IJCTA, 9(25), 2016, pp. 269-278 ISSN: 0974-5572

Wisem: Design and Development of a Wireless Sensor Mote For Providing Adaptive Qos-based Route Management

V. Tamizhazhagan*, R. Saminathan and A.M. Sameeullah*****

ABSTRACT

Research in the field of Wireless Sensor Network (WSN) has always been a demand specifically toward quality of service (QoS) and data aggregation primarily due to implementation in scientific and application domains. Though various QoS schemes have been demonstrated for session management in WSN and data aggregation, this survey discusses simulated approach. This paper discusses the design and development of WSN, which is capable of supporting QoS through an adaptive route management approach. The scheme supports variable data management and route establishment. WISEM has been designed using PIC microcontrollers with multiple sensors. Data transfer is established using adaptive route discovery and management approaches. Experiment is conducted using real-time and simulated approaches and compared with existing schemes.

Keywords: WISEM, QoS, WSN, Route Management, PIC Microcontroller, Sensors.

1. INTRODUCTION

The sensor nodes can be deployed in a self-organizing environment by establishing radio communication paths from various sources to a sink, which is shown in Fig. 1. The environmental sensing devices [1,2] are low-powered devices in terms of volts, which possess a microcontroller for information processing, with a storage device and antenna for radio communication. Few common sensors include sensing environmental parameters such as temperature, humidity, light intensity, and more.

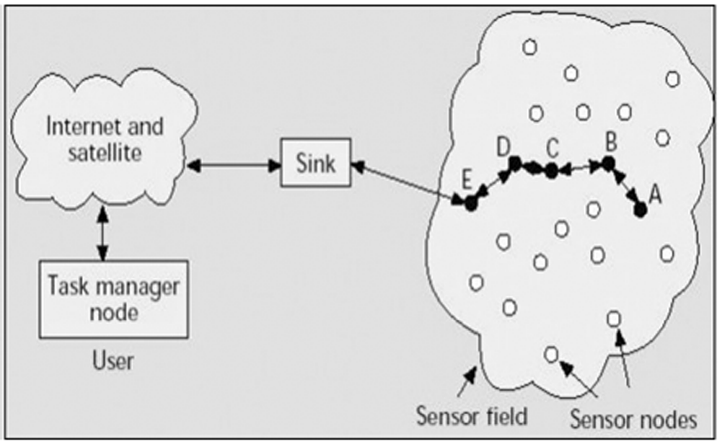

Figure 1: Sensor nodes scattered on field

---

* Assistant Professor, Department of Computer Science & Engineering, Annamalai University Annamalainagar–608 002, Tamil Nadu, India, *Email: tamizh5956@gmail.com*

** Assistant Professor Department of Computer Science & Engineering, Annamalai University Annamalainagar–608 002, Tamil Nadu, India, *Email: samiaucse@yahoo.com*

*** Professor Department of computer Science & Engineering, Sri Sai Ram Institute of Technology, West Tambaram, Chennai-44, *Email: Sameeullah.cse@sairamit.edu.in*

Wireless sensor network (WSN) devices are devised to be resource limited, and have minimal process capability since they possess low processing speed, storage capacity, and network communication bandwidth with limited mobility. WSN nodes are expected to operate for a long period of time, since the nodes are powered by battery, and hence should maintain their energy resources by limiting their tasks or processes.

An important characteristic of WSN is the sensor nodes have significant processing capability to compute and categorize their services based on context aware nature of system behavior [3]. Nodes have to organize themselves, administer and manage the network all together, and it is much harder than controlling individual devices. Any change in the physical environment where a network is deployed makes also nodes experience wide variations in connectivity which influences the system organization.

The major factors that complicate the protocol design [4,5] for WSNs can be listed as below:

✓ Fault tolerance: the necessity to sustain sensor networks functionalities without any interruption, after a node failure.

✓ Scalability: the possibility to enlarge and reduce the network.

✓ Deployment: given a certain environment it should be possible to find the suitable deploying location for each sensor.

✓ Power management: the network lifetime needs to be maximized.

✓ Security issues: to provide data security and session security.

In spite of a greater effort required for building a WSN, the interest in this technology is increasing. Recently, major research works carried out on WSN with support for numerous applications in the industrial and commercial field have encouraged lot of responses among users to discover, discuss, and exploit all possible system potentialities.

2. MOTE

MOTEs can be considered as processing nodes or computers which work in a cooperative way to form networks as shown in Fig 2. MOTEs list out specific requirements such as small in size, energy efficient, multifunctional, and being wireless. Collections of multiple MOTEs communicate together to achieve a common or specific goal. A MOTE [6] can be defined as a miniature of WSN which performs all the tasks of sensor networks as well as supports routing capabilities.

