TURKISH NAVAL ACADEMY
NAVAL SCIENCE AND ENGINEERING INSTITUTE
DEPARTMENT OF COMPUTER ENGINEERING

POWER AWARE MANY TO MANY DATA CENTRIC ROUTING IN WIRELESS SENSOR AND ACTUATOR NETWORKS

MASTER THESIS

TARIK DÖĞÜŞCÜ

Advisor: Assist.Prof. Vedat Coşkun
Co-Advisor: Assoc.Prof. Erdal Çayırcı

İstanbul, 2004

CERTIFICATE OF COMMITTEE APPROVAL

POWER AWARE MANY TO MANY DATA CENTRIC ROUTING IN WIRELESS SENSOR AND ACTUATOR NETWORKS

Submitted in partial fulfillment of the requirements for degree of

MASTER OF SCIENCE IN COMPUTER ENGINEERING
from the

TURKISH NAVAL ACADEMY

Author : ______________________________
Tarık Döğüşcü

Defense Date : 26 / 07 / 2004
Approved by : 26 / 07 / 2004

____
Assist.Prof. Vedat Coşkun (Advisor)

____
Assoc.Prof. Erdal Çayırcı (Co-Advisor)

____
Prof. M.Yahya Karslıgil (Defense Committee Member)

____
Prof. Şebnem Baydere (Defense Committee Member)

____
Assist.Prof. Gökhan Yavuz (Defense Committee Member)

ABSTRACT (TURKISH)

TELSİZ DUYARGA VE AKTİVATÖR AĞLARINDA GÜÇ BİLİNÇLİ ÇOKTAN ÇOĞA VERİ MERKEZLİ YOL ATAMA

Anahtar Kelimeler : Çoktan-çoğa İletim, Güç Bilinçli İletim, Çoklu Haberleşme, Telsiz Duyarga ve Aktivatör Ağları.

Bu tezde, telsiz duyarga ağlarında kullanılan veri yol atama tekniklerini ve bu konudaki araştırma alanlarını gözden geçiriyoruz. Telsiz duyarga ve aktivatör ağları hakkında bir temel oluşturduktan sonra, duyarga ve aktivatör ağları tarafından ihtiyaç duyulan yol atama teknikleri sistemimize bir altyapı oluşturmak maksadıyla sunulmuştur. Önerdiğimiz protokolde aktivatörler ihtiyaç duydukları sezilen veriler için bir görev kayıt mesajı yayınlarlar. Görev kayıt mesajı ağ içinde yayılırken mesajı taşıyan algılayıcı düğümler, her aktivatör ve hissedilecek görev için çoklu ağaç inşa ederler. Bir hissedilen veri paketini alan bir düğüm, ilgili görev için kayıt mesajı gönderen komşu düğümlerine bu paketi gönderir. Önerdiğimiz sistem aktivatörlerin sayılarındaki, yerlerindeki ve görev ilgilerindeki değişimler ile düğümlerdeki güç değişimlerine kendini adapte edebilmektedir.

ABSTRACT (ENGLISH)

POWER AWARE MANY TO MANY DATA CENTRIC ROUTING IN WIRELESS SENSOR AND ACTUATOR NETWORKS

Keywords : Many to Many Routing, Power Aware Routing, Multicast Communication, Wireless Sensor And Actuator Networks.

In this thesis, we review the data routing techniques and related issues in wireless sensor networks. After supplying a basic background about the wireless sensor and actuator networks, data routing techniques required by sensor and actuator networks is introduced to constitute an infrastructure of our scheme. In our protocol actuators broadcast a task registration message for the sensed data that they need. While the task registration message is being disseminated through the network, the sensor nodes relaying the message construct a multicast tree for each actuator and sensing task. A node that receives a sensed data packet forwards it to the neighboring nodes that relay a registration message for the related task. Our scheme can also adapt itself to the changes in the number, the locations, and the sensing task interests of the actuators, and the power available in the nodes.

DISCLAIMER STATEMENT

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Turkish Navy, Naval Academy, and Naval Science and Engineering Institute.

DEDICATION

I would like to thank my mother, Manolya, and my sister, Tuğba, for all the support they have given me. Their constant encouragement, and belief in my abilities has been invaluable. I would not be where I am without them.

I dedicate my thesis to them.

ACKNOWLEDGMENTS

I wish to extend my sincere appreciation to my advisor Assist.Prof. Vedat Coşkun for his guidance, assistance, encouragement, and friendship.

I would like to extend my sincere gratitude and appreciation to my thesis other advisor, Assoc.Prof. Erdal Çayırcı, for his intelligent supervision, constructive guidance, friendship, and his encouragement for my endeavors, personal and academic. I feel very fortunate to have had the opportunity to benefit from his brilliant insights and invaluable comments, which have been a great source of inspiration for me.

I would like to thank Prof. Süleyman Özkaynak and the Naval Science and Engineering Institute for providing me with this research opportunity and providing all kind of support for the growth and the quality of our Computer Engineering Department.

Finally, I would like to thank all my friends who help me when I need.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF FLOW CHARTS

LIST OF ABBREVIATIONS, ACRONYMS, AND SYMBOLS

AC	Address-Centric Protocol
AODV	Ad Hoc Demand Distance Vector
CDMA	Call Division Multiple Access
DCP	Data-Centric Protocol
DSNs	Distributed Sensor Networks
ID	Identification
GAF	Geographical Adaptive Fidelity
GEAR	Geographical And Energy Aware Routing
GPS	Global Positioning System
LEACH	Low Energy Adaptive Clustering Hierarchy
MANETs	Mobile Ad-Hoc Networks
MCU	Microcontroller Unit
MECN	Minimum Energy Communication Network
M2MI	Many To Many Invocation
MEMS	Micro-electromechanical Systems
OS	Operating System
PEGAGIS	Power-Efficient Gathering In Sensor Information Systems
QoS	Quality Of Service
RS	Random Sources
SAFE	Sinks Accessing Data From Environments
SAR	Sequential Assignment Routing
SEAD	Scalable Energy Efficient Asynchronous Dissemination Protocol
SINA	Sensor Information Networking Architecture
SMECN	Small Minimum Energy Communication Network
SPIN	Sensor Protocols For Information Via Negotiation
SPT	Shortest Paths Tree
SQTL	Sensor Query And Tasking Language
TDD	Task Dissemination Duration
TDMA	Time Division Multiple Access
TTDD	Two Tier Data Dissemination
TEEN	Threshold Sensitive Energy Efficient Sensor Network Protocol
TORA	Temporally Ordered Routing Algorithm
WINS	Wireless Integrated Network Sensors
WSN	Wireless Sensor Network
WSAN	Wireless Sensor And Actuator Network

I. INTRODUCTION

A. WIRELESS SENSOR AND ACTUATOR NETWORKS

The physical world presents an incredibly rich set of input modalities, including acoustics, image, motion, vibration, heat, light, moisture, pressure, ultrasound, radio, magnetic, and many more exotic modes [1]. Enhancing the human-to-physical world interaction in daily life raises the importance of smart environments like sensor networks. As a result of the recent advances in the micro-electro-mechanical systems (MEMS) and the wireless communications, we frequently encounter tiny sensor nodes at homes, in workplaces, supermarkets, plantations, oceans, streets, highways, forests and special military applications. These sensor nodes which can be connected via a wireless network, constitute wireless sensor networks (WSNs) which represent a new paradigm for extracting data from the environment and enable the reliable monitoring of a variety of environments for applications that include surveillance, machine failure diagnosis and chemical/biological detection...etc [2].

Wireless sensor networks which consist of many sensor nodes differ from traditional networks in several ways [2,3,4,5,6];

- They require multi-hop communication since hundreds or thousands of cheap sensor nodes are dispersed or deployed in sensor field,
- Nodes may not have global identification because of the large number of nodes,
- They use many-to-one flows in which sensed data is conveyed to the single point called sink,
- Nodes in network have low battery capacity,

- Nodes use wireless and broadcast communication paradigm with low-power radio frequency,
- Nodes have memory and computational constraint,
- Network topology changes very frequently,
- Nodes are prone to failure,
- Nodes are very cheap,
- Nodes may be mobile or stationary.

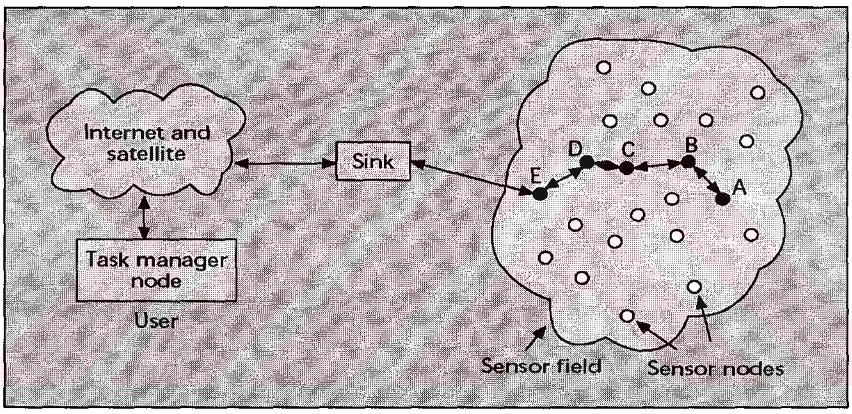


Figure 1. Wireless Sensor Network [3].

