant differences.

First, in [5], all nodes maintain the path cache containing the source path, and update the path caches whenever the new paths are changed. This can cause much routing overhead. On the other hand, in the proposed protocol, intermediate nodes only maintain the information of uplink neighbor nodes to reduce their routing overhead. This can improve routing overhead by reducing the routing table size of the intermediate nodes.

Second, in [17], while the single-path was used to forward the exploratory data, the proposed protocol uses the multi-path selection algorithm to find as many multiple paths as possible to forward the exploratory data.

Third, the proposed protocol generates node-disjoint multipaths by exchanging the control message during the Building Network Topology phase without additional control messages.

Chapter 4  
Protocol Design

**4.1 Building network topology considering the hop count to the sink node**

In the proposed protocol, the first phase is to build the initial network topology. This process is started by broadcasting the BNTQ(Build Network Topology reQuest), setting the hop count to 0, the sequence number to 1, and the BNTQ sender address to itself. The sink node then broadcasts BNTQ(Build Network Topology reQuest) over the entire network.

The hop count increases by 1 as the number of nodes that are forwarded BNTQ(Build Network Topology reQuest) increases. The sequence number also increases by 1 as the sink node broadcasts BNTQ(Build Network Topology reQuest). The sensor node that receives the BNTQ(Build Network Topology reQuest) from the sink, or another sensor, processes the packet as in Figure 4.1.

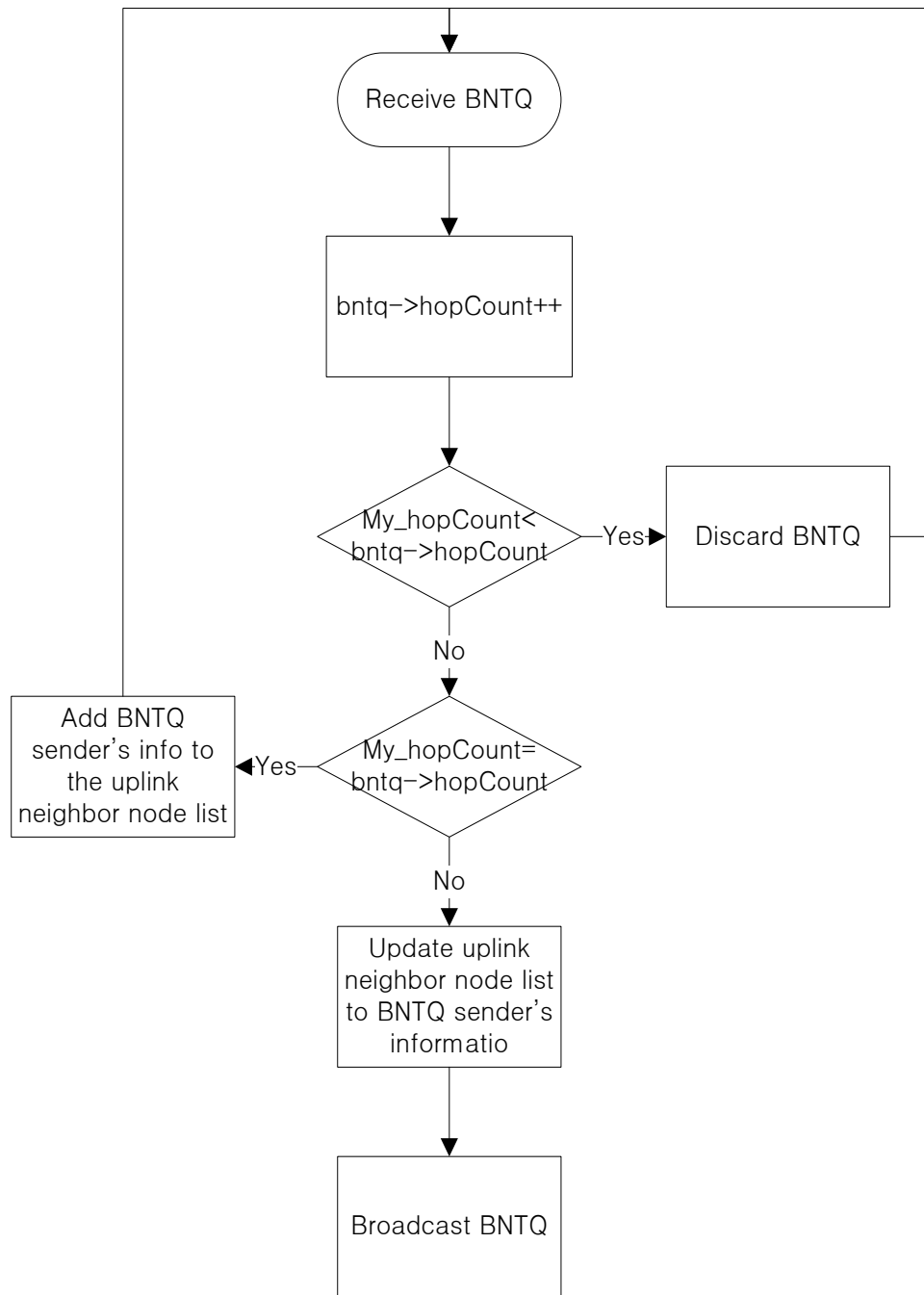


Figure 4.1: BNTQ process

Intermediate nodes receive the BNTQ packet first. The node checks the hop count from the sink node. The hop count is the number of nodes that the packet comes through. Table 4.1 shows the hop count of each node. As shown in Figure 4.2, when the 14 nodes have formed multiple paths, the sink node broadcasts BNTQ. Nodes 1,2,3,4, and 5 that receive the BNTQ packet increase their hop count and process the packet for a certain time (based on Figure 4.1). In this way, if the increased hop count is less than the existing hop count, the node updates the uplink neighbor node list to the BNTQ sender's information. If hop counts are the same, the node adds the sender's information to the existing uplink neighbor node list. If the increased hop count is larger than the existing hop count, the node discards the packet.

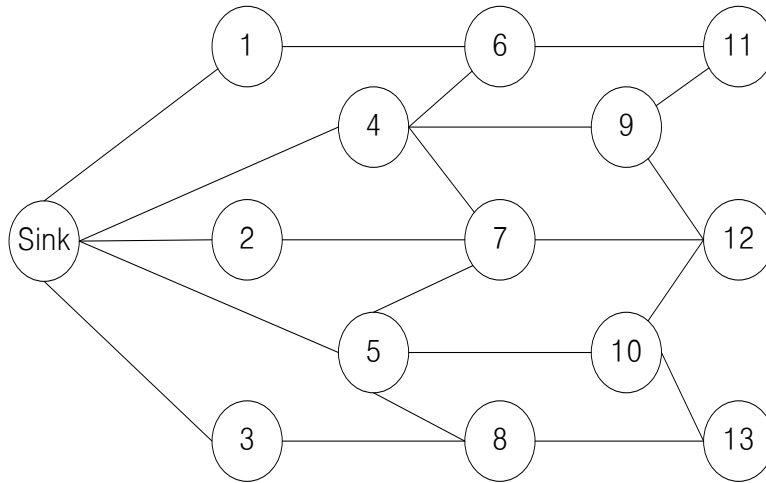


Figure 4.2: BNTQ broadcasting using hop count

Hop count	node
0	sink
1	1, 2, 3, 4, 5
2	6, 7, 8, 9, 10
3	11, 12, 13

Table 4.1: The Hop count of each node

For example, the hop count of BNTQ that node 6 received from node 1 is 2. In addition, the hop count of BNTQ that node 6 received from node 4 is also 2. Therefore, the uplink neighbor list of node 6 contains node 1 and node 4. However, the hop count of BNTQ that

node 6 received from node 7 is 3. Therefore, the BNTQ packet received from node 7 is discarded because hop count 3 is larger than 2, which is the existing hop count. By setting up the network topology based on the least hop count, the proposed protocol can not only create an optimal path that has the shortest length, but also prevent packet loops from occurring. In addition, by adding the information of nodes that has same hop count, the proposed protocol guarantees multiple paths.

After a certain time, the intermediate node sends BNTP(Build Network Topology rePly) with its hop count and uplink neighbor node lists to the sink.

Since the sink node broadcasts BNTQ(Build Network Topology reQuest), the sink node has been waiting to receive BNTP(Build Network Topology rePly) from each sensor node for a certain time. The sink node that receives BNTP(Build Network Topology rePly) compares the sequence number that the sink node finally broadcasted with the sequence number in received BNTP(Build Network Topology rePly). Only in the circumstance that two sequence numbers are the same, the sink node modifies the information of the BNTP(Build Network Topology rePly) sender on its' routing table. In the case that the sink node receives all BNTP(Build Network Topology rePly) from each sensor node, the sink node can generate its routing table. Table 4.2 shows the routing table of the sink node in the case of Figure 4.2.

Node	Available	Hopcount	Uplink neighbor nodes
1	T	1	sink
2	F	1	sink
3	T	1	sink
4	T	1	sink
5	F	1	1, 4
6	T	2	2, 4, 5
7	T	2	3, 5
8	T	2	3, 5
9	F	2	4
10	T	2	5
11	F	3	6, 9
12	T	3	7, 9, 10
13	T	3	8, 10

Table 4.2: Sink's routing table based on Figure 4.2



Based on the performance of the MAC protocol used in the sensor network, the loss of BNTQ(Build Network Topology reQuest) or BNTP(Build Network Topology rePly) can occur during the Building Network topology phase. Because the BNTQ(Build Network Topology reQuest) that the sink node broadcasts is broadcasted over the entire network, the loss can be reduced by broadcasting after a slight delay. In addition, a little loss doesn't affect the overall operation of the protocol. However, if the BNTP(Build Network Topology rePly) is lost, the sink node can't find the optimal route to the sensor node and then this can render the communication impossible. Therefore, it is necessary for the sink node to find sensor nodes that do not send a BNTP(Build Network Topology rePly) in order to set up the routing table for the sensor nodes.