MOTEs adopt a high interference in variable environments and surroundings. Hence the maximum broadcast range of MOTE does not exceed 20-30 meters. It should be understood that the MOTEs placed

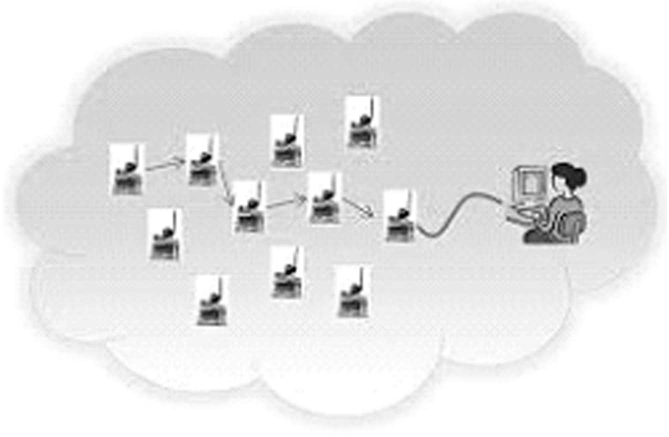

Figure 2: MOTE structure

in an open ground or behind trees can also communicate with others. The radio frequency range of an MOTE is designed to support in minimal energy consumption and hence follows a limited broadcast range.

WSNs [7] designed as MOTEs collect data from surrounding region or environment, gather events happening, as well as perform actions according to the data collected. The action could be a movement, alarm setting off, or recording the data in a file. All the actions change the scenario that the MOTE adopts, hence causing other changes. The interconnectivity between the MOTEs effects the changes and also affects each MOTE, as well as the data collected by the MOTE. Each MOTE routes its data to the parent MOTE or coordinator, while the parent MOTE connects to an internet server or high processing system that performs the functionality for which the MOTEs are designed.

3. WISEM: MOTE

Though the WISEM kit is developed for effective routing among large networks, WISEM can be defined as an MOTE with the following properties (Table 1):

Table 1
Properties of WISEM

Program flash memory	32 KB (16K × 16)
Measured flash memory	512 kbytes
Configuration of EPROM	256 × 8 bytes
Number of IO	15 slots
Data rate	256 kbits/s
Frequency of communication :	2.4 GHz
Size (mm)	58 × 32 × 7
Operating voltage	2.2 V to 3.3 V

WISEM is developed using a PIC 18F microcontroller [8], which is considered as a simple, cheap, and effective microcontroller for industry-specific applications. It can operate at a low voltage of 2.2 V for simple applications. It possesses a maximum of 15 slots for carrying out multiple services and handles event-specific tasks based on actuations of users. The kit has been developed using C programming which captures multiple devices for interactivity. The nodes are heterogeneous and hence multiple sensors can be built in the kit to actuate as per the need of application. Due to good flash and programming memory, the kit can handle large datasets to be stored and monitored for applications such as habituate monitoring and agricultural farm-specific environment.

3.1. WISEM (W-MOTE) Architecture and Functionality

W-MOTE works on software that manages a fixed array of wireless sensor network nodes fitted with wireless interface backchannel boards allowing data logging and transmission of data over multiple nodes. WISEM MOTE sensor network test-bed supports quality of service (QoS) and addresses challenges through wireless interconnectivity as an interface. The kit supports automated test-bed programming and gathers log data generated by experiments into a PC or a server as a persistent database [9]. Users who conduct tests can retrieve sensor data as source from the database. The database maintains consistent data collected at various iterations including sensor data, time, sensor name, and service adopted from test-bed. W-MOTE is attached with different software components, such as

a) Data logger and reading sensor data

b) Wireless network interface for communication

c) Port/USB interface for collecting the sensor data

The architecture is based on a multi-tiered protocol structure, which relies on ZigBee [10,11] (Network and Application Layers) and 2.4 GHz, IEEE 802.15.4 (Physical and Data Link Layers) communication protocol standards, which serves as a network backbone for data transfer and routing phenomenon (Table 2). To understand and analyze the behavior of the WSN network, an application-based test-bed is being developed with the objectives of implementing, assessing, and validating WISEM architecture (Fig. 3).

Two major functionalities of WISEM support data aggregation and QoS management factors for variable services. WISEM controls data gathering and analysis works through interrupt handler functionality, where nodes collect environmental parameters through sensors, process them through external processing units, support QoS through route analysis.

WISEM adopts the stack architecture structure [12,13], where each node is equipped with sensory interface setup which can interface multiple sensors such as temperature, humidity, day light intensity, and humidity, which are highly sensitive to variable environments.