An example of WSN is shown in Figure 1. The primary function of sensor networks is the collection and delivery of sensory data. Sensor nodes obtain the data from their sensors that sense any phenomenon from the environment. This data is routed to the sink by a multihop infrastructureless architecture. The sink acts like a gateway of network. Therefore, the end user sends a query to the WSN over the sink and receives a respond from it. Until now, WSNs and their applications have been developed without considering a management solution [6]. For this reason, we try to obtain more autonomous network by scattering in sensor field many actuators and sensor nodes together. Collection of the data from the environment is similar to that in all WSNs. But the decision makers are different and they are the actuators. The sensor nodes convey the data toward the actuators. The actuators decide to make any action based on this data and apply it via their attached devices. Hence, wireless sensor and actuator networks (WSANs) is different from WSNs. An example of WSAN is shown in Figure 2.

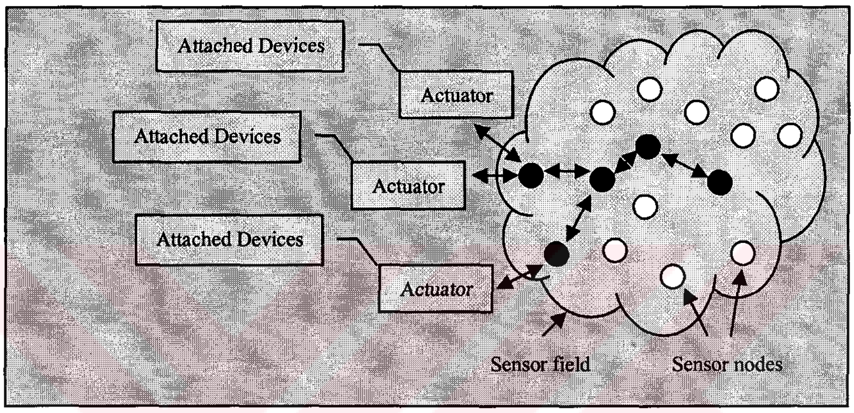


Figure 2. Wireless Sensor and Actuator Network.

B. PROBLEM DEFINITION AND PROPOSED SCHEME

Energy constraints may be the most important design criteria for sensor networks in which multi hop communication is done. Therefore, routing protocols must select the best path to minimize the total power needed to route data packets on the network and to maximize the lifetime of all nodes. Traditional routing protocols developed for wireless ad hoc networks can not fit the requirements of sensor networks [3]. In addition, WSN routing protocols do not fit the requirements of WSANs, either. Because, an actuator may fetch the data from multiple sensor nodes and a sensor nodes may feed multiple actuators in WSANs. So, the main focus of this thesis starts at this point. In this thesis, a new power aware many to many data centric routing protocol is introduced for WSANs. By using our protocol, the sensor nodes construct a multicast tree for each actuator and sensing task which the actuators interest in. Thus, the data packets are relayed efficiently by sensor nodes to the actuators.

C. CONTRIBUTION OF THIS THESIS

Sensors and actuators have undergone a revolution with the emergence of microelectromechanical systems technology, in which mechanical devices such as accelerometers, barometers and movable mirrors are constructed at minute size on a silicon chip using lithographic process similar to those for integrated circuits [1]. As a result of these advances, we encounter the smart systems in our daily life frequently. We introduce a novel practical protocol which does not require more computational memory capacity for such systems. In our protocol, the routing decisions are taken easier to convey the data to the multiple actuators. In addition, the power level of the nodes is considered to enhance the lifetime of the network. Meanwhile, our protocol is data centric since this approach best fits the characteristics of many WSAN applications like as in WSNs.

D. STRUCTURE OF THIS THESIS

In section I, we briefly present the architecture of WSNs and WSANs. In section II, we review the routing techniques in WSNs that have been an active research area in the context of sensor networks and we examine why they are not suitable for WSANs. Then, we mention the architecture of WSANs and design criteria for routing in a WSAN. After we discuss why we select power aware, data centric and many to many routing, we introduce a new scheme that puts forward an efficient routing protocol for WSANs in section III. We evaluate the performance of our protocol in section IV. At last, we conclude our thesis in section V.

II. BACKGROUND INFORMATION AND RELATED WORK

A. WIRELESS SENSOR NETWORKS

Advances in digital electronics, embedded systems and wireless communications leaded the way to a new class of ad hoc networks, namely wireless sensor networks (WSNs). WSNs are becoming an increasingly important technology that will be used in a variety of applications such as environmental monitoring, infrastructure management, public safety, medical, home and office security, transportation and military [6]. They differ from conventional network systems in many aspects. WSNs usually involve a large number of spatially distributed, energy-constrained, self-configuring and self-aware nodes. Furthermore, they tend to be autonomous and require a high degree of cooperation and adaptation to perform the desired coordinated tasks and networking functionalities. As such, they bring about new challenges and design considerations, which go much beyond conventional network systems.

An example of a sensor network application is a group of mobile decontaminating robots with wireless computing devices that use the sensor network for monitoring the radioactive contamination level of some region [7]. Another example is a group of stationary nodes that use for habitat monitoring in a forest.

1. Design Criteria of WSNs

The infrastructure of a WSN consists of the sensor nodes and their current deployment status [8]. More specifically, the infrastructure is influenced by the characteristics of the sensors (e.g., sensing accuracy, memory size, battery life, transmission range) and deployment strategy (e.g., sensor density, sensor location, sensor mobility). Basically, there are three types of nodes [6]; *common nodes* responsible for collecting sensing data, *sink nodes* responsible for receiving, storing and processing data from common nodes and *gateway nodes* that connect sink nodes to external entities called *observers* which may communicate with the sink via internet or satellite.

The observer(s) interests in the phenomenon are queries from the observer(s) about the phenomenon as approximated by the distributed data that the sensors are capable of sensing [8]. These queries could be static (the sensors are preprogrammed to report data according to a specific pattern) or dynamic.

Furthermore, the network protocol that is used by all nodes in WSNs is responsible for creating paths and accomplishing communication between the sensor nodes and the observer(s) [8].

By studying the communication patterns systematically, the network designer will be able to choose the infrastructure and communication protocol that provide the best combination of performance, robustness, efficiency and deployment cost [8]. Therefore, design factors are important as they serve as a guideline to design a protocol or an algorithm for sensor networks [3,9];

- *Fault tolerance*: Some sensor nodes may fail or be blocked due to lack of power physical damage or environmental interference. The failure of sensor nodes should not affect the overall task of the sensor network.

- *Scalability*: The number of sensor nodes deployed in studying a phenomenon may be in the order of hundreds or thousands. Depending on the application the number of nodes can be more. New schemes must be able to work with this many number of nodes.

- *Production costs*: Since sensor networks consist of a large number of sensor nodes, the cost of a single node is very important to justify the overall cost of the network. If the cost of the network is more expensive than deploying traditional sensors, the sensor network is not cost justified.

- *Hardware constraints*: A sensor node comprising of many hardware components should be smaller than a cubic centimeter, consume extremely low power, operate unattended, operate in high volumetric densities, have low production cost and adapt to the environment.

- *Transmission media*: In a multi-hop sensor network, communicating nodes are linked by a wireless medium. To enable global operation of these networks, the chosen transmission medium must be available worldwide.

- *Power consumption*: The wireless sensor node can be equipped with a limited power source. Sensor node lifetime therefore shows a strong dependence on battery lifetime. In a multi-hop ad hoc sensor network, each node plays the dual role of data originator and data router. The malfunctioning of a few nodes can cause significant topological changes and might require rerouting of packets and reorganization of the network.

Programs developed to execute in a wireless sensor node according to above criteria also must take into account node's hardware restrictions [6]. A wireless sensor node in WSNs comprises one or more sensor elements, battery, memory, processor, transceiver and receivers. Some of them may also have the components such as a location finding system, power generator and a mobilizer.

2. Main Features and Benefits of WSNs

WSNs generally have several important challenges that need to be overcome [10,11];

- Sensor nodes use their limited supply of *energy*. Therefore, the short, multihop transmission schemes is preferred.

- *Computation* to which some algorithms are overcome by collaborative effort is the other obstacle.

- *Communication* is another one which the bandwidth of the wireless links connecting sensor nodes is often limited.

- *Scalability* is the other important feature of a WSN. The network must be *scalable* to thousands of sensor nodes to support the sensor density needed to cover an area.

- *Power aware* protocols that perform signal processing and communications with low energy consumption are designed for WSNs.