The method is as follows:

After a certain time of receiving BNTP, the sink node searches for a node that is on the uplink neighbor list, but is not on the its routing table. By doing this, the sink node can find which node did not receive a BNTP(Build Network Topology rePly) from.

Node	Available	Hopcount	Uplink neighbor nodes
3	T	1	sink
8	T	2	3, 5
10	T	2	5
13	T	3	8, 10

Table 4.3: Sink's routing table based on Figure 4.3

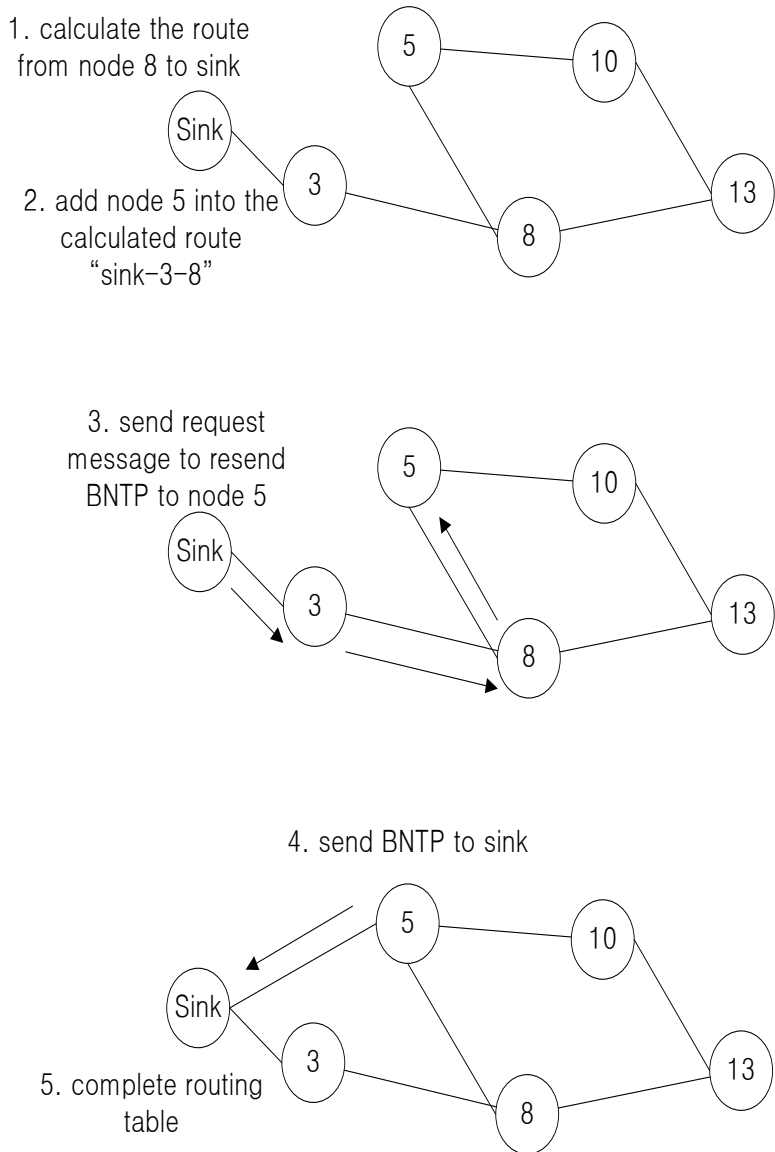


Figure 4.3: BNTTP error process

In order to send a request message to the problem node that didn't send BNTP(Build Network Topology rePly), the sink node finds a node that has a problem node as an uplink neighbor node and then computes the path from the node to the sink. Then the sink node adds the problem node to the path. Using this path, the sink node unicasts a control message requesting BNTP(Build Network Topology rePly); and the sensor node that receives this control message resends BNTP(Build Network Topology rePly). By doing this, the sink node can configure the routing table completely.

In this phase, the sink node uses a hop count metric based on its routing table to establish an uplink gradient from the sink node to each sensor node. Subsequently, the links from the sink node to each sensor node are generated in a downlink direction by reversing the gradient.

For example, Table 4.3 shows the sink's incomplete routing table based on the Figure 4.3.

In this case, node 5 is on the uplink neighbor list of node 8 and 10; but node 5 does not exist on the sink's routing table. That means the sink node did not receive BNTP(Build Network Topology rePly) from node 5. In order to ask node 5 to send BNTP(Build Network Topology rePly), sink computes the route from itself to node 8 (which has node 5 as an uplink neighbor node). The route computed by the sink is "sink-3-8." Then the sink adds node 5 to the computed path to make the entire route "sink-3-8-5". Using this route, the sink forwards the control message to node 5 asking it to send BNTP(Build Network Topology rePly).

## **4.2 Route discovery using the priority, shortest hop number and source routing**

### **4.2.1 Data propagation from the sink node to the sensor node**

When the sink node has a message to send the destination node, it computes the entire route to the destination and then sends the data packet containing the entirely computed route. The intermediate node that received the data packet does not need to maintain its

routing table because the information on where to send the data packet is contained within the packet itself. As a result, this proposed protocol can reduce routing message overhead and prevent looping since the intermediate node does not compute the route on where to send the packet itself using local information.

As described in this protocol, the sink node computes the entire path to the destination by traversing the uplink gradient. A loop will not occur because the direction of the uplink gradient is toward to a sink node and each node adds uplink neighbor nodes that have only the smallest hop number during Building Network Topology phase.

#### 1) Multiple path selection at the sink node

Suppose that the sink node has a data packet to send to node 12 in Figure 4.4. If the sink node computes the entire path as "Sink-4-7-10-12", the sink node will not find three different paths ("Sink-4-9-12", "Sink-2-7-12" and "Sink-5-10-12") that consist of different nodes, because node 10 selects node 7 since it has the same hop number. The proposed protocol uses the priority queue based on the hop number and uplink neighbor nodes' numbers in order to find multiple node-disjoint routes. In this priority queue, if the hop number is larger, the priority to select that node as an uplink node is higher; and if the hop number is same, the priority of the node that has smaller neighbor nodes is higher. In this way, this protocol can find as many multiple routes that consists of different nodes as possible.

In Figure 4.4, node 12 can select node 9, node 7, or node 10 since they all share same hop number, 2. Then node 12 inserts nodes into the priority queue based on the node number, from smallest to largest, like the 7 - 9 - 10 order. But the extension order of the node to be selected as an uplink node in the priority queue depends on the number of the uplink neighbor node that each node has. That is, node 12 extends to node 9 first because the uplink neighbor node of node 9 is less than that of node 7, and node 9 was inserted into the queue earlier than node 10. Like the preceding, in search step 3, even if node 7 and node 10 are in the same queue and node 7 was inserted earlier than node 10, it guarantees finding the multiple routes by extending node 10 first since it has less uplink neighbor node than

node 7. By reversing the route from the destination node to the sink node, the sink node can find node-disjoint multiple routes.

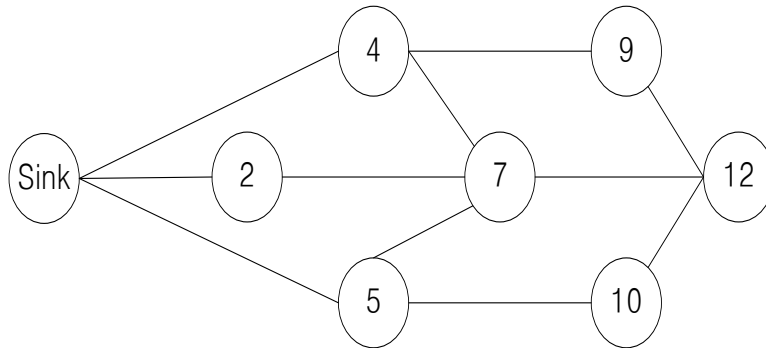


Figure 4.4: Example of multipath selection

The order of the queue's extension is as seen in Table 4.4:

Step	Queue(hop number, uplink neighbor number)	extension	Path
1	7(2,3), 9(2,1), 10(2,1)	9	12-9
2	7(2,3), 10(2,1), 4(1,1)	10	12-9 12-10
3	7(2,3), 4(1,1), 5(1,1)	7	12-9 12-10 12-7
4	4(1,1), 5(1,1), 2(1,1)	4	12-9-4 12-10 12-7
5	5(1,1), 2(1,1)	5	12-9-4 12-10-5 12-7
6	2(1,1)	2	12-9-4 12-10-5 12-7-2

Table 4.4: The order of queue's extension

#### 4.2.2 Data propagation from the sensor node to the sink node

The priority of a sensor node's uplink neighbor node is set to 10. When selecting a uplink node, the sensor node generates a random number between 1 and the total sum of all uplink nodes' priorities; and then looks up the uplink neighbor nodes. Whenever looking at

uplink neighbor nodes, the priority of each uplink neighbor node is progressively subtracted from the generated random number and the difference of the last node's priority. When the generated random number is 0 or less, that uplink node is selected as an uplink path node. If the transmission is successful using selected uplink node, 5 is added to the priority of the node. On the other hand, if the transmission fails, 5 is subtracted from the priority of the node. The maximum priority is 20 and the minimum priority is 0. When uplink neighbor node is 0, the sensor node deletes the uplink neighbor node that has the priority 0 from its uplink neighbor nodes, and then broadcasts RREQ if the sensor node has no more uplink neighbor node. Figure 4.5 shows how to find the uplink neighbor node reaching toward the sink node.