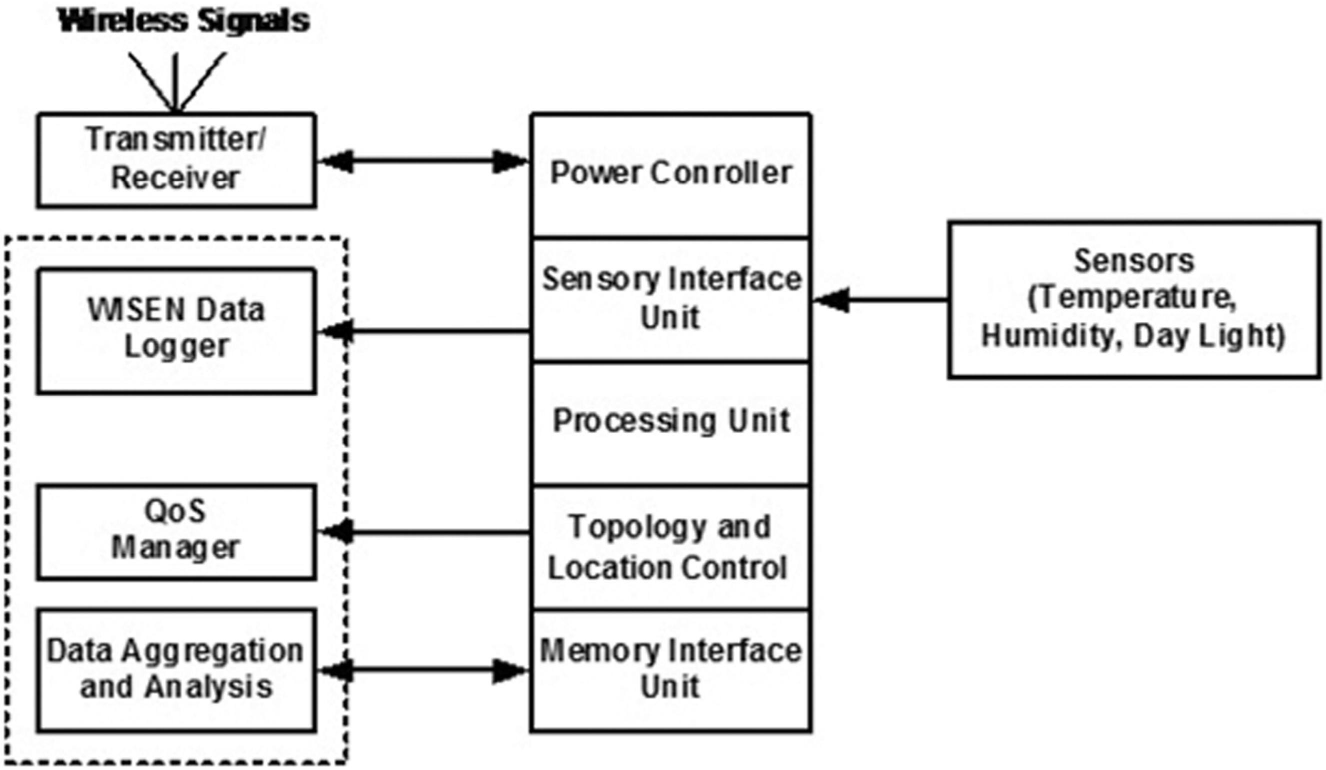

Figure 3: WISEM architecture and functionality

Table 2
WISEM - components

Sensor node name	*WISEM*
Microcontroller	PIC 8F
Transceiver	IEEE 802.15.4
Data memory	10 KB RAM
External memory	EPROM
Transfer data rate	100 Kbps–256 Kbps
Communication range	10-25 feet

4. CONFIGURATION OF WISEM NODE – INDOOR EXPERIMENT

WISEM test bed creation to define QoS and efficient route control approach is arranged indoor within a room of 20 × 25 meters as shown in Fig. 4. WISEM nodes are placed with an inter-spacing of 5_10 meters intermittently. Based on the naming of nodes, node ‘a’ is initially defined as the source node (WISEM_1), while node ‘d’ as the receiver node. The sensor data transmitted over the network generate traffic over the route created with packet streams whose mean packet sizes possess 400 bytes (including TCP/UDP, IP headers). The data transmission over generates an approximation of 3500_5000 bytes of data traffic over multi-nodal network.