- An IP based addressing is not useful for WSNs since sensor networks require local unique addresses instead of global unique addresses.

- The WSN should be able to cope with the resultant distribution and form connections between the nodes.

- The WSN should be adaptive in nature to changing connectivity and node failure.

- The configuration should be done automatically and repeatedly.

Furthermore, the nodes in WSNs are small, low-cost nodes and have low sensing and transmitting range. They are densely deployed very close to the phenomenon in sensor field to reduce energy consumption while sensing and transmitting the data.

In addition, the following definition generally is used for WSNs [6];

- A WSN is said to be *homogeneous* when all nodes have the same hardware; otherwise it is *heterogeneous*.

- The nodes are *autonomous* when they are able to execute self-configuration tasks without human intervention.

- A WSN is *static* when nodes are stationary and *dynamic* otherwise. Note that the topology may be *dynamic* even when nodes are stationary since new ones can be added to the network or existing nodes become unavailable.

- A WSN is *symmetric* when each transceiver has the same transmission range and *asymmetric* otherwise.

- A WSN is *continuous* when sensor nodes collect data and send them to an observer continuously along time and *on demand* when they answer to observer's queries.

- A WSN is *reactive* when sensor nodes send data referring to events occurring in the environment and *programmed* when nodes collect data according to conditions defined by the application.

- A WSN is *hybrid* when it has at least two of the above characteristics concerning dissemination of information.

A detailed description of the various components and functionalities provided by a typical WSN is given in [12]. Moreover, the advantages of monitoring an area using a distributed sensing network versus a large stand-alone sensor system are as follows [13]:

- Allow greater fault tolerance through a high level of redundancy,

- Provide coverage of a very large total space through the union of many small sensors' coverage areas.

- Incrementally extend coverage area and density,

- Ensure greater sensing quality by combining information from different spatial perspectives,

- Improve sensing performance with multiple sensor types,
- Characterize continuous phenomena and localize discrete phenomena,
- Using short-range sensor technologies inherently capable of sensing phenomena only at close distances,
- Overcome ambient environmental effects by having sensors in close proximity to the object of interest.

B. ROUTING TECHNIQUES IN WIRELESS SENSOR NETWORKS

The main goal in conventional wireless networks is providing high quality of service and high bandwidth efficiency. In contrast, in a sensor network conservation of energy is considered to be important than the performance of the network [9]. Therefore, the current routing protocols designed for traditional networks cannot be used directly in a sensor network due to the following reasons [5,14,];

- Sensor networks are data centric, that it unlike traditional networks where data is requested from a specific node, data is requested based on certain attributes.
- The requirements of the network change with the application and so, it is application-specific.
- Adjacent nodes may have similar data. So rather than sending data separately from each node to the requesting node, similar data is aggregated and then sent. Data aggregation process is useful paradigm for wireless routing in WSNs. Minimizing the number of transmission, eliminating redundant data packets, and combining the data coming from different sources are some of the possible data aggregation processes to increase lifetime of the network.
- Each node is given a unique id which is used for routing in traditional networks. This cannot be effectively used in sensor networks because they are data

centric. Also, the large number of nodes in the network implies large ids which might be substantially larger than the actual data being transmitted.

1. Design Criteria for Routing in WSNs

The routing protocols must select the best path to minimize the total power needed to route packets on the network and maximize the lifetime of all nodes. That is, these protocols should be scalable to obtain different energy and quality operating points as the relative importance of different resources as requirements might change over the system lifetime [9]. Also, wireless sensor networks need protocols which are data centric, capable of effective data aggregation, distribute energy dissipation evenly, efficiently use their limited energy to increase the longevity of the network and avoid any single point bottleneck (except the sink) [9]. As the energy gets depleted, the network may be required to reduce the quality of the results in order to reduce the energy dissipation in the nodes and hence lengthen the total system lifetime.

Design criteria for the data dissemination are the following [15,16];

- *Immediate deployment*: The protocol should be designed not to require a long-term startup (e.g., network topology construction) after sensor node placement, to get ready for the actual sensor data dissemination.

- *Adaptability*: The protocol should be scalable to both the number of data sources and the data sink populations, and allow the diversity of user requests in terms of desired update rates and service durations.

- *Fast response to data requests*: It is desirable that users do not experience a substantial amount of delay after they request sensor data updates.

- *Energy efficiency*: Given data update demands, the protocol should be able to satisfy them with lower energy dissipation and ultimately extend the network lifetime.

- *Reliable delivery*: Reliable data delivery to the sinks must be ensured in spite of unreliable individual sensors.
- *Limited resources*: Sensors have limited memory size and processing power. Their activity in data forwarding should thus be minimized in order to prolong their lifetime.
- *Error-prone wireless medium*: Sensor nodes communicate through the wireless medium that is more error prone than the wired medium and has significantly lower bandwidth.
- *No globally unique id*: Due to the large number of participating nodes, each node cannot be assigned a globally unique id.

2. Routing Protocols

The presence of a routing architecture connects all the sensors in a transparent manner [17]. The routing infrastructure is established as an interconnection between the nodes and the sink. Routing protocols can in general be divided three categories as shown in Figure 3. Note that some sources also use two categories for routing as hierarchical and flat routing protocols [10].

Routing Protocols

Data Centric	Hierarchical	Location Base
Spin	LEACH	MECN
Directed Diffusion	PEGASIS	SMECN
Classic Flooding	TEEN	GAF
Gossiping	SEAD	GEAR
Ideal dissemination	TTDD	
Rumor Routing	SAFE	
Minimum Cost Forwarding Alg.	Multiple Chain Sheme	
SAR		
Energy Aware Routing		

Figure 3. The routing protocol in WSNs.

Our aspect to all these routing protocols are from the point of general features and challenges. We investigate if they are suitable for many-to-many communication protocol in WSANs and they fit the requirements of WSANs or not.

a. Data Centric Routing

In this routing, the sink sends out a query/interest for data, the sensor nodes which have the appropriate data then respond with data. As shown in Figure 4, address-centric protocol(AC) and data-centric protocol (DC) differ in the manner the data is sent from the sources to the sink; if each source independently sends data along the shortest path to the sink based on the route that the queries took(end-to-end routing), it is address-centric protocol. Otherwise, if the sources send data to the sink, but routing nodes enroute look at the content of the data and perform some form of aggregation/consolidation function on the data originating at multiple sources, then it is data-centric protocol [5].

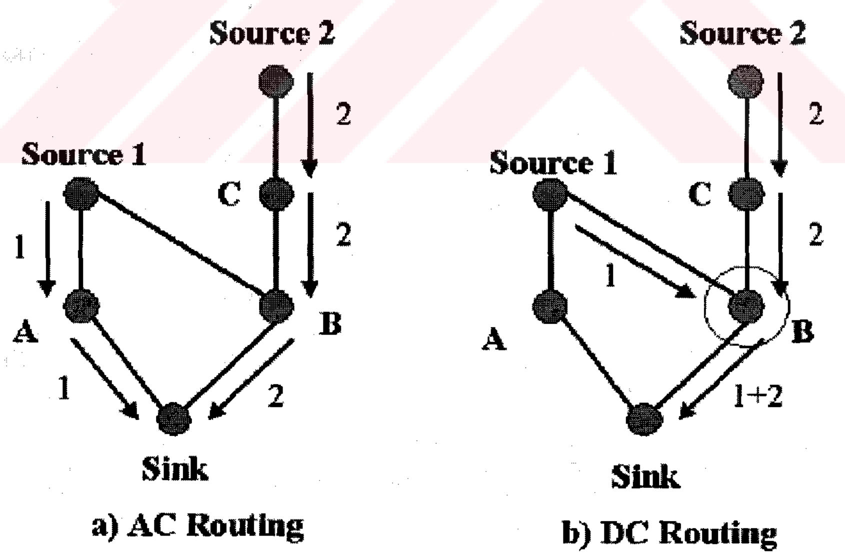
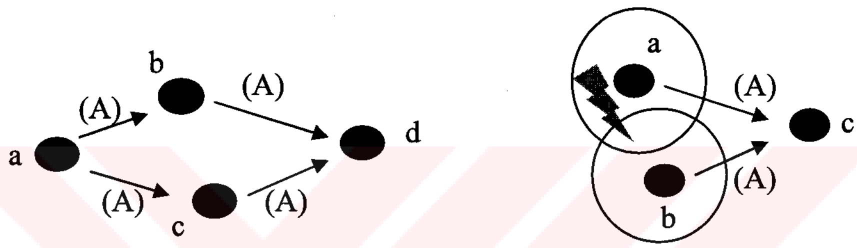


Figure 4. Illustration of AC versus DC Routing [5].

(1) Sensor Protocols for Information via Negotiation (SPIN) [18]

In this protocol, the efficient dissemination of individual sensor observations to all the sensors in a network is obtained by treating all sensors as potential sink nodes. Thus, it overcomes the problems of implosion and overlap. These problems are shown in Figure 5. SPIN nodes negotiate with each other before transmitting data. Negotiation helps ensure that only useful information will be transferred. To negotiate successfully, however, nodes must be able to describe or name the data they observe. The descriptors used in SPIN negotiations as *meta-data*.