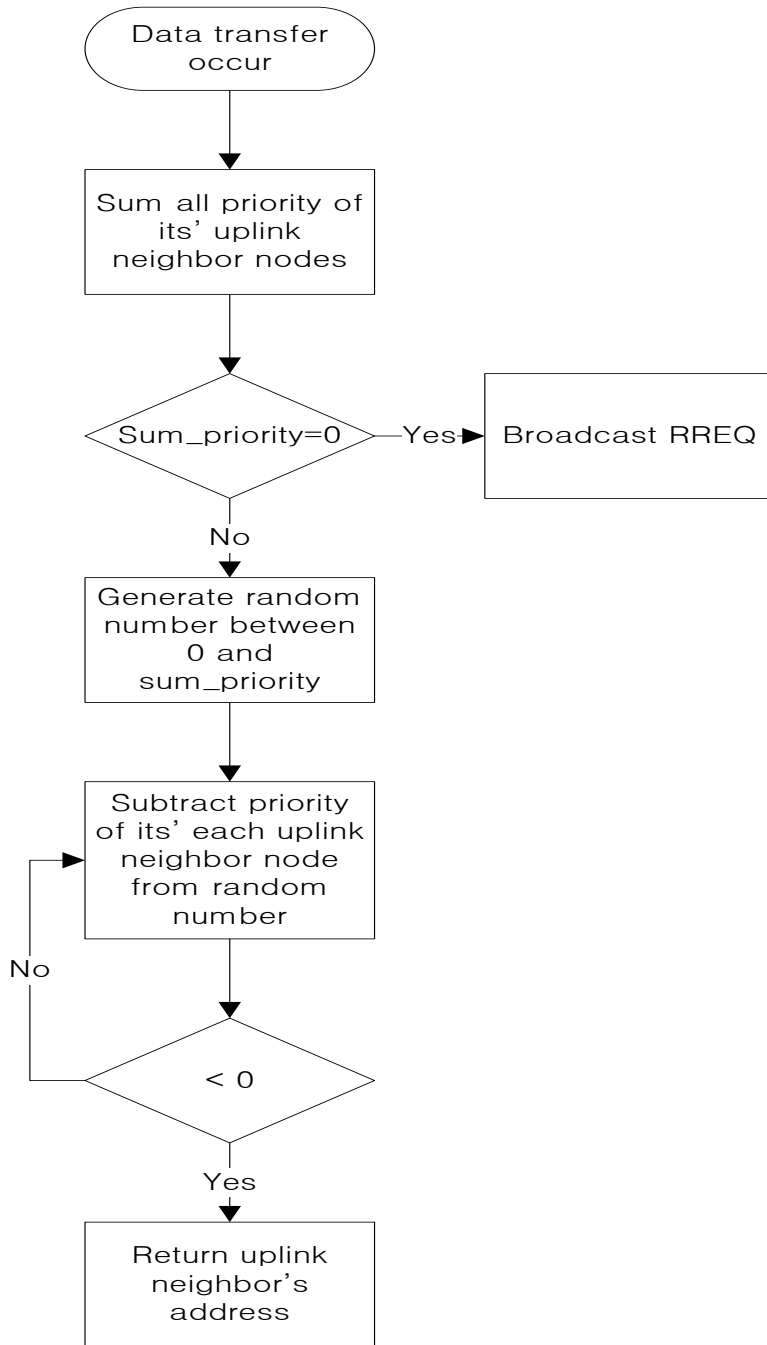


Figure 4.5: Uplink path node selection process

### 4.3 Route Maintenance

Because this protocol assumes that the node is not mobile, the meaning of a path disconnection can be interpreted from two perspectives. First, in the case that the sensor node does not work anymore, it would be better to not use that node. Second, in the case that the network is congested, it would be better that the sink node makes the node temporarily inactive, and reuses the problem node after the network congestion has dissolved. This section considers how the proposed protocol recovers in case of route disconnection.

#### 4.3.1 From sink node to destination

After the sink node completes a route computation, it is possible that the packet with the entire path source cannot reach the next node. In this case, proposed protocol considers two situations.

The first one is that the sink node finds the path disconnection. In this case, the sink node makes the problem node inactive, and keeps propagating a packet by using the other routes.

The second one is that the intermediate node finds the path disconnection. In this case, the intermediate node sends REM(Route Error Message) to the sink. The REM contains an unreachable node address and original destination address. If there is no response from the sink node for a certain time, the intermediate node drops the packet.

The sink node that received REM(Route Error Message) makes the unreachable node inactive on its routing table, and then recomputes the route from the original destination to the sender of REM(Route Error Message). The sink node sends RRM(Route Recovery Message) with a newly computed path to the sender of REM(Route Error Message). The intermediate node that receives the RRM(Route Recovery Message) keeps forwarding the packet without any loss using the new source route in the RRM(Route Recovery Message). If the sink node does not find any route from the original destination to the sender of REM and



does not send RRM(Route Recovery Message) to the sender of REM(Route Error Message), the sender that didn't receive RRM(Route Recovery Message) drops the packet.

In order to confirm whether the error of the node is caused by energy problems or by temporary network congestion, the sink node sends a checking message to the unreachable node after a certain time. If a node receives the checking message, it sends "checking reply message" to the sink node with its status as normal. The sink node that received the "checking reply message" updates its routing table to make the status of that node active. If the sink node does not receive "checking reply message", it tries to check the status of the node by increasing the transmission cycle. By exchanging the checking message between the sink node and inactive node, the proposed protocol prevents the possibility that a node which is inactive due to the temporary network congestion will not be considered for use later.

Figure 4.6 shows an example of the downlink route maintenance.

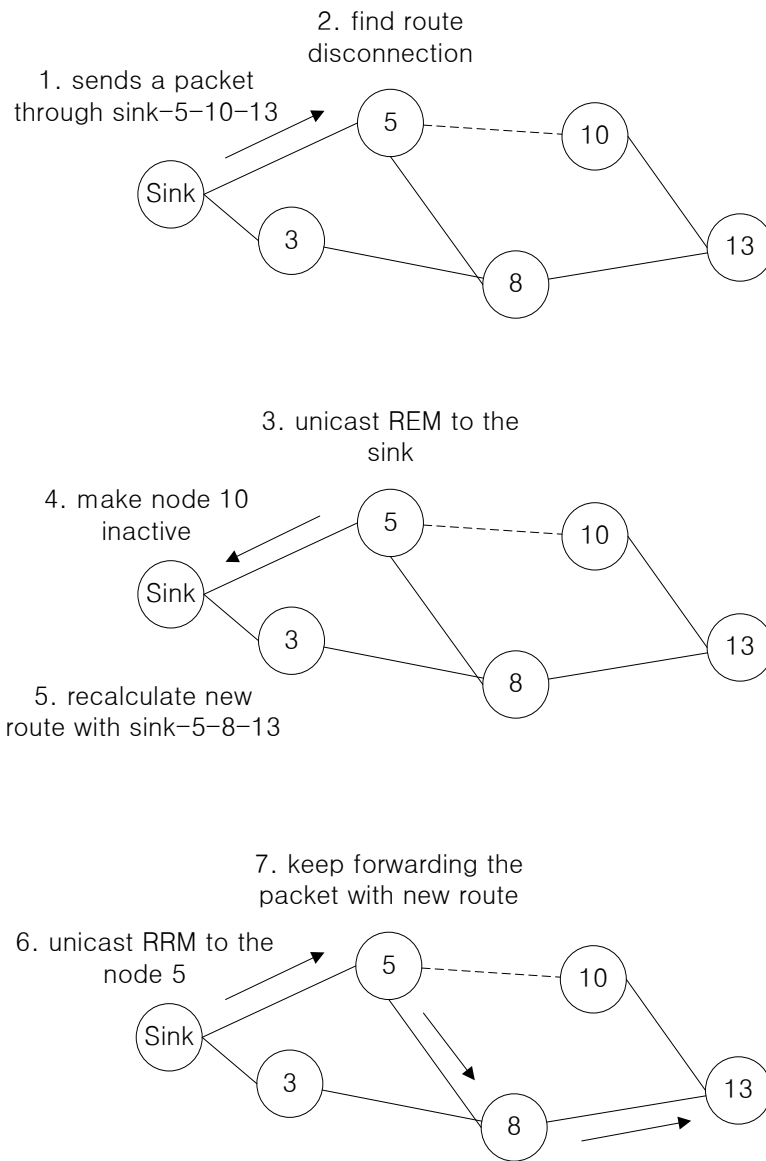


Figure 4.6: Downlink route maintenance

The sink node forwards a packet to the node 13 through the route "sink-5-10-13". Node 5, which received the packet, finds a link disconnection between node 5 and node 10. After storing the packet to be sent in the queue, node 5 unicasts REM(Route Error Message) with the address of the unreachable node 10 and destination 13 to the sink node. The sink node that receives REM(Route Error Message) makes node 10 inactive on its routing table, and then recomputes the path from node 5 to the node 13. The sink node unicasts RRM(Route Recovery Message) to node 5 with the newly computed path. When node 5 receives RRM(Route Recovery Message), it continues forwarding the packet through the route "sink-5-8-13" without any packet loss.