Experiments were conducted over varying sets of 5 WISEM nodes in a rectangular room of 20 x 25 feet surrounded by walls of 1 inch of brick mortar. A single node acts as a coordinator to control, create,

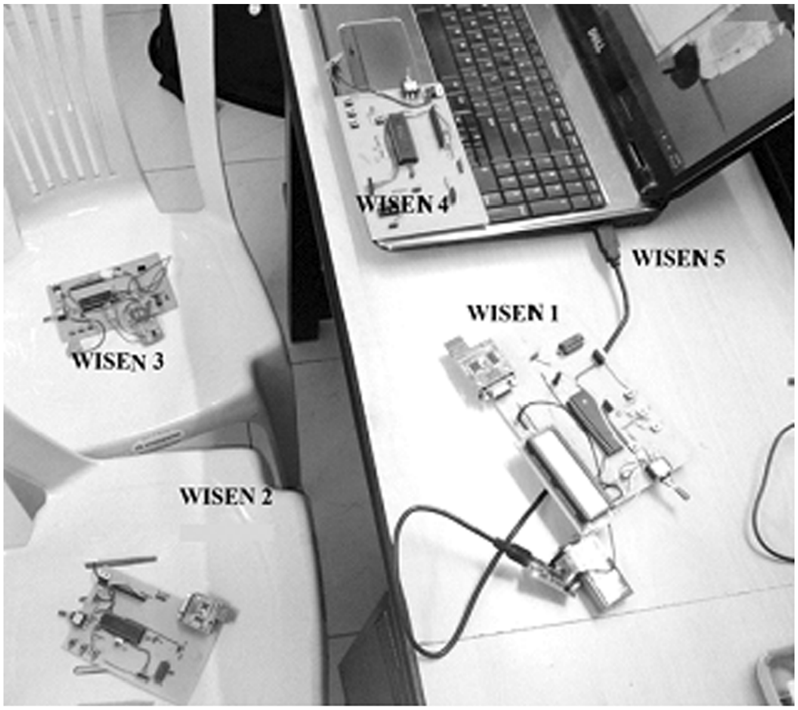

Figure 4: WISEM deployment for indoor experiments

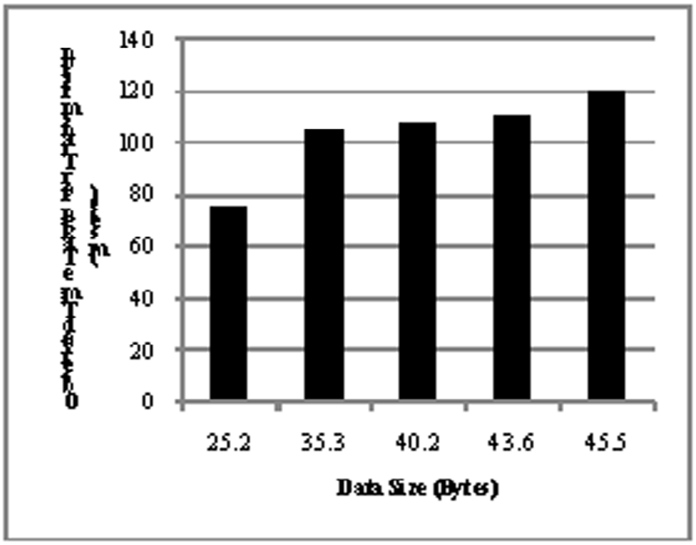

Figure 5: Observed time taken for transmission of different data

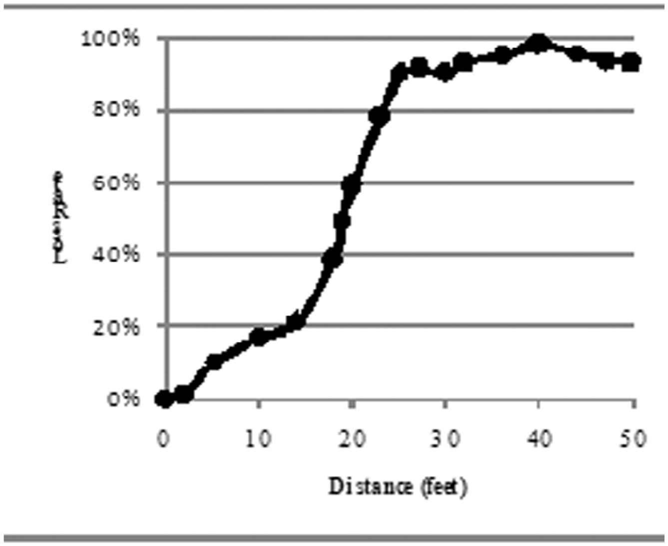

Figure 6: Observed loss rate

setup, and run the routing algorithms, while other nodes can inter-communicate with each other. The nodes do transmit at variable bit rates due to their varying link capacity, the route selection process gets updated.