(a). The implosion problem.
Node d receives two copies of *packet A* from different neighbours.

(b) The overlap problem.
Node a and *b* cover an overlapping geographic region. When these sensor nodes sense a phenomenon and send the data to the *node c*, two copies of *packet A* are received by node c.

Figure 5. The Implosion and Overlap problems.

SPIN is an application-level approach to network communication. SPIN is implemented as middleware application libraries with a well defined API. These libraries will implement the basic SPIN message types, message handling routines, and resource management functions.

SPIN applications are resource-aware and resource adaptive. They can poll their system resources to find out how much energy is available to them. They can also calculate the cost, in terms of energy, of performing computations and sending and receiving data over the network.

SPIN uses three kinds of messages to communicate, which are *ADV* (advertisement), *REQ* (request) and *DATA*.

When a node obtains a new data, it advertises the data to its neighbors by sending an ADV packet as shown in Figure 6.a. A node received an ADV packet checks if it needs this data or not. If it does, the node sends a REQ packet to the originator of ADV packet as shown in Figure 6.b. Then, the source node sends the data to the nodes which send a REQ packet as shown in Figure 6.c. At last, the algorithm is processed as the same way by all the nodes that receive the data packet as shown in Figure 6.d.

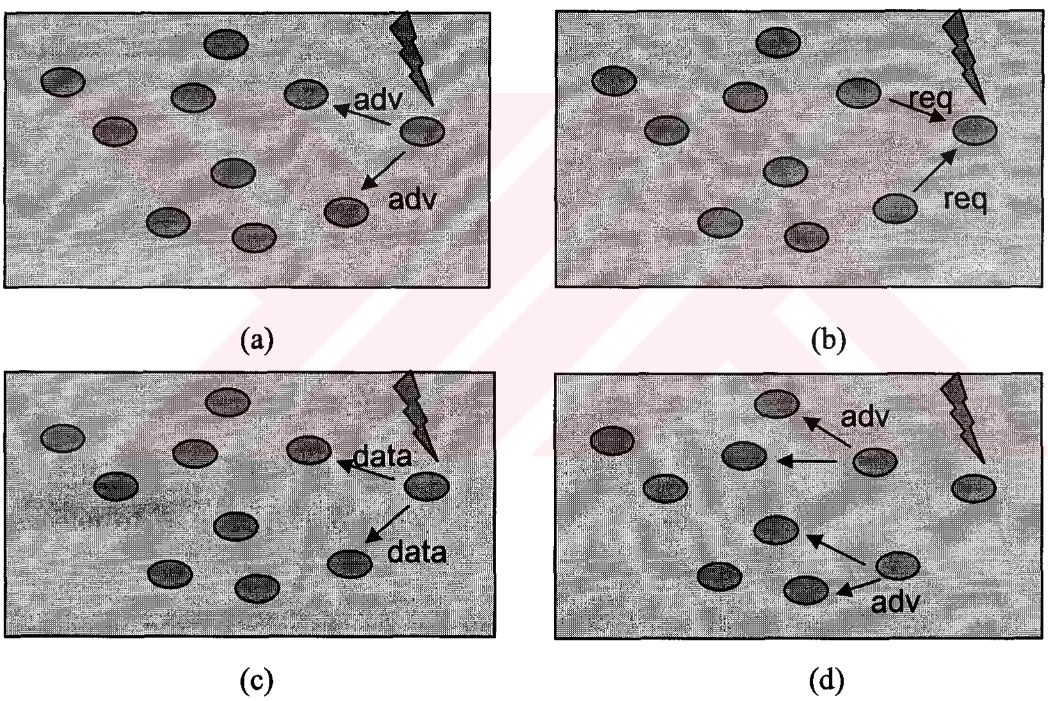
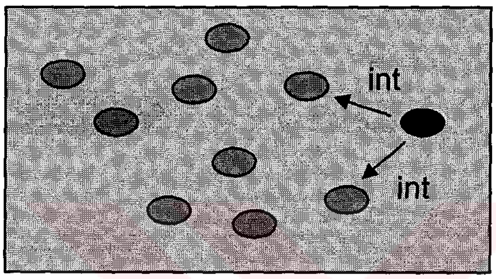


Figure 6. SPIN protocol.

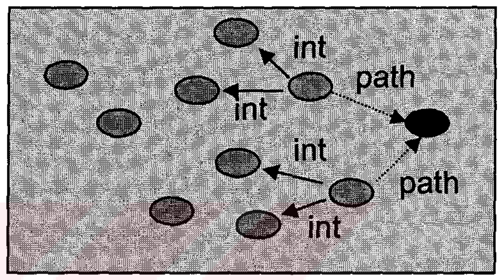
The main challenges of SPIN for WSANs that there are great number of control packets. Many actuators that may require the same sensed data in WSANs are the end point for the conveyed data. When they are willing to obtain the sensed data, a lot of ADV and REQ packets are sent between the sensor nodes. This control packet traffic reaches awful high.

Furthermore, the implosion and overlap problems increase very fast in addition to high number of packet traffic. So, the system designer have to take the additional precautions for it.

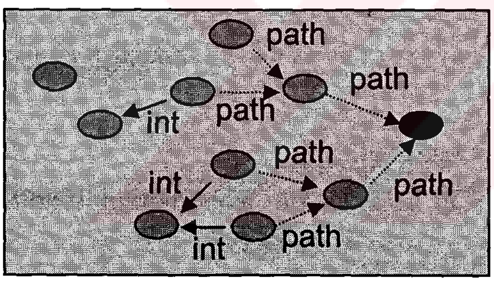
(2) Directed Diffusion [19]



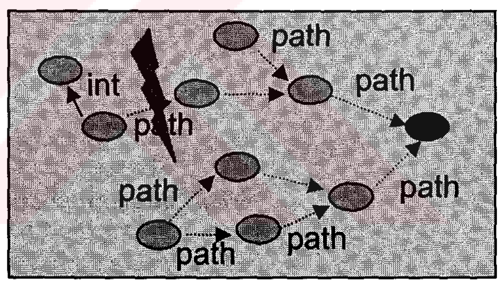
(a) Interest propagation by sink.



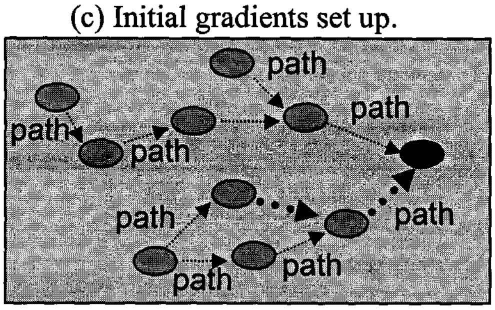
(b) Initial gradients set up.



(c) Initial gradients set up.



(d) Event detection and being reinforced one path by sink.



(e) Data delivery along reinforced path.

Figure 7. Directed Diffusion protocol.

This protocol is one of the most well-known protocols for WSNs. It is data-centric in that all communication is for the data which is attribute based naming used by the sensor nodes.

The sink queries to the network by transmitting an *interest* which is periodically refreshed by the sink as shown in Figure 7. This initial interest is a low event-rate notification, i.e. higher interval. A node requests data by sending this interest for named data. While interest propagates through the network, the relaying nodes establish the *gradients* of the data towards the sink. Intermediate nodes can cache, or transform data and may direct interests based on previously cached data. When the sink receives a respond of its query, it reinforces one particular neighbor in order to draw down higher quality (higher data rate) events. When source receives again the interest which is high date rate, i.e. smaller interval, it begins to send the data packets from this path for smaller interval. Thus, reinforced path is established between sink and the source node.

An important feature of directed diffusion is that interest and data propagation and aggregation are determined by *localized interactions* (message exchanges between neighbors or nodes within some vicinity).

The challenges of directed diffusion for WSANs are as follows [20];

- A source of sensory data periodically broadcasts, at a low rate, events describing detections of the external phenomenon that is being sensed.

- Upon receiving multiple copies of these events, the sink sends a *reinforcement* message to one of its neighbors indicating that it prefers to receive notifications of detection events at a higher frequency from this neighbor.

- That reinforcement message is propagated to the source, hop-by-hop by nodes. Each node makes an independent, local decision about which of its neighbors it chooses to forward the reinforcement.

- When a node on the reinforced path fails, the sink detects an absence of detection events and reinitiates reinforcement. For this to work, the sink must continue to periodically send reinforcement messages.

When many actuators in a WSAN require the same data, the control packet transmission in directed diffusion reaches high level according to above criteria. In addition to that, periodically broadcasting an interest or data can not be necessary for WSAN architecture as mentioned in section II.C.