#### **4.3.2 From sensor to the sink**

When the sensor node has a data packet to forward the sink node, the sensor node uses the gradient path generated for a building network topology phase. The gradient path method is widely used when several nodes send packets to one node. The proposed protocol also uses this gradient method to forward the data packet from sensor node to the sink node. Each sensor node sends the data to one of its uplink neighbor nodes that have smaller hop count than the sender's. If the uplink neighbor node selected first can not be reached, the route can be recovered by selecting other nodes among uplink neighbor nodes. At this time, the selection priority of the unreachable intermediate node is lowered because its unreachability can be temporarily caused by network congestion, and not node error. This prevents wrongly deleting useful nodes from their uplink neighbor list. If the selection priority of the unreachable intermediate node is less than zero, it is considered to be a node error (like energy exhaustion); and in this case the unreachable intermediate node is totally deleted from the uplink neighbor's list of nodes that should forward the data packet. Then the node that should forward data sends REM(Route Error Message) with the unreachable node's address to the sink node by selecting another uplink node. The sink node that received REM(Route

Error Message) makes the unreachable node inactive on its routing table. By doing this, the sink node can compute the route to the destination node correctly without any concern.

If all uplink neighbor nodes of a node are unable to receive some packets, that means there is no uplink neighbor node able to forward the packet. The node stores REM(Route Error Message) and the packet to be forwarded in the transmission queue. Then it broadcasts RREQ(Route REQuest) to neighbor nodes within one hop of itself to check whether neighbor nodes have some routes to reach to the sink. RREQ(Route REQuest) contains the generating node's address. The node that sent RREQ(Route REQuest) waits for the RREP(Route rePly) for a certain time. The neighbor nodes that received RREQ(Route REQuest) delete the sender of RREQ(Route REQuest) from their uplink neighbor list, and the neighbor node sends RREP(Route rePly) with its hop number to the sender of RREQ(Route REQuest) if the neighbor node has another uplink neighbor node to reach the sink node. By deleting the sender of RREQ(Route REQuest) from its uplink neighbor list, the proposed protocol can reduce the packet delivery delay and guarantee reliable packet forwarding.

When the node that broadcasted RREQ(Route REQuest) receives the RREP(Route rePly) from neighbor nodes, the node updates its uplink neighbor list by following the process of receiving a BNTQ(Build Network Topology reQuest) to update its uplink neighbor list. That is, the node adds a node that has the smallest hop number as an uplink neighbor node. When path recovery is completed successfully, the node keeps forwarding the data stored in the transmission queue without packet loss.

Figure 4.7 shows the example of the uplink route maintenance.

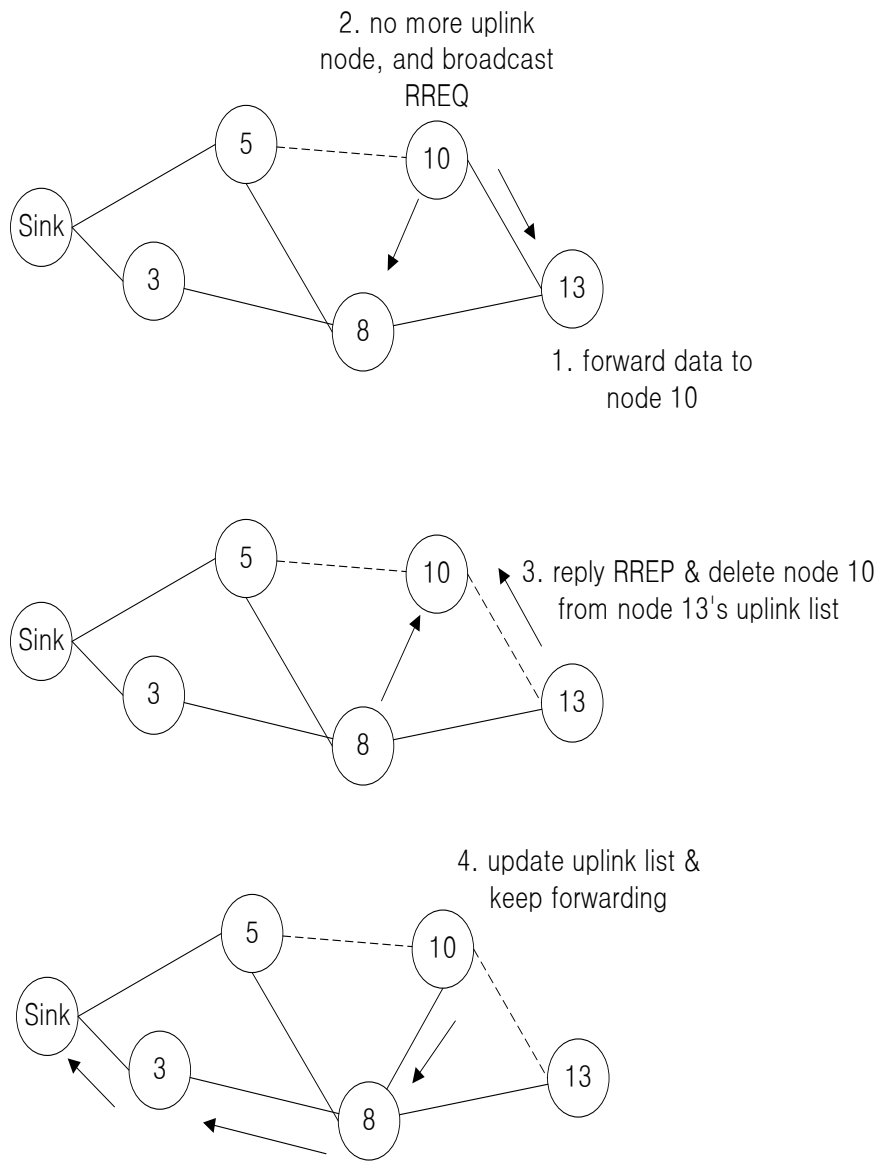


Figure 4.7: Uplink route maintenance

When the sensor node 13 has data to send the sink node, it sends the data to node 10 based on the uplink neighbor selection algorithm. Node 10 finds a link disconnection to node 5 because the priority of node 5 is 0. Therefore, node 10 has no uplink neighbor node to reach the sink node. In this case, node 10 buffers the REM(Route Error Message) and data in the queue and then broadcasts RREQ(Route REQuest) to the neighbor nodes (node 8 and node 13), within 1 hop of its own hop number. Nodes 8 and 13 receive RREQ(Route REQuest) and then delete node 10 from their uplink neighbor list and unicast to node 10 with RREP(Route rePly) with its hop number, because those nodes have a possible path to reach a sink node. When node 10 receives the RREP(Route rePly) of node 8 and node 13, it updates its uplink neighbor list to add node 8 as a new node on its uplink neighbor list; but not node 13 because the hop number of node 8 is 2 which is smaller than the hop number of node 13, which is 3. Then node 10 forwards the buffered REM(Route Error Message) and data to node 8. The other process is the same as the uplink forwarding process.

While the intermediate sensor node recovers and exchanges RREQ(Route REQuest) and RREP(Route rePly) with the sink node, the sink node can delete the problem node from its routing table by receiving an REM(Route Error Message) with data that the sensor node sent. However, the sink node can't modify which neighbor node was added to the sensor node's uplink neighbor list. Therefore, the sensor node should notify the sink which neighbor node was added by unicasting NLU(Neighbor List Update) message. NLU(Neighbor List Update) contains the node's hop number, address, and uplink neighbor lists. The sink node that received NLU(Neighbor List Update) updates the uplink neighbor list of the sender of NLU(Neighbor List Update) on its' routing table.

If the sensor node that broadcasted RREQ(Route REQuest) does not receive any RREP(Route rePly) from neighbor nodes, it broadcasts RREQ(Route REQuest) again after increasing the waiting time twice because the problem can be caused by network congestion. If the sensor node still does not receive any RREP up to the maximum times, the sensor node stops communicating because it is considered unreachable to any other node.

## Chapter 5

### Performance Evaluation

#### 5.1 The goal of performance evaluation

In this chapter, the validity of EMRP protocol proposed in this thesis is verified by evaluating its performance and finding the optimal value among selectable parameters. Since the proposed EMRP protocol is for solving the problem in the existing multi-path routing algorithm, with many control message overheads being used to find multiple paths, the goal of the proposed protocol is to evaluate how load-balancing by node-disjoint multiple routes affects the entire network. In order to do this, based on the discovered node-disjoint multiple routes, how a path number affects network performance will be evaluated and the optimal simulation parameters for the proposed routing protocol will be found. In addition, the network delay, the size of the routing table, and packet delivery ratio will all be evaluated.

#### 5.2 Simulation methodology

The proposed protocol operates under the following assumptions:

- Nodes are deployed randomly on the topology, and one sink node and one source node exist on the sensor network. The sink node is in the left bottom corner and the source node is in the top right corner.
- Almost all sensor nodes have two or more uplink neighbor nodes because the sensor nodes were deployed densely.
- Every node has no mobility.

The same simulation environment as the DCHT[20] is used. Table 5.1 shows the simulation environment.