4.1. Performance Analysis

The routing algorithms are implemented in C++ and share a common set of classes. The C++ classes include events such as create route, discover route, and unicast/multicast, queue scheduling and log maintenance support. Based on these key features, algorithm-specific code is confined to packet handler classes which process incoming packet control and data packets. The timer handler classes process timed actions (such as route expiration), the logging classes handle log algorithm events, and utility classes serialize and un-serialize the request (W-REQ) and reply (W-RLY) packets (Fig. 5).

Fig. 6 explains the observed loss rate among 5 WISEM nodes based on the distance between nodes. It could be observed that as the distance between nodes increases the session established between the nodes breaks and hence data loss increases. As the number of nodes is being increased, the possibility of data loss could be minimized since more forwarding nodes could be added between the source and the destination.

5. EXPERIMENT SETUP - AGRICULTURAL LAND

Research developments in the field of agriculture or environmental pollution control focus on adaptation of wireless sensor fields, which have suggested precision in farming and improving the yield.

WISEM equipped with soil moisture and temperature sensor has been deployed as data loggers used in the agriculture field to gather soil moisture data at regular intervals. The data are gathered through a multi-hop routing process of WISEM, and finally through a coordinator node, which can be connected to a laptop or a server to disseminate the collected data.

The WISEM nodes are deployed in an agricultural farm land measuring 10 feet across 20 feet, since the nodes are left in open land as shown in Fig. 7. WISEM kit is connected to an independent battery source of 9 V, so that it can work in the best environmental situations. The soil moisture sensor used for gathering data is dipped into soil such as wet soil, dry, and rocky soil for gathering the moisture level as temperature as shown in Fig. 8.

Flash memory is divided into 16 blocks, where each block has 256 pages. The sensor data send 100 bytes of data per session. The sensor kit has been deployed for 3 consecutive days. Based on WISEM