(3) Classic Flooding [3,18]

One of the oldest techniques in WSNs is flooding. In this protocol, each node received packets repeats it by broadcasting. When the maximum number of hops for the packet is reached or the destination of the packet is the node itself, broadcasting process is finished. Flooding does not require costly topology maintenance and complex route discovery algorithm.

The amount of time it takes a group of nodes to receive some data and forward that data on to their neighbors is called round. Flooding converges in O(d) rounds, where d is the diameter of the network, because it takes at most d round for a piece of data to travel from one end of the network to the other.

Since flooding does not solve either the implosion or the overlap problem and the broadcast process is its key feature, it is not suitable for WSANs.

(4) Gossiping [21]

This protocol is an alternative to the classical flooding approach. The nodes only send the data packets to one randomly selected neighbor instead of broadcasting to all neighbors. Thus, gossiping protocol avoids implosion problem because it only makes one copy of each message at any node. If a gossiping node receives data from a given neighbor, it can forward data back to that neighbor if it randomly selects that neighbor. Whenever data travels to a node with high degree in a classic flooding network, more copies of the data start floating around the network. At some point, however, these copies may end up imploding. But gossiping has still the overlap problem [18].

In addition, since the source sends to only one of its neighbors, and that neighbor sends to only one of its neighbors, the fastest rate at which gossiping distributes data is 1 node/round. Thus, if there are c data sources in the network, gossiping's fastest possible distribution rate is c nodes/round.

As the consequences of all these challenges, we can say that this protocol can not be used in WSANs where the communication between the actuators and the sensor nodes must be done by a fast and efficient way.

(5) Ideal Dissemination [13,18]

Every node sends observed data along a shortest-path route and every node receives each piece of distinct data only once. This algorithm is called as ideal dissemination because observed data arrive at each node in the shortest possible amount of time. No energy is ever wasted transmitting and receiving useless data. Current networking solutions offer several possible approaches for dissemination using shortest-paths. One such approach is *network-level multicast*, such as IP multicast. In this approach, the nodes in the network build and maintain distributed source-specific shortest-path trees and themselves act as multicast routers. To disseminate a new piece of data to all the other nodes in the network, a source would send the data to the network multicast group, thus ensuring that the data would reach all of the participants along shortest-path routes. In order to handle losses, the dissemination protocol would be modified to use reliable multicast. Unfortunately, multicast and particularly reliable multicast both rely upon complicated protocol machinery, much of which may be unnecessary for the solving the specific problem of data dissemination in a sensor network. We review the detail multicast communication paradigm in section II.B.2.d.

(6) Rumor Routing [22]

The idea of this protocol lies behind fact that two straight lines in a plane are likely to intersect. This protocol that is a logical compromise between flooding queries and flooding event notifications, creates paths leading to each event; whereas event flooding creates a network-wide gradient field. In this way, when a query is generated it can be sent on a random walk until it finds the event path; instead of flooding it throughout the network as shown in Figure 8. As soon as the query discovers the event path, it can be routed directly to the event. If the path cannot be found, the application can try resubmitting the query, or as a last resort, flooding it.

The algorithm employs a set of long-lived agents that create paths (in the form of state in nodes) directed towards the events they encounter as shown in Figure 9. Whenever an agent crosses a path leading to an event it has not yet seen, it adapts its behavior and thenceforth creates path state that leads to both (or multiple) events. The agents also optimize the paths in the network if they find shorter ones. Moreover, when an agent finds a node whose route to an event is more costly than its own, it will update the node's routing table to the more efficient path.

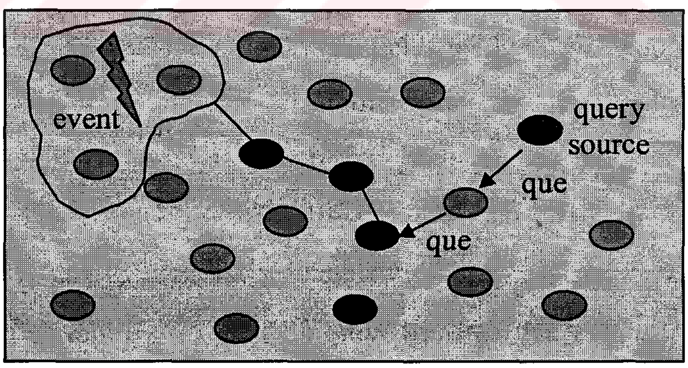


Figure 8. Finding path to event in Rumor Routing

Query is originated from the query source and searches for a path to the event. As soon as it finds a node on the path, it's routed directly to the event.

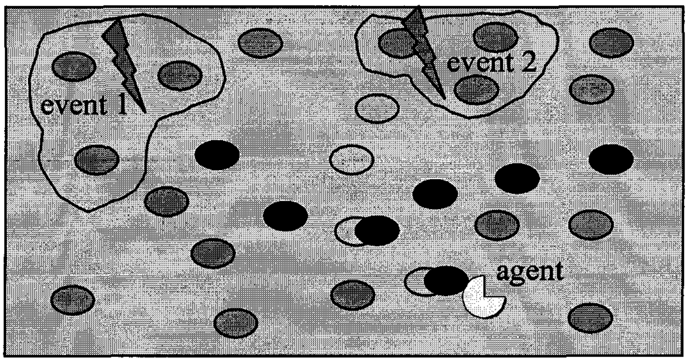


Figure 9. Creating path by an agent in Rumor Routing

When agent prorogating the path to Event 2 comes across a path to Event 1, it begins to propagate the aggregate path to both.

The queries in rumor routing are delivered to events in the network. Rumor Routing is tunable, and allows for tradeoffs between setup overhead and delivery reliability. It's intended for contexts in which geographic routing criteria are not applicable because a coordinate system is not available or the phenomenon of interest is not geographically correlated.

The challenges of rumor routing in aspect of WSANs;

- The events may not be continues in WSANs. An event occurs frequently with specific interval in a certain region. Therefore, it may not be possible that the queries is delivered to the events.
- Events may not be in large region in WSANs. Therefore, the queries in rumor routing can not find any path to these kind of events which have little range.
- Since there are a lot of actuators in WSANs, it is possible that many queries are in searching process for an event in initial state of network. So, the packet transmission can be very high.

(7) Minimum Cost Forwarding Algorithm for Large Sensor Networks [13,23]

This protocol exploits the fact that data flow in sensor networks is in a single direction and is always towards the fixed base station. Their method neither requires sensor nodes to have unique identity number nor maintain routing tables to forward the messages. Each node maintains the least cost estimate from itself to the base station. Each message to be forwarded is broadcasted by the node. On receiving a message the node checks if it is on the least cost path between the source sensor node and the base station. If so it would forward the message by broadcasting.

In principle, the concept behind minimum cost forwarding is similar to the natural gravity field that drives waterfalls from top of mountain to the ground. At each point water flows from a high post to a low post along the shortest path. For this algorithm to work each node needs to have the least cost estimate from itself to the base station. The base station broadcasts an advertisement message with the cost set to zero. Every node initially has the estimate set to infinity. On receiving the advertisement message if the estimate in the message plus the cost of link on which it is received, is less than the current estimate, current estimate and the estimate in the advertisement message is updated as the estimate in the message plus the cost of the link on which it is received. If the received advertisement message is updated with a new cost estimate it is forwarded else its purged. As a result of forwarding advertisement message immediately after updating, it is noticed in the simulations that some nodes will get multiple updates and do multiple forwards as lesser cost estimates flow in. It was also reported by the authors that nodes far away from the base station get more updates than those close to the base station. To avoid this instability during the setup phase of the algorithm, a back-off algorithm is proposed. According to this back off algorithm on updating the current cost estimate, the advertisement message is not forwarded until $A * Cnode$ units of time from the time of update. *A* is a constant determined through simulations and *Cnode* is the cost of link on which the advertisement message was received.

In this protocol, the message can go along multiple paths and is not subject to node failures along a single. Moreover, node failures after the cost field setup phase can make the cost field not optimal. This issue can be addressed by refreshing the cost field, either in a time-driven (i.e., periodic) or event-driven.

(8) Sequential Assignment Routing (SAR) [24]

The Self-Organizing Medium Access Control for Sensor Networks (SMACS) is an infrastructure-building protocol that forms a flat topology (as opposed to a cluster hierarchy) for sensor networks. SMACS is a distributed protocol which enables a collection of nodes to discover their neighbors and establish transmission/reception schedules for communicating with them without the need for any local or global master nodes.

In order to achieve this ease of formation, the neighbor discovery and channel assignment phases in the SMACS protocol are combined. In SMACS, a channel or TDMA slots, to links between neighboring nodes is assigned to a link immediately after the link's existence is discovered. By this way links begin to form concurrently throughout the network. By the time all nodes hear all their neighbors, they will have formed a connected network. In a connected network, there exists at least one multihop path between any two distinct nodes.