Parameter Type	Value
Mac Protocol	802.11
Channel bandwidth	2Mbps
Transmission range	250m
Interference range	550m
Packet size	128 byte
Simulation time	1000s

Table 5.1: The simulation methodology

The sink node starts broadcasting the BNTQ(Building Network Topology reQuest) packet in the proposed protocol. Then, the sink node buffers the BNTP(Building Network Topology rePly) packet until a timer expires and configures its routing table containing all source nodes' information. The timeout value depends on the network size. On expiration of the timer, the sink node computes the route from sink node to destination node by reversing an uplink gradient. Then, the sink node forwards the packet with the total route, and the intermediate nodes on the route forward that packet to the next node on the total route without any computing. Upon receiving the REM(Route Error Message) and NLU(Node List Update) from intermediate nodes, the sink node can modify its routing table successfully. This guarantees highly reliable forwarding.

The source node that has data to forward to the sink node gets the next hop node address through an uplink neighbor node selection process. By doing this, the proposed protocol guarantees the load-balancing of the network. Then, the source node forwards the data to the next hop node, and the next hop node forwards the data to the next hop node, and so on until it reaches the sink node. By increasing and decreasing the priority of each uplink neighbor node based on whether the forwarding is successful or not, intermediate nodes prevent wrongly removing an active node.

### 5.3 Performance evaluation metric

The performance evaluation metric used in this thesis is as shown in Table 5.2.



Content	Measure
Throughput	total packets / total simulation time
End to end delay	Total latency / The number of data
The number of node-disjoint multipath	Count the number of node-disjoint multipath
Multipath ratio	The ratio of node-disjoint multipath per network size
Routing table size	The total size of the routing table each node maintains
Packet delivery ratio	received packet / sent packet
The effect of uplink neighbor node's number	the number of neighbor node : 4, 8, 10, 12, and 16
Transmission interval effect (Throughput, End-to-End delay, PDF, Packet loss rate)	Interval : 10ms, 20ms, 30ms, 40ms, and 50ms
The effect of path number(End-to-End delay, PDF)	path number : 1, 2, 3, and 4

Table 5.2: The performance evaluation metrics

## 5.4 Performance comparison

The proposed protocol, EMRP, is compared with DCHT and Directed Diffusion. The different protocols are compared over different network sizes of 46, 77, 116 and 163 nodes configured in a rectangular area of 2000m by 2000m.

### 5.4.1 Throughput

Figure 5.1 shows that EMRP guarantees an almost constant throughput as the network size increases, while the fluctuation range of other protocols are sharp.

In addition, EMRP gives better throughput than DCHT when the network size increases to 163. Basic Diffusion does not consider quality link at all. This basically prevents it from being able to receive any packet once the network size has reached or exceeded 77[20].

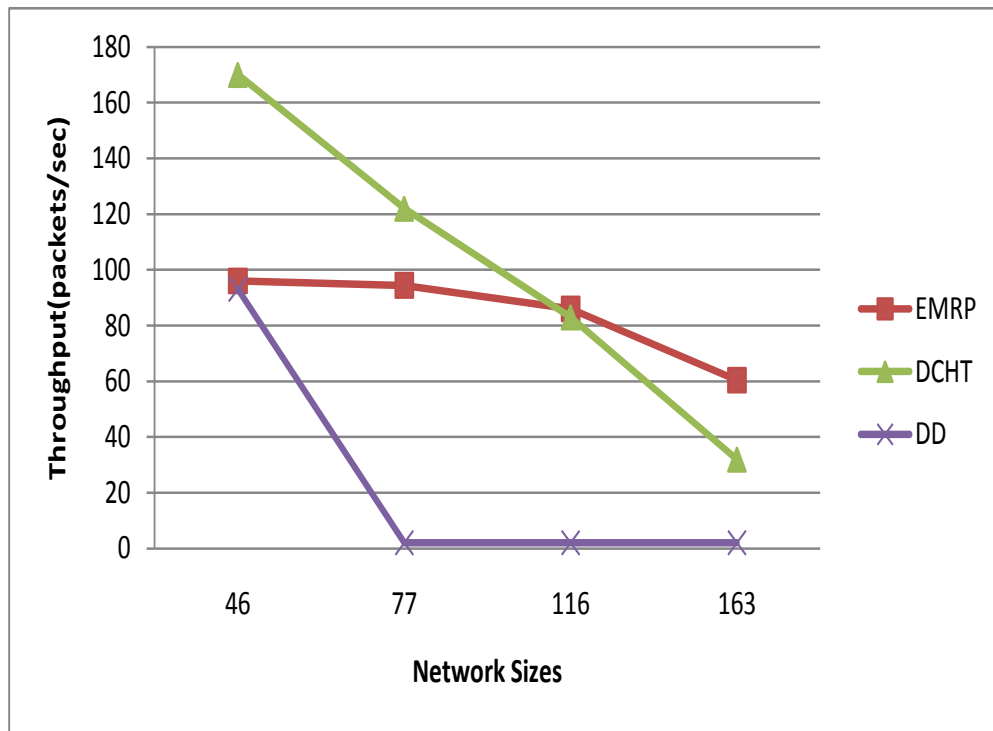


Figure 5.1: Throughput of EMRP, DCHT, and Directed Diffusion with different network sizes

### 5.4.2 End to end delay

In Figure 5.2, the average end-to-end delay includes all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times of data packets [24]. Even though Directed Diffusion shows lower delay than DCHT because it reinforces at least one neighbor after the sink starts receiving events [25], the delay of Directed Diffusion increases as the network size increases. However, that of EMRP stabilizes a little less than 50ms regardless of the network size because EMRP can find the shortest, optimal route and can reduce the occurrence of congestion significantly by distributing the load weight on each node.

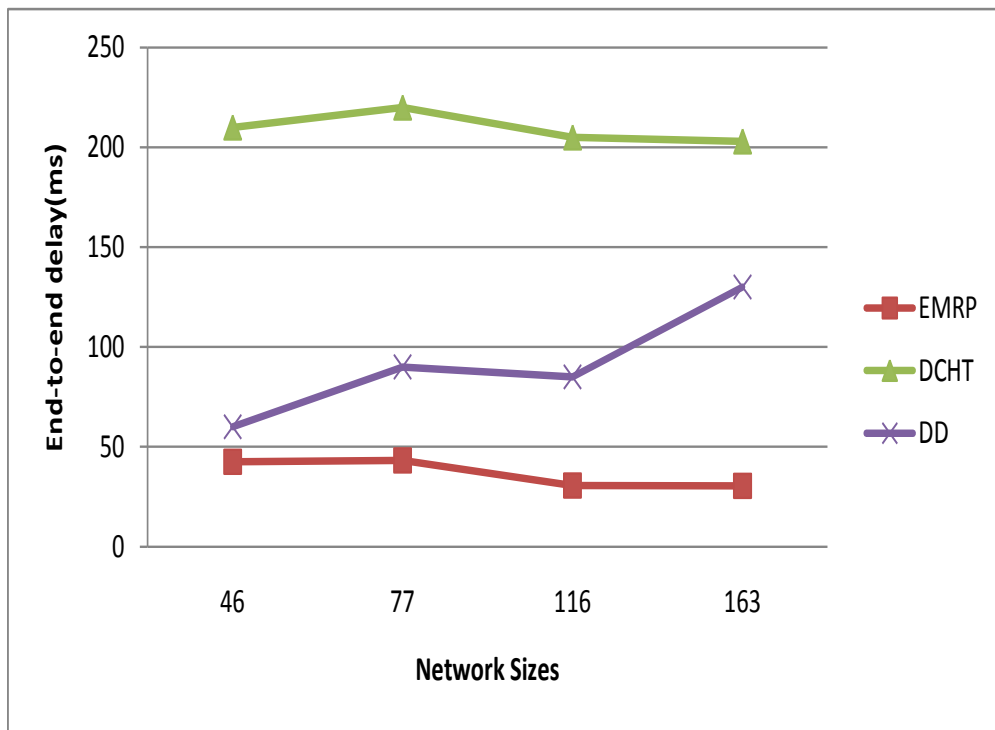


Figure 5.2: End-to-end delay of EMRP, DCHT, and Directed Diffusion with different network sizes

### 5.4.3 The number of node-disjoint multipath and Multipath ratio

The number of node-disjoint multipath represents the sum of the number of independent paths from the sink node to each sensor node. Figure 5.5 and Figure 5.4 show how to count the path number from sink node to each sensor node. Intuitively, in the case that node deployment is dense, the number of node-disjoint multipath would be founded more than that of scattered deployed because the number of selectable uplink neighbor node increase. In order to do this, EMRP protocol will be tested at different sizes to find the optimal network size. Figure 5.3 and Figure 5.6 show the number of node-disjoint multipath and the ratio of the multipath number to the network size. As the network size increases, the founded multipath number also increases. As a result, EMRP is most effective to find multiple node-disjoint paths when the network size is 116.