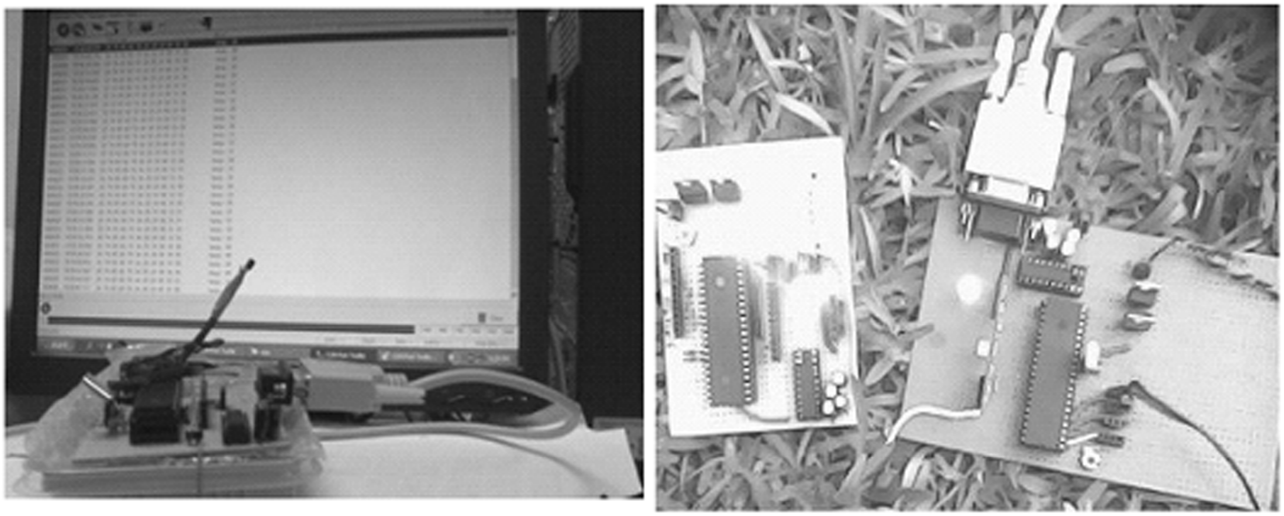

Figure 7: WISEM deployment for agricultural land

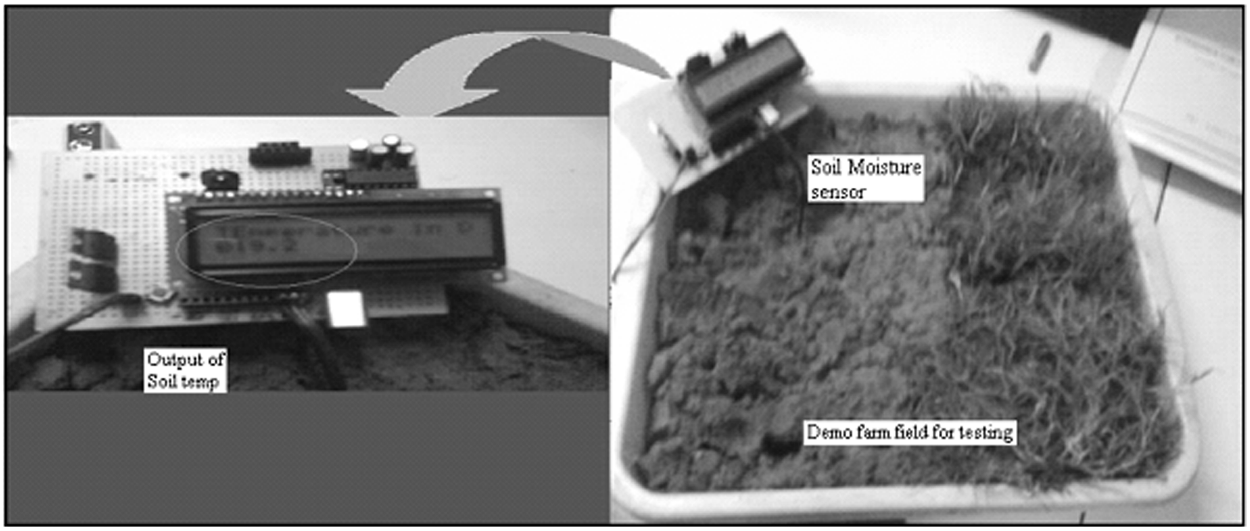

Figure 8: Experimental setup for farm field testing

sensor node's id, the page is allocated to storage of sensing node. The coordinator node receives the WISEM node id based on the message type (request, reply, and store), then the data are stored in flash.

5.1. Performance Analysis

Fig. 9 shows the throughput analysis of the difference in time taken to send the data from a WISEM node to receiver WISEM node and demonstrates the minimal delay achieved as time increases. The observed throughput of network for data transfer service is improved and hence accepted as per experimental requirements.

6. SIMULATED APPROACH

To understand the behavior of WISEM for large set of nodes, the routing algorithm is simulated using ns2 and its performance is compared with other traditional WSN routing algorithms. WSN routing protocol

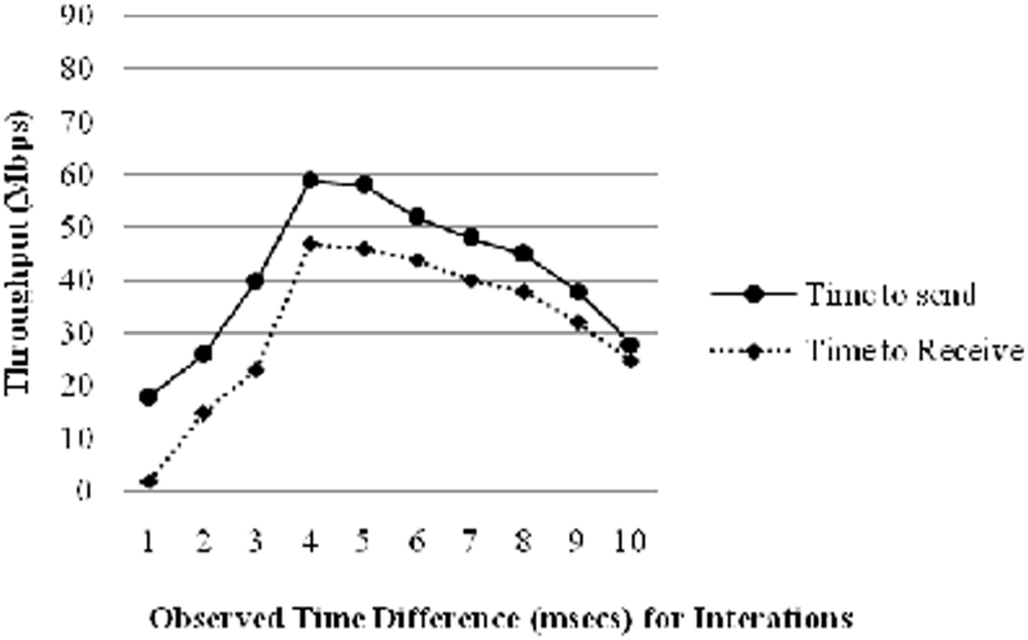

Figure 9: Throughput analysis

schemes such as SPIN and LEACH are simulated over ns2. WISEM outperforms SPIN significantly in terms of routing load balancing and bandwidth overhead primarily in terms of low mobility, as well as manages LEACH in terms of throughput over a large network (30 nodes). However, its performance deteriorates slowly when the number of nodes '*n*' is increased and its bandwidth usage gets overloaded.