Since only partial information about radio connectivity in the vicinity of a node is used to assign time intervals to links, there is a potential for time collisions with slots assigned to adjacent links whose existence is not known at the time of channel assignment. To reduce the likelihood of collisions, it is required that each link to operate on a different frequency. This frequency band is chosen at random from a large pool of possible choices when the links are formed.

Briefly, the SAR algorithm creates multiple trees where the root of each tree is a one-hop neighbor from the sink [3]. Each tree grows outward from the sink while avoiding nodes with very low QoS (i.e., low throughput/high delay) and energy reserves. At the end of this process each node will belong to multiple paths and each sensor can control which one-hop neighbor of the sink will be used for relaying the message.

There are two parameters associated with each path (i.e., a tree) back to the sink [3]:

- *Energy resources*: The energy resources are estimated by the number of packets the sensor node can send if the sensor node has exclusive use of the path.

- *Additive QoS metric*: A high additive QoS metric means low QoS.

The SAR algorithm selects the path based on the energy resources and additive QoS metric of each path, and the packet's priority level. The SAR algorithm attempts to minimize the average weighted QoS metric throughout the lifetime of the network. A periodic recomputation of paths is triggered by the sink to account for any changes in topology. Failure recovery is done through a handshaking procedure between neighbors and local path restoration is used.

(9) Energy Aware Routing for Low Energy Ad Hoc Sensor Networks [25]

The potential problem in current protocols is that they find the lowest energy route and use that for every communication. However, that is not the best thing to do for network lifetime. Using a low energy path frequently leads to energy depletion of the nodes along that path and in the worst case may lead to network partition. Therefore, this protocol proposes a more useful metric for routing protocol performance; *network survivability*. This means that the protocol should ensure that connectivity in a network is maintained for as long as possible, and that the energy health of the entire network should be of the same order.

The basic idea of energy aware routing is that to increase the survivability of networks, it may be necessary to use sub-optimal paths occasionally. This ensures that the optimal path does not get depleted and the network degrades gracefully as a whole rather than getting partitioned. To achieve this, multiple paths are found between source and destinations, and each path is assigned a probability of being chosen, depending on the energy metric. Every time data is to be sent from the source to destination, one of the paths is randomly chosen depending on the probabilities. This means that none of the paths is used all the time, preventing energy depletion. Also different paths are tried continuously, improving tolerance to nodes moving around the network. Energy aware routing is also a reactive routing protocol. It is a destination-initiated protocol where the consumer of data initiates the route request and maintains the route subsequently.

The protocol has three phases:

- Setup phase or interest propagation: Localized flooding occurs to find all the routes from source to destination and their energy costs. This is when routing (interest) tables are built up.

- Data Communication phase or data propagation: Data is sent from source to destination, using the information from the earlier phase. This is when paths are chosen probabilistically according to the energy costs that were calculated earlier.

- Route maintenance: Route maintenance is minimal. Localized flooding is performed infrequently from destination to source to keep all the paths alive.

b. Hierarchical Routing Protocols

This is another methodology for routing in WSNs. The basic idea behind this approach is to divide the sensor network into number of overlapping clusters. A change in the network topology corresponds to a change in the cluster membership. The performance of the proposed routing approach (reconvergence time, and topology update overhead) will then be determined by the average cluster size in the network [26].

Besides that, instead of discovering a path towards the sink within very large number of sensor nodes, grouping the sensor nodes and establishing a path towards the sink reduces the many challenges in WSNs such as scalability.

(1) Low Energy Adaptive Clustering Hierarchy (LEACH) [2,9,27]

LEACH is a clustering-based protocol that utilizes randomized rotation of local cluster base stations to evenly distribute the energy load among the sensors in the network. It uses localized coordination to enable scalability and robustness for dynamic networks, and incorporates data fusion into the routing protocol to reduce the amount of information that must be transmitted to the base station[9]. The purpose of LEACH is to randomly select sensor nodes as cluster-heads, so the high energy dissipation in communicating with the base station is spread to all the sensor nodes in the sensor network.

The operation of LEACH is separated into two phases, the setup phase and the steady state phase. In the setup phase the clusters are organized and cluster-heads are chosen. Then, in the steady state phase, the data transfer occurs to the base station. The duration of the steady state phase is longer than the duration of the setup phase in order to minimize overhead. During the setup phase, a sensor node chooses a random number between 0 and 1. If this random number is less than a threshold T (n), the node becomes a cluster-head for the current round. The threshold T (n) is calculated from the equation below;

$$T(n) = \begin{cases} \dfrac{P}{1-P*[r \bmod (1/P)]} & \text{if } n \in G \\ 0 & \text{otherwise} \end{cases}$$

Where P is the desired percentage to become a cluster-head, r is the current round, and G is the set of nodes that have not being selected as a cluster-head in the last 1/P rounds. Once the cluster-head has been elected, these nodes broadcast an advertisement message to the rest of the nodes in the network that they are the new cluster-heads. All the non-clusterhead nodes after receiving this advertisement, decide on the cluster to which they want to belong to. This decision is based on the signal strength of the advertisement. The noncluster-head nodes inform the appropriate cluster-heads that they will be a member of the cluster. After receiving all the messages from the nodes that would like to be included in the cluster and based on the number of nodes in the cluster, the cluster-head node creates a TDMA schedule and assigns each node a time slot when it can transmit. This schedule is broadcast to all the nodes in the cluster. During the steady state phase, the sensor nodes can begin sensing and transmitting data to the cluster-heads. The cluster-head node after receiving all the data aggregates them before sending to the base station. After a certain time which is determined a priori, the network goes back into the setup phase again and enters another round of selecting cluster-heads. Each cluster communicates using different CDMA codes to reduce interference from nodes belonging to other clusters.

(2) Power-Efficient Gathering in Sensor Information Systems (PEGAGIS)[28,29]

PEGASIS is a linear-chain based scheme used for gathering data in WSNs. A linear chain connects all the sensor nodes within the network. Data is transmitted from one end of the chain to the other end. Each node attaches its own data to the received data to form a larger packet and sends it to the next node. When data finally reaches the end of the chain, the same procedure starts from that end back to the beginning node. To conserve energy, each node maintains the same size of the header. Thus all nodes transmit the same number of bits except for the end nodes, because the end nodes only transmit their own data. It is useful to distribute the energy usage among nodes to prevent nodes from being unwisely overused. Using this scheme, nodes die randomly. When a neighbor dies, a node simply skips the dead node and transmits data to the next living neighbor in the chain.

$$\begin{tikzcd} C_0 \arrow[r] & C_1 \arrow[r] & \boxed{C_2} \arrow[d] & C_3 \arrow[l] & C_4 \arrow[l] & C_5 \arrow[l] \\ & & \boxed{\text{Base station}} & & & \end{tikzcd}$$

Figure 10. Data gathering in PEGASIS.

PEGASIS which is claimed an improvement over LEACH is a near optimal chain-based protocol, in which each node communicates only with its closest neighbor and takes turns to transmit to the base station as shown in Figure 10. This reduces the power required to transmit data per round and hence will increase lifetime of the network by two times when compared to LEACH. The main motivation behind this protocol was to increase the lifetime of each sensor node, decrease the bandwidth consumed by using local collaboration among the nodes and tolerate node failures.

On the other hand, the delay cost from data sensing to data transmission to the base station is high in PEGASIS. For this reason it is not suitable in WSANs which have many actuators that can be named as base station in aspect of PEGASIS and require to take sensed data reports fast.

(3) Multiple Chain Scheme[29]

Before reviewing the multiple chain scheme, we review another protocol which balances the energy and delay cost to avoid delay problem of PEGASIS. It is binary combining scheme using CDMA as shown in Figure 11. It performs parallel communication in about log n units of delay (n is the number of nodes in the network) and spends a little more energy than PEGASIS. Chain-based binary approach divides each communication round into log n levels. Each node transmits data to a close neighbor in a given level. Only those nodes that receive data can rise to the next level. Finally, one node, LEADER, sends data to base station. By then one transmission round completes. Similar to PEGASIS, each intermediate node performs data aggregation. Each round has only one LEADER, and live node takes turn to be LEADER. The scheme distributes workload evenly throughout the network.