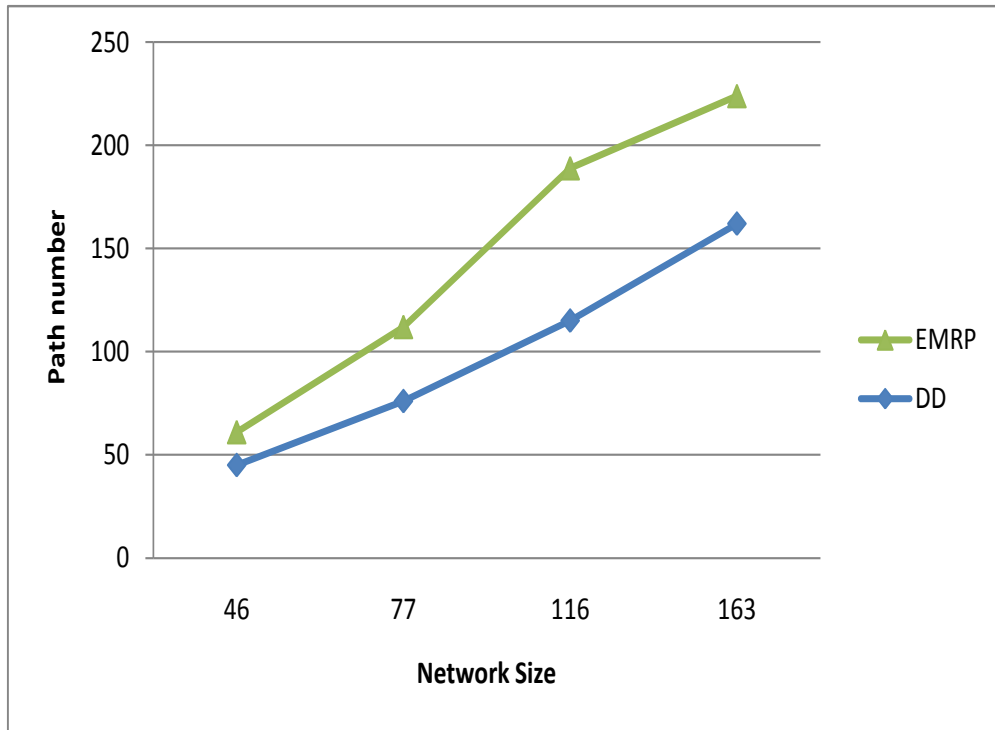


Figure 5.3: Multipath number from sink node to source node with different network sizes

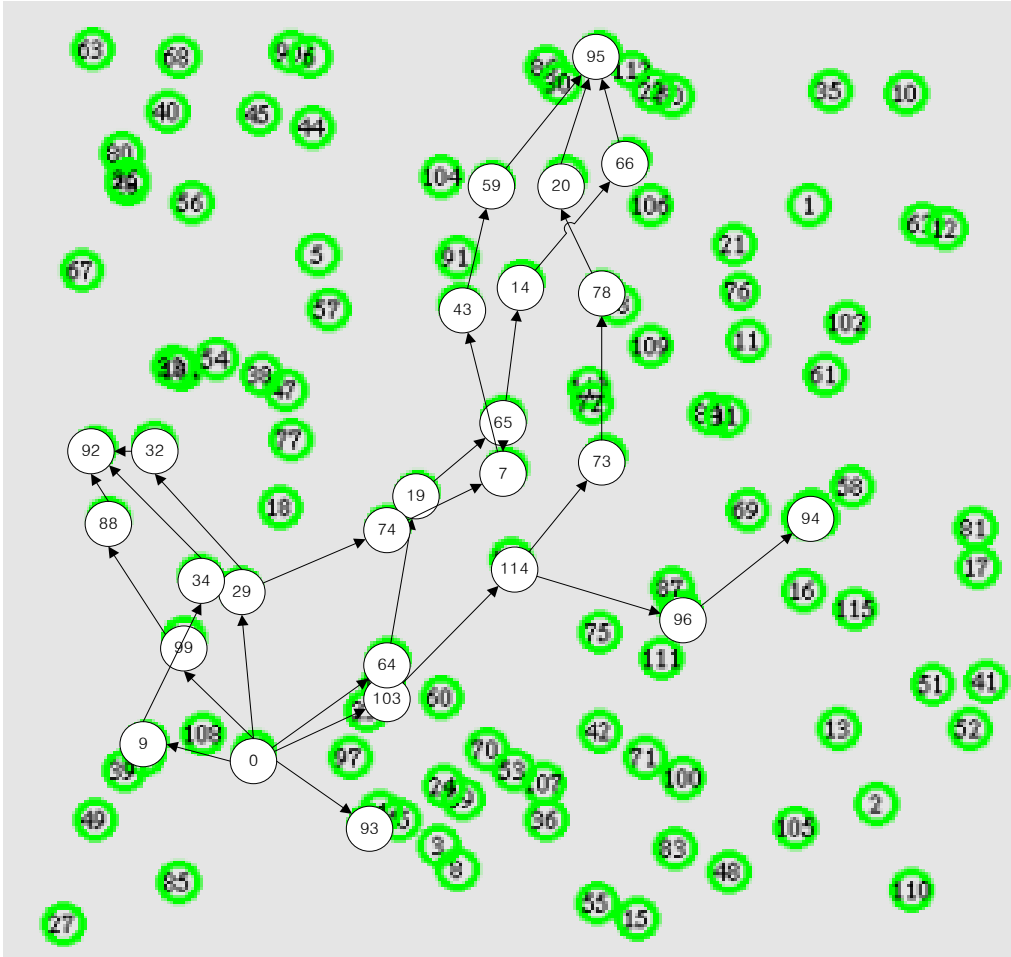


Figure 5.4: Count example 1 of multipath number from sink node to each source node

```

Node 92, hopCount 3
1: 29 -> 32
2: 99 -> 88
3: 9 -> 34
4: no.
Node 93, hopCount 1
no calc path cause its 1-hop
Node 94, hopCount 4
1: 103 -> 114-> 96
2: no.
Node 95, hopCount 6
1: 64 -> 19-> 65-> 14-> 66
2: 103 -> 114-> 73-> 78-> 20
3: 29 -> 74-> 7-> 43-> 59
4: no.
Node 96, hopCount 3
1: 103 -> 114
2: no.

```

Figure 5.5: Count example 2 of multipath number from sink node to each source node2

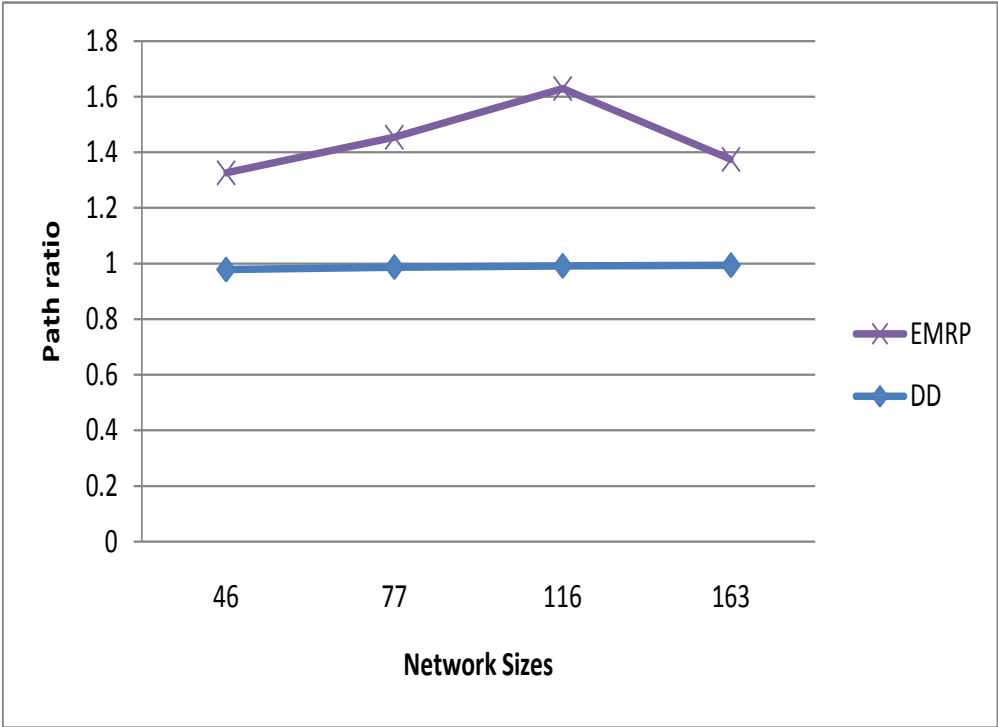


Figure 5.6: Ratio of Multipath number to network sizes

#### 5.4.4 Routing table size

In order to check how big a network can be supported by EMRP, the size of the routing table is measured. Figure 5.7 shows the result. The routing table size represents the value of the size of the node address multiplied by the number of the node. In addition avg-interm represents the average routing table size intermediate nodes maintain by dividing the total size to network size. The size of the routing table that the sink maintains is larger than that which intermediate nodes maintain (which is less than 15 bytes), because intermediate nodes only maintain their uplink neighbor nodes. By analyzing this result, it can be concluded that the memory consumption of an intermediate node is very low; and EMRP is very efficient in terms of memory consumption because all nodes in other protocols almost maintain the same routing table.