During degradation of QoS discovery process, source node indicates this information to neighboring nodes, which supports in identifying the source to identify multiple routes to its destination. This enables the source node to switch to cached routes in case of route break, which significantly reduces the possibility to restart a route discovery process and completely over again. However, under stressful situations, the cached routes are considered as invalid status, which thus reduces unnecessary delay and handles network traffic effectively.

The data generated from experiment demonstrate the performance of WISEM with varying number of packets per source node. Since each WISEM node creates new packet with a fixed sampling interval, the total number of packets created by each node is determined by the duration of the experiment. The traffic generated suggests that the basic performance properties of the node use the CBR traffic model. It has to be noted that the default experimental settings are used in the traffic test, except the number of packets created per node, which varies in experimental approach. The experiment starts with generating 50 packets per node for each time interval, which equals to 250 packets in total in network. As time interval moves on the total number of packets created increases by 40% to 100 packets per node, such that 500 packets and finally the number of packets created is increased by 120% to 200 packets per node, which are 1200 packets in total. The experiment results are shown in Fig. 10 and Fig. 11 which suggest the loss of data observed over simulated test run and end to end delay.

The simulated test-bed result in Fig. 11 shows the observed throughput for 30 WISEM nodes. The test is carried out on similar type of data service, transferred between the source node and receiver nodes. The performance of WISEM is compared with other schemes such as LEACH, SPIN, and HEED. The WISEM scheme is observed to show an improved throughput with an average of 1080Mbps of transmitted data whereas LEACH and SPIN show an average of 640 Mbps and 750 Mbps, respectively. WISEM is able to show a higher performance due to the effective usage of cached routes.