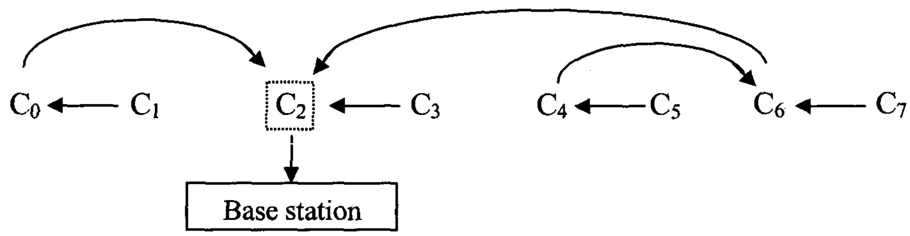


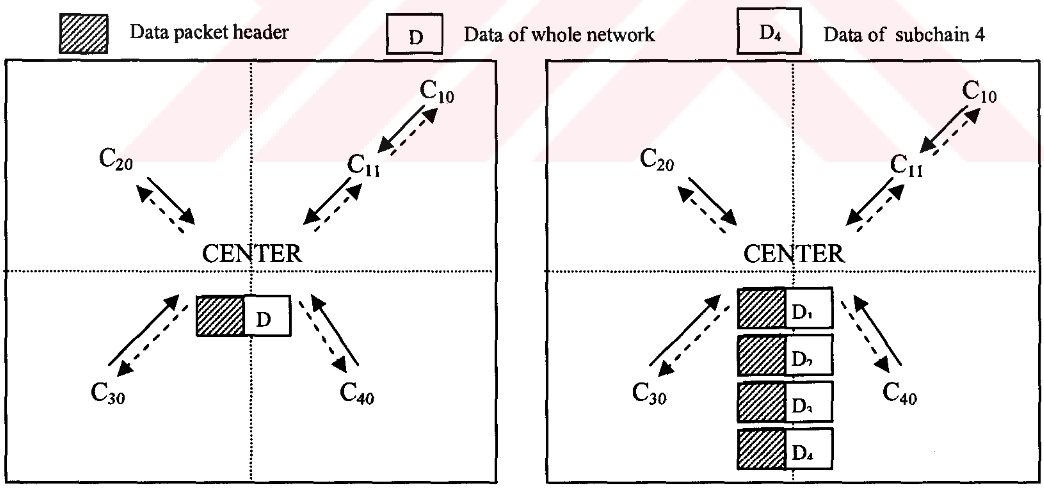
Figure 11. Data gathering in a Binary-Combining Scheme.

A multiple-chain scheme decreases the total transmission distance for all-to-all broadcasting. The main idea of this protocol is to divide the whole sensing area into four regions centered at the node that is closest to the center of the sensing area as shown in Figure 12. A linear subchain is constructed in each region. The maximum distance between two neighboring nodes in a subchain is smaller than that in a single linear chain generated within the whole area. For example, an 1 x 1 area is divided into four equal regions. The maximum neighboring distance in a subchain is $\frac{\sqrt{2}}{2}$1. However, it doubles to $\sqrt{2}$1 in a single chain. This scheme will minimize the total neighbor distance, especially for sparse node distribution. CENTER Node does become a bottleneck as it has to handle more receivings and transmissions than other nodes.

There exist two approaches to transfer data through the whole network, *collecting* and *relaying*.

- Collecting approach as shown in Figure 12.a.: Each subchain independently transfers and concatenates data from the end of chain to CENTER. After CENTER collects all data, it produces a new packet which contains information of the whole network. It then sends the packet to all four neighbors, which will relay the packet to the end of all subchains. However, every node except CENTER receives redundant data and, hence consumes more power than necessary.

- Relaying approach as shown in Figure 12.b.: The first part is the same as in Collecting. Each subchain independently transmits and accumulates data acket to CENTER. Then CENTER relays all received packets to all subchains. To avoid the redundancy, the neighbors of CENTER check each receiving packet. If a packet is the same as what a node just sent to CENTER, then it will discard the packet. If a packet is the first new packet the node receives, it will send this packet combined with its own sensing data along the chain. Each downstream node will perform the similar combination, create and send a larger packet to the next node, until transmission terminates at the end of subchain. If the packet does not belong to the proceeding two cases, the neighbor node will only relay this packet. No other data will be concatenated to the packet.



(a) Collecting Approach

(b) Relaying Approach

Figure 12. Multiple-Chain Scheme[29].

(4) Threshold Sensitive Energy Efficient Sensor Network Protocol (TEEN)[14]

Proactive networks continuously monitor the environment and thus have data to be sent at a constant rate. LEACH suites such sensor networks. On the other hand, in *reactive networks*, when the variable being monitored increases beyond a threshold, the sensing node needs to transmit the data. TEEN is proposed for this networks. TEEN is a cluster based routing protocol and is similar to LEACH except that sensor nodes do not have data to be sent at a fixed rate.

The main features of this protocol are as follows:

1. Time critical data reaches the user almost instantaneously. So, this scheme is eminently suited for time critical data sensing applications.

2. Message transmission consumes much more energy than data sensing. So, even though the nodes sense continuously, the energy consumption in this scheme can potentially be much less than in the proactive network, because data transmission is done less frequently.

3. The soft threshold can be varied, depending on the criticality of the sensed attribute and the target application.

4. A smaller value of the soft threshold gives a more accurate picture of the network, at the expense of increased energy consumption. Thus, the user can control the trade-off between energy efficiency and accuracy.

5. At every cluster change time, the attributes are broadcast afresh and so, the user can change them as required.

The main drawback of this scheme is that, if the thresholds are not reached, the nodes will never communicate, the user will not get any data from the network at all and will not come to know even if all the nodes die. Thus, this scheme is not well suited for applications where the user needs to get data on a regular basis. Another possible problem with this scheme is that a practical implementation would have to ensure that there are no collisions in the cluster.

(5) Minimum Energy Asynchronous Dissemination to Mobile Sinks in Wireless Sensor Networks (SEAD)[7]

A Scalable Energy-efficient Asynchronous Dissemination (SEAD) protocol considers the distance and the packet traffic rate among nodes to create near-optimal dissemination trees. The sinks can move without reporting their location to the tree while receiving data updates successfully. When sinks are mobile in sensor networks, communication consists of three main parts: building the dissemination tree (d-tree), dissemination data and maintaining linkage to mobile sinks. SEAD updates the d-tree dynamically. It is essentially composed of two main functions: one to add a node to the tree and one to remove a node from the tree. SEAD focuses on dissemination in which a source sends its data to multiple sinks. Dissemination has the source as the root of the d-tree as in multicast trees. If multiple sources are present in the sensor network, multiple dissemination trees will be constructed separately.

When a mobile sink requests to join to the network, it selects one of the neighbor node as an *access node*. This access node which represents the sink in d-tree stores the location of the sink and communicates with the sink. Thus, the data conveyed among the tree reaches the access node first and then the access node relays it to the sink. The d-tree is not updated unless the access node is changed. As the sink moves, the number of hop between the access node and the sink is computed frequently. The access node is not changed unless the number of hops exceeds a threshold value. The value of this threshold allows trade-offs to be made between path delay and energy spent on reconstructing the tree.

Source data is replicated at selected nodes between the source and sinks. This selected node is called a replica as a sensor node that stores a copy of the source data. The replica temporarily stores the latest data incoming from the source and asynchronously disseminates it to others along the tree. The replica needs only a very small amount of memory enough for a single data record. Replicas are members of the d-tree.

SEAD has the following basic assumption;

- Each sensor node is assumed to be aware of its own geographic location. The location estimation does not require Global Positioning System (GPS) at every node.

- After having been deployed, sensor nodes remain stationary at their initial locations.

- The sensor nodes are homogeneous and wireless channels are bidirectional. Each sensor node has a constrained battery energy.

- Sensor nodes communicate with sinks by delivering data across *multiple* hops. That is to say, sources and sinks are typically much further apart than a single radio radius.

In addition to above assumptions, SEAD does not use mobile sinks as intermediate members of the tree.

(6) A Two Tier Data Dissemination Model for Large-scale Wireless Sensor Networks (TTDD)[30]

A Two-Tier Data Dissemination (TTDD) approach is used to address the multiple, mobile sink problem. Instead of propagating query messages from each sink to all the sensors to set up data forwarding information, TTDD design uses a grid structure so that only sensors located at grid points need to acquire the forwarding information. Upon detection of a stimulus, instead of passively waiting for data queries from sinks, the approach taken by most of the existing work, the data source proactively builds a grid structure throughout the sensor field and sets up the forwarding information at the sensors closest to grid points (henceforth called dissemination nodes). With this grid structure in place, a query from a sink traverses two tiers to reach a source. The lower tier is within the local grid square of the sink's current location (henceforth called cells), and the higher tier is made of the dissemination nodes at grid points. The sink floods its query within a cell. When the nearest dissemination node for the requested data receives the query, it forwards the query to its upstream dissemination node toward the source, which in turns further forwards the query, until it reaches either the source or a dissemination node that is already receiving data from the source (e.g. upon requests from other sinks). This query forwarding process lays information of the path to the sink, to enable data from the source to traverse the same two tiers as the query but in the reverse order.

The basic assumption of TTDD;

- A vast field is covered by a large number of homogeneous sensor nodes which communicate with each other through short-range radios. Long distance data delivery is accomplished by forwarding data across multiple hops.

- Each sensor node is aware of its own location. However, mobile sinks may or may not know their own locations.

- Once a stimulus appears, the sensors surrounding it collectively process the signal and one of them becomes the source to generate data reports.

- Sinks (users) query the network to collect sensing data. There can be multiple sinks moving around in the sensor field and the number of sinks may vary over time.