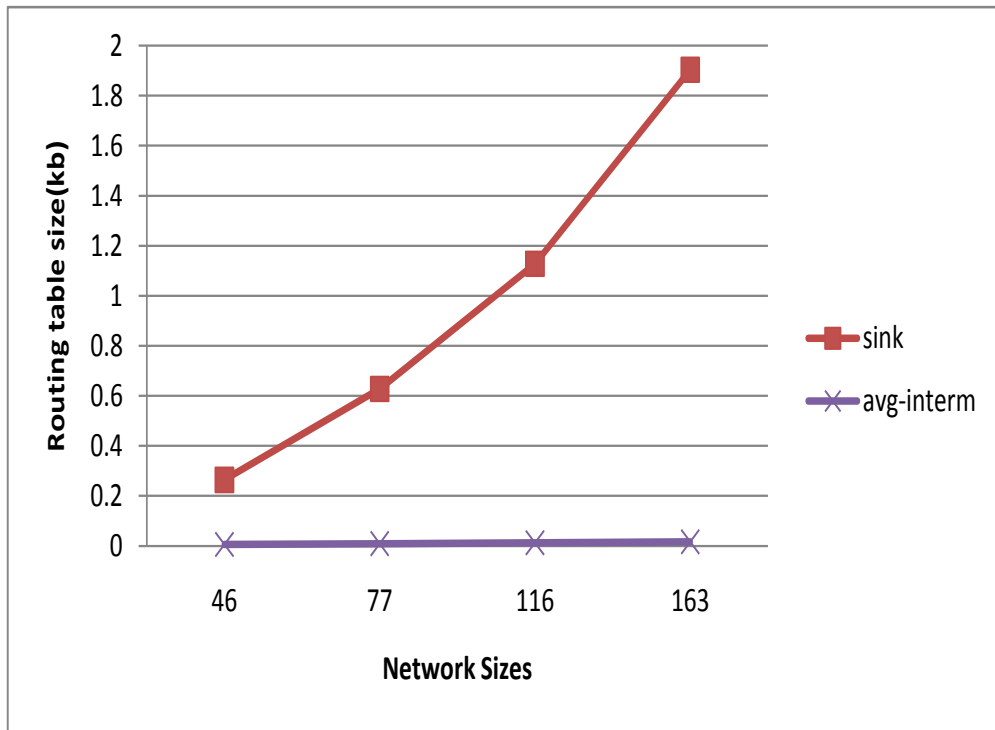


Figure 5.7: Routing table size of sink and each sensor nodes with different network sizes

### 5.4.5 The effect of the uplink neighbor number

I measured the protocol changing the number of uplink neighbor nodes each node can have to 4, 8, 10, 12, and 16 in order to check how the number of uplink neighbor nodes affects EMRP. Intuitively, the multipath number may increase because each node has ample opportunity to form the multipath as the network size increases. Figure 5.8 shows the result. The multipath number increases generally as the number of uplink neighbor node increases, but there is no significant change when the number of uplink neighbor node is between 12 and 16.

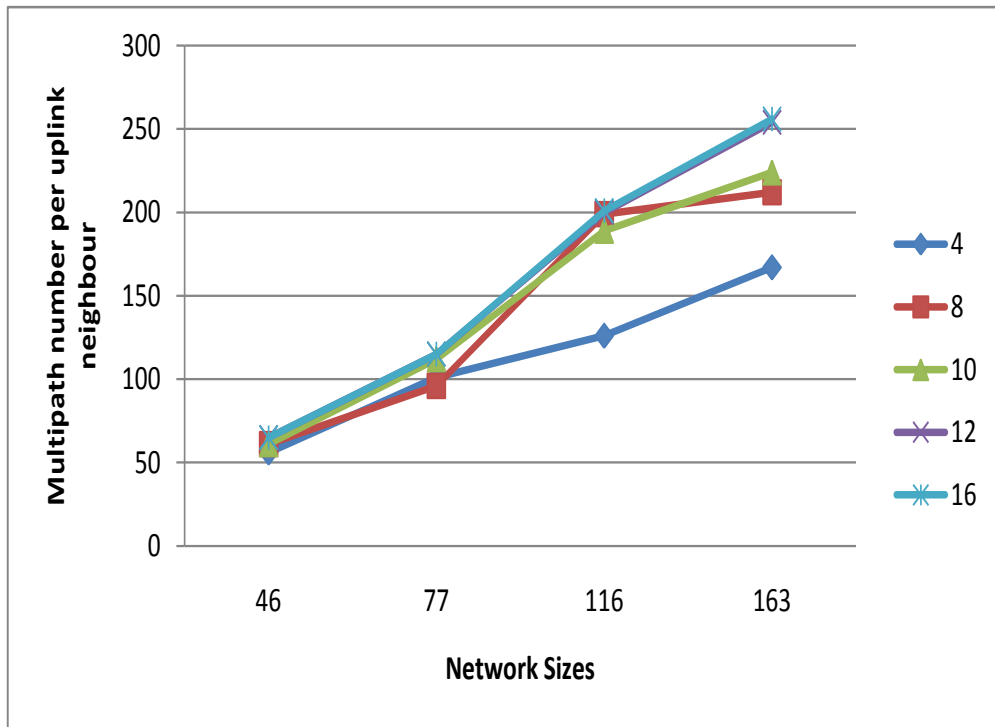


Figure 5.8: Multipath number with different uplink neighbor node number that sensor node can maintain



### 5.4.6 The effect of path number

In order to find three or four paths easily, we experiment in a network size of 116 and also experiment with EMRP by changing its transmission interval to 10ms, 20ms, and 30ms. Figure 5.9 and Figure 5.10 show the results. While DCHT has great improvement of packet delivery ratio and end-to-end delay as the discovered path increases, the performances in EMRP does not improve in proportion to the number of discovered paths, even if the performance improves slightly as the number of discovered paths increases.

The performance difference between a single path being used and 2 paths being used is large in EMRP. That is, the effect on performance when using 2 paths is greater than a single path. However, when the path number is increased to 3 or 4, there is no big difference of the performance improvement. By observing this result, this thesis finds that the performance improvement in EMRP is not proportional to the path number.

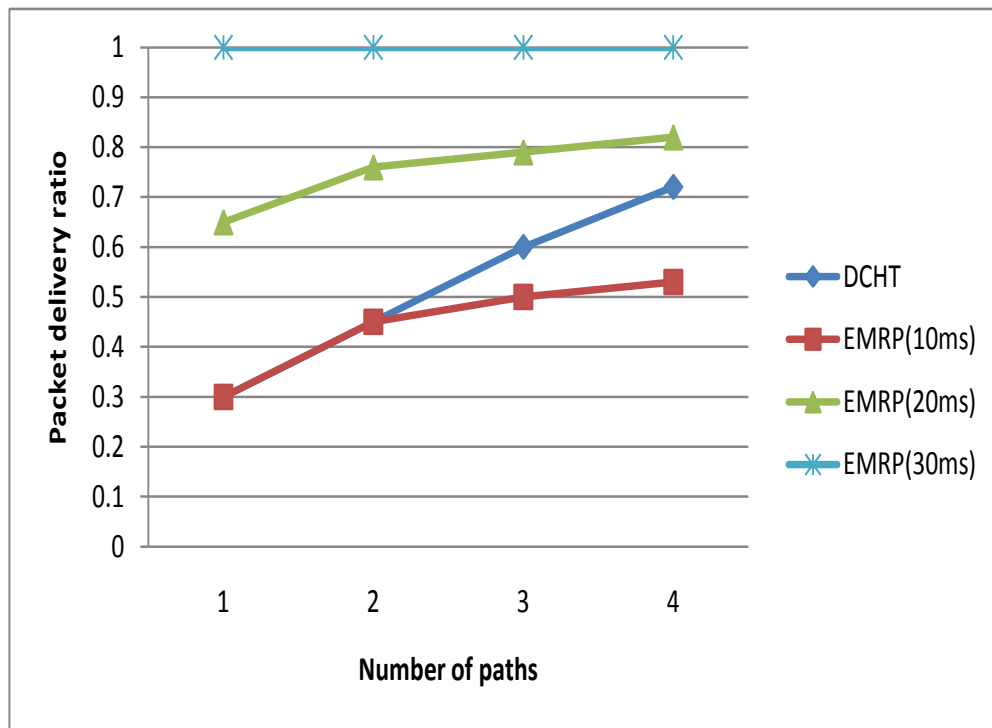


Figure 5.9: Packet delivery ratio of EMRP with different path number. The network size is 116 nodes

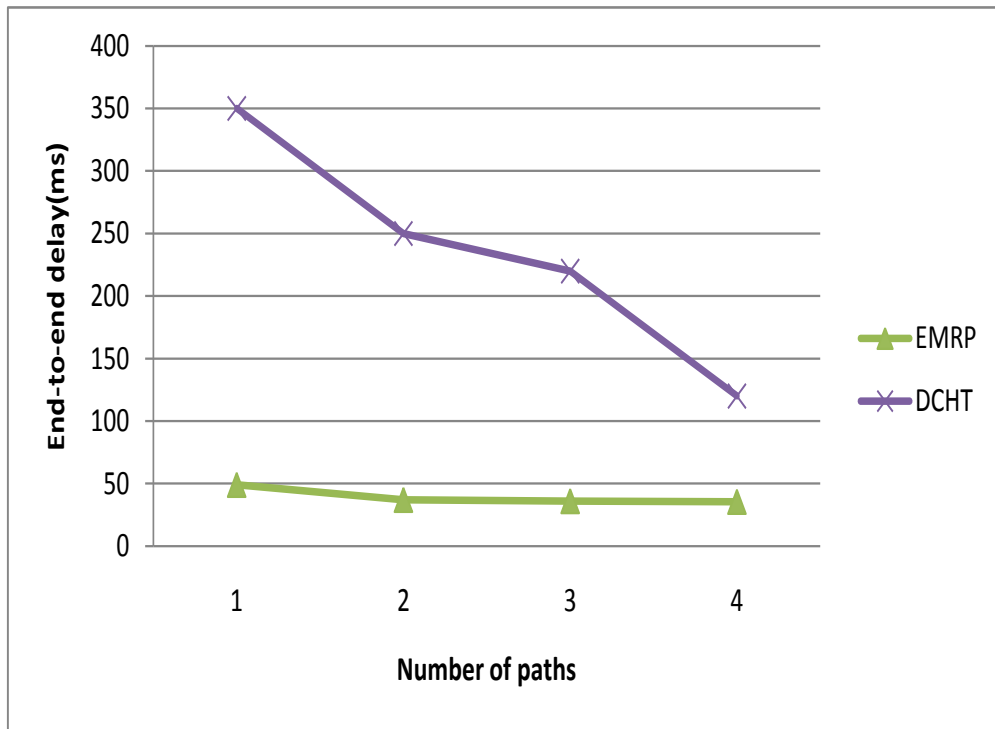


Figure 5.10: Packet loss rate of EMRP with with different path number. The network size is 116 nodes. Data rate is 50 packets/sec