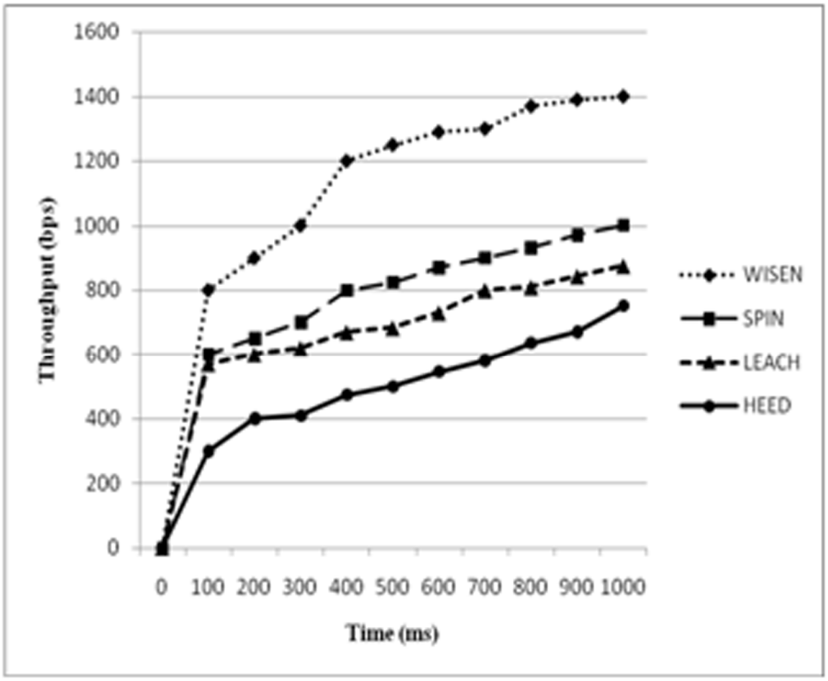

Figure 10: Observed throughput for 30 nodes

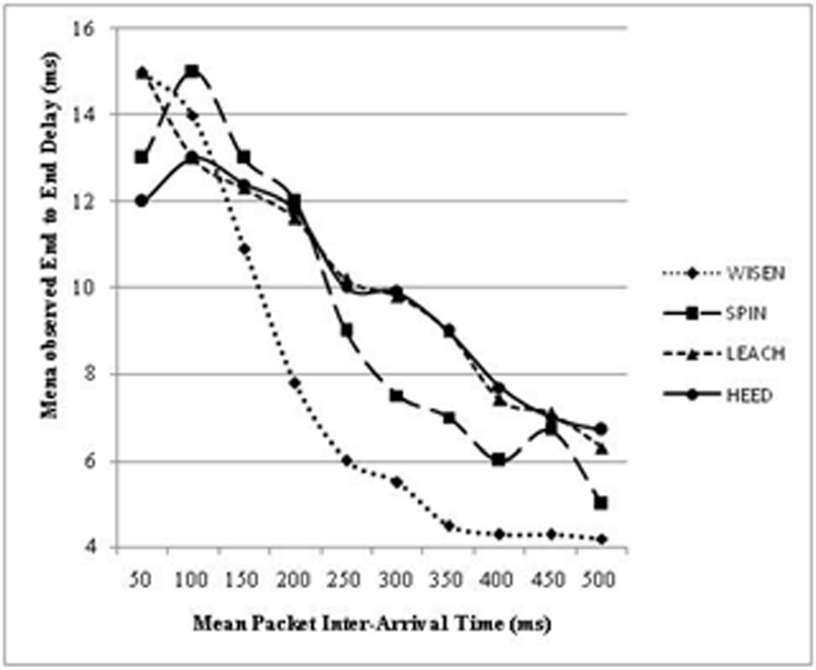

Figure 11: Delay observed over 30 WSN nodes

7. CONCLUSION AND FUTURE WORK

WISEM scheme focuses on a cost effective and minimal processing-based WSN node implementation. WISEM focuses on QoS routing and session management approaches as an algorithmic approach. This paper focuses on five WISEM nodes developed and their performance is compared with existing routing schemes such as SPIN and LEACH. The work has been carried out in experimental ends as a real-time approach as well as simulated using sensorsim patch in ns2 simulator. The performance shows that WISEM has minimal packet loss and delay with improved throughput over HEED, SPIN, and LEACH approaches.

Present implementation of WISEM only supports general configuration of nodes but specific parameter node configuration cannot be carried out using coordinator nodes, hence this approach does not support heterogeneous network environments that need non-trivial distribution of multiple nodes of differing hardware capabilities which requires middleware diverse network stacks to be deployed.

REFERENCES

[1] Akyildiz, W Su, Sankarasubramaniam Y, And Cayirci E, “A Survey On Sensor Networks,” Ieee Communications Magazine, 40 (8),102–114(2002).

[2] Culler D, Estrin D, Srivastav M, “Introduction: Overview Of Sensor Networks,” Ieee Computer, 37(8), pp. 41–49(2004).

[3] Alexander Becher, Zinaida Benenson, Maximillian Dornseif, Tampering With Motes: Real-World Physical Attacks On Wireless Sensor Networks, Security In Pervasive Computing, Lecture Notes In Computer Science, Volume 3934, pp. 104-118(2006).

[4] Hill J And Culler D, “Mica: A Wireless Platform For Deeply Embedded Networks”, Ieee Micro,22(6), pp. 12–24(2002).

[5] Chong C Y And Kumar S P, “Sensor Networks: Evolution, Opportunities, And Challenges,” Proceedings Of The Ieee, August 2003, 91(8), pp. 1247–1256.

[6] Gian Pietro Picco, Programming Wireless Sensor Networks: Fundamental Concepts And State Of The Art, Acm Computing Surveys, 43,(3), (2011).

[7] Ho L, Moh M, Walker Z, Hamada T, Su C, “A Prototype On Rfid And Sensor Networks,” Acm Sigcomm Workshop On Experimental For Elder Healthcare: Progress Report (2005).

[8] Dutta P, Dawson-Haggerty S, Chen Y, Liang C J, And Terzis A “Design And Evaluation Of A Versatile And Efficient Receiver-Initiated Link Layer For Low-Power Wireless,” 8th International Conference On Embedded Networked Sensor Systems(Sensys10), Zurich, November 2010, pp. 1–14

[9] Barardo L, Oliveira R, Pereira M, Macedo M, And De Lisboa N, “A Wireless Sensor Mac Protocol For Bursty Data Traffic,” 18th Ieee International Symposium On Personal Indoor And Mobile Radio Communications (Pimrc07), Athens, Greece, September 2007, pp. 1–5.

[10] Chatterjea.S, Van Hoesel.L, Havinga.P(2004), “Ai-Lmac: An Adaptive, Information-Centric And Lightweight Mac Protocol For Wireless Sensor Networks, Second International Conference On Intelligent Sensors,” Sensor Networks And Information Processing(Issnip04), December 2004, pp. 381–388.

[11] Gibbons P, Karp B, Ke Y, Nath S, Seshan S, “Irisnet: An Architecture For A Worldwide Sensor Web, Ieee Pervasive Computing”, 2003, 2(4), pp. 22–33.

[12] Heinzelman W, Chandrakasan A, And Balakrishnan H, “An Application -Specific Protocol Architecture For Wireless Micro Sensor Networks,” Ieee Transactions On Wireless Communications, 1(4), pp. 660–670 (2002).

[13] Akkaya K, Younis M, “A Survey On Routing Protocols For Wireless Sensor Networks. Ad Hoc Networks.” 3, 325–349 (2003).