## Chapter 6

### Conclusion

Wireless sensor network is a key technology of ubiquitous networks, and networks that consist of independent sensor devices for monitoring physical or environmental conditions. In wireless sensor networks, the design of routing protocols has been focused on how to most efficiently use limited energy and how to cope with frequent network changes due to node error; since many sensor nodes are used for one-time purposes rather than permanently. In addition, much relevant research associated with data forwarding using multiple paths have been studied.

However, in most studies, they considered energy efficiency in terms of individual sensor nodes or suggested routing protocol that requires flooding a lot of control messages to find multiple paths. This makes it inefficient in terms of load-balancing because some nodes are included in several paths, even though multiple paths may be found.

In this thesis, I proposed a highly reliable, and load-balancing routing protocol to find node-disjoint multipaths which are efficient in a wireless sensor network that has severe resource constraints, such as a node's processing power, the amount of available memory, and network bandwidth. In this proposed protocol, a sink node can maintain the overall network information by broadcasting BNTQ and receiving BNTP from sensor nodes, enabling the sink node to compute the path to each sensor node based on the overall network information. By doing this, the number of control messages for setting up the gradient of network can be reduced significantly. In addition, the routing table size can also be reduced causing memory consumption of each sensor node to decrease, because the intermediate sensor nodes only have to maintain their uplink neighbor nodes' information. In addition, the load-balancing

created by the multiple paths decreases the load on specific nodes, so it helps keep packet loss rate low.

When a packet or data is forwarded, they are transmitted through sensor nodes using the entire route computed by the sink node. This prevents intermediate nodes from having to do an individual route discovery process and maintain a routing cache itself. In addition, in the case that a node error occurs, EMRP updates a new route easily because the sink node maintains the overall network information.

Through NS-2 simulation, this thesis observed the change in fluctuation (which is 37%) is 44% less than that of DCHT (which is 81%), although throughput is decreased as the network size increases. In end-to-end delay, EMRP was verified as having the lowest delay. Also, EMRP was found to have an average that was 1.5 times the number in DD (path number); has the largest path number when the network size is 116.

This thesis also observed how the number of uplink neighbor nodes that sensor nodes can maintain affects the path number. Intuitively, the more uplink neighbor nodes, the more paths can be founded. However, result show that there is no big difference when the uplink neighbor node number is between 12 and 16. The reason for this might be that one of the goals of EMRP is to find node-disjoint paths. Based on this, the optimal simulation parameters in finding multiple node-disjoint paths in EMRP are when network size is 116 and the uplink neighbor node number that a sensor node can maintain is 12.

The sink node maintains most of the routing table. The size of the sink's routing table is larger, but that is not a big issue because the sink node is less constrained in resources than sensor nodes. On the other hand, sensor nodes (which are more constrained with resources) maintain an average of less than 15 bytes. This fact shows that the memory consumption of each sensor nodes can be decreased.

Finally, this thesis observed how node-disjoint path numbers affect the packet delivery ratio and delay. Performance difference (which are which are the pakekt delivery ratio and end-to-end delay) between a single path being used and 2 paths being used is large in EMRP.

That is, the effect on performance (which are the packet delivery ratio and end-to-end delay) when using 2 paths is greater than a single path. However, when the path number is increased to 3 or 4, there is no the performance improvement. By observing this result, this thesis finds that the performance improvement in EMRP is not proportional to the path number.

EMRP is implemented under an assumption that there is no node movement. Therefore, a more efficient algorithm is needed with nodes that have large movement, and proper path selection strategy is needed as the network environment changes. This thesis leaves these issues open to future work.

## Bibliography

- [1] Y. M. Lu, and V. W. S. Wong "An Energy-Efficient Multipath Routing Protocol for Wireless Sensor Networks", International Journal of Communication Systems, Vol. 20, pp.747-766 , 2007.
- [2] S.B. Cho, "Sink-to-Sensors Reliable Transport based on Congestion Control in WSN(Wireless Sensor Network)," <http://home.postech.ac.kr/alpha44/cs499/intro.html>
- [3] "The Network Simulator - ns-2", <http://www.isi.edu/nsnam/ns/>
- [4] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed diffusion for wireless sensor networking," IEEE/ACM Transactions on Networking (TON), vol 11, no.1 , pp.2-16, 2003
- [5] C. E. Perkins and E. M.Royer, "Ad-hoc On-Demand Distance Vector Routing," <http://www.utdallas.edu/ksarac/courses/Papers/AODV.pdf>
- [6] Y. Li, S. Mao, S. S. Panwar, "The Case for Multipath Multimedia Transport overWireless Ad Hoc Networks", Proceedings of the First International Conference on Broadband Networks, 2004
- [7] K. ISHIDA, Y. KAKUDA, T. KIKUNO, and K. AMANO, "A Distributed Routing Protocol for Finding Two Node-Disjoint Paths in Computer Networks", IEICE TRANS. COMMUN., VOL.E82-B, No.6, 1999
- [8] S. Rai and D. P. Agrawal,"Distributed computing network reliability", IEEE Computer Society Press, 1990
- [9] A. S. Tanenbaum, "Computer networks(second edition), Prentice-Hall, 1988
- [10] K. Ishida, Y. Kakuda, and T. Kikuno, "A Routing Protocol for Finding Two Node-Disjoint Paths in Computer Networks", IEICE TRANSACTIONS on Communications Vol.E82-B No.6 pp.851-858, 1999
- [11] S.Venkatasubramanian and Dr.N.P.Gopalan, "A QoS-Based Robust Multipath Routing Protocol for Mobile Adhoc Networks", IACSIT International Journal of Engineering and Technology, Vol.1, No.5, 2009.
- [12] W. Lou, "An Efficient N-to-1 Multipath Routing Protocol in Wireless Sensor Networks", Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on, pp.664-672, 2005

- [13] A. F. Akyildiz, W. Su, Y. Sankarasubramainiam, E. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, 2002
- [14] C. E. Perkins and P. Bhagwat, "Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers," Applications, Technologies, Architectures, and Protocols for Computer Communication, p.234-244, 1994
- [15] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," Mobile Computing, vol. 353, pp.153-181, 1996
- [16] R. C. Shah, and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," Proc. IEEE Wireless commun. and Network Conf., vol. 1, Mar. 2002, pp. 350-55.
- [17] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking", ACM/IEEE Transactions on Networking, Vol. 11, pp. 2-16, 2002
- [18] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," ACM SIGMOBILE Mobile Computing and Communications Review , vol 5, p.11-25, 2001
- [19] S.J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad hoc Networks", Communications, 2001. ICC 2001. IEEE International Conference on, vol.10, pp.3201-3205, 2001
- [20] S. Li, R. kisore, C. Liu, and Alvin Lim, "Efficient Multi-path protocol for wireless sensor networks," International Journal of Wireless & Mobile Networks(IJWMN), Vol.2, No.1, 2010
- [21] D. Ganesan, R. Govindan, S. Schenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks", ACM SIGMOBILE Mobile Computing and Communications Review, Vol. 5, Issue. 4, 2001.
- [22] R. Shah and J. Rabaey, "Energy aware routing for low energy ad hocsensor networks," in Proc. of IEEE WCNC'02, Orlando, FL, March 2002, pp. 350-355.
- [23] K. Wu and J. Harms, "On-demand multipath routing for ad hoc networks," in Proc. of European Personal and Mobile Communications Conference (EPMCC), Vienna, Austria, Feb. 2001.
- [24] N. V. Yedida, and R. R. Challa, "Performance Comparison of AODV, DSR and OLSR Routing protocols in Static Scenarios," [http://www.ucslouisiana.edu/rxc2763/Report\\_Protocols.pdf](http://www.ucslouisiana.edu/rxc2763/Report_Protocols.pdf)
- [25] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed Diffusion for Wireless Sensor Networking", IEEE/ACM Transactions on Networking, 2003

- [26] S.R. Jung, J.H. Lee, B.H. Roh, "An Optimized Node-Disjoint Multi-path Routing Protocol for Multimedia Data Transmission over Wireless Sensor Networks," *ispa*, pp.958-963, 2008 IEEE International Symposium on Parallel and Distributed Processing with Applications, 2008
- [27] G. Jayakumar and G. Ganapathy, "Performance comparison of Mobile Ad-hoc Network Routing Protocol", *IJCSNS International Journal of Computer Science and Network Security*, Vol.7, No.11, 2007