In addition to the above assumption, the sensor nodes are aware of their missions and the mission of a sensor network changes only infrequently.

In this protocol, when a large number of nodes have potential to be data sources, it might be heavy load to construct grid networks per data source[31].

Therefore, It is not suitable for WSANs in which the actuators requires to receive new sensed data rapidly for a fast reaction.

(7) SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks [15]

This protocol is used for data dissemination in two-tiered networks comprised of stationary sensor nodes and mobile data users who request periodic sensor data updates .

Using SAFE (sinks accessing data from environments), individual sensor observations are disseminated to data sinks that explicitly present their interests by sending data requests. Each data sink is allowed to specify its own desired data update rate, and SAFE finds a subscription point through which the sink gets updated, trying to minimize the message transfers in the entire network. SAFE provides energy efficiency and fast response, without severe degradation as the number of data sources and data sinks increases.

In this protocol, there are two major phases: query transfer and dissemination path setup. In query transfer phases, the sensor node transfers the query to its neighbors via one-hop broadcast according to data management table and recent query. In dissemination path setup phases, all the intermediate nodes on the route overhear the pathSetup message and decide how to deliver the data message to the sink.

c. Location Based Routing

The location information of a sensor node is used in these kind of protocols to cope with existing challenges and constraints of sensor networks. Obtaining a scalable networks is also the aim of these protocols.

In these protocols, Global Positioning System (GPS) or location estimating algorithms are mostly used to find the location of sensor nodes. Therefore, cost price of GPS or initial cost of location based routing protocols is generally higher than the other protocols since they have extra devices or extra processes to determine the location of the node. For this reason, we don't choose to use any location based routing protocol in WSANs.

(1) MECN and SMECN[32,33,34]

MECN identifies a relay region for every node. The relay region consists of nodes in a surrounding area where transmitting through those nodes is more energy efficient than direct transmission. The relay region for node pair (i, r) is depicted in Figure 13, redrawn from [32]. The enclosure of a node i is then created by taking the union of all relay regions that node i can reach. The main idea of MECN is to find a sub-network, which will have less number of nodes and require less power for transmission between any two particular nodes. In this way, global minimum power paths are found without considering all the nodes in the network.

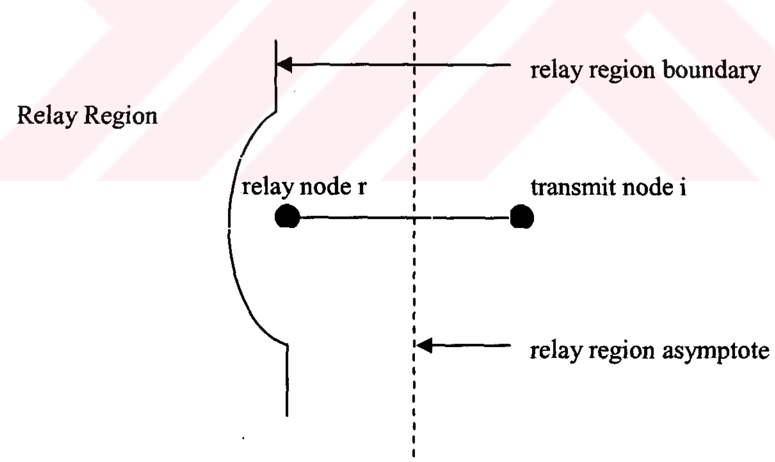


Figure 13. Relay region of the transmit-relay node pair (i,r) in MECN[32].

The small minimum energy communication network (SMECN) is an extension to MECN. In MECN, it is assumed that every node can transmit to every other node, which is not possible every time. In SMECN possible obstacles between any pair of nodes are considered. However, the network is still assumed to be fully connected as in the case of MECN. The sub-network constructed by SMECN for minimum energy relaying is provably smaller (in terms of number of edges) than the one constructed in MECN if broadcasts are able to reach to all nodes in a circular region around the broadcaster. As a result, SMECN uses less energy than MECN.

(2) Geographical Adaptive Fidelity (GAF)[35]

Each GAF node uses location information to associate itself with a "virtual grid", where all nodes in a particular grid square are equivalent with respect to forwarding packets. Nodes in the same grid then coordinate with each other to determine who will sleep and how long; this determination is moderated by application and system information. Nodes then periodically wake up and trade places to accomplish load balancing.

In GAF, nodes are in one of three states: sleeping, discovery, active. Initially nodes start out in the discovery state. When in state discovery, a node turns on its radio and exchanges discovery messages to find other nodes within the same grid. The grid id of a node is determined by using the node's location and grid size. A node periodically re-broadcast its discovery message at determined interval, when it is in active state. A node in discovery or active states can change state to sleeping when it can determine some other equivalent node will handle routing. A node in sleeping mode wakes up after the determined time.

(3) Geographical and Energy Aware Routing (GEAR)[36]

Disseminating information to a geographic region which is a very useful primitive in many location aware systems and especially sensor networks is used in this protocol.

In GEAR, the process of forwarding a packet to all the nodes in the target region consists of two phases:

1. Forwarding the packets towards the target region: GEAR uses a geographical and energy aware neighbor selection heuristic to route the packet towards the target region. There are two cases to consider:

(a) When a closer neighbor to the destination exists: GEAR picks a next-hop node among all neighbors that are closer to the destination.

(b) When all neighbors are further away: In this case, there is a hole. GEAR picks a next-hop node that minimizes some cost value of this neighbor.

2. Disseminating the packet within the region: Under most conditions, Recursive Geographic Forwarding algorithm is used to disseminate the packet within the region. However, under some low density conditions, recursive geographic forwarding sometimes does not terminate, routing uselessly around an empty target region before the packet's hop-count exceeds some bound. In these cases, restricted flooding is used.

The assumptions of this protocol are as follows;

- Each query packet has a *target region* specified in some way (for the description of the algorithm, a rectangular region specification is used).
- Each node knows its own location and remaining energy level, and its neighbors' locations and remaining energy levels through a simple neighbor hello protocol. Note that a node can obtain its location information at low cost from GPS or some localization system, which presumably is already available due to the needs of sensor net applications.
- The link is bi-directional, i.e., if a node hears from a neighbor N_i, then its transmission range can reach N_i. This is not an unreasonable choice as most MAC layer protocols, such as IEEE 802.11, assume symmetric links.

d. Multicast Routing in WSNs

When an application must send the same information to more than one destination, multicasting is often used, because it is much more advantageous than multiple unicasts in terms of the communication costs.

In conventional multicasting algorithms, a multicast group is considered as a collection of hosts which register to that group. It means that, if a host wants to receive a multicast message, it has to join a particular group first. When any hosts want to send a message to such a group, they simply multicast it to the address of that group. All the group members then receive the message.

An efficient implementation of multicasting permits much better use of the available bandwidth by transmitting at most one copy of the any data on each link in the network [37]. Therefore We use the multicast paradigm in our protocol, PAMR. In our protocol, the actuators and the sensor nodes consist of the multicast tree according to the interested tasks of the actuators in order to convey the sensed data efficiently as we will mention in Section III.

(1) A Self-Organizing Approach To Data Forwarding In Large Scale Sensor Networks[16]

In the absence of sink interests, sensors lie in the field inactive and periodically switch on to listen to the advertisement messages. This way, energy is not wasted unnecessarily. Once sinks develop an interest in a certain stimulus, they go through the advertisement process in which the multicast tree is constructed. In the absence of stimuli, this tree maintained by exchanging low-frequency hello messages between sensors on the tree. Once a stimulus arrives in the field, a new branch is formed that connects this stimulus to the tree, and report generation and data forwarding are triggered. At this stage, reports are used as hello messages to maintain the tree structure. If after some time a sink loses interest in the stimulus, it tears down its branch. If all sinks tear their branches, the branch towards the stimulus is torn down and all sensors along this branch become inactive again.

In this protocol there are four phase to build a multicast tree. These are; "broadcasting sink advertisements", "election of candidate merge points", "election of one merge point", and "branch formation towards the sinks".

(2) Spatiotemporal Multicast in Sensor Networks[38]

Spatiotemporal multicast, called mobicast, is facilitated the task which involves the monitoring of mobile phenomena. Mobicast is a novel spatiotemporal multicast protocol that distributes a message to nodes in a delivery zone that evolves over time in some predictable manner. Mobicast allows applications to specify their spatiotemporal constraints by requesting a mobile delivery zone, which in turn enables the application to build a continuously changing group configuration, according to their spatial and temporal locality. A key advantage of mobicast lies in its ability to provide reliable and just-intime message delivery to mobile delivery zones on top of a random network topology.

The service for maintaining a dynamic sensor group for tracking a mobile entity can be easily implemented on top of mobicast. When an interesting entity is discovered and a group is initiated, a group leader initiates a mobicast session to a delivery zone that moves according to the estimated velocity of the mobile entity. The mobicast message includes the location and time of the discovery of the entity. A node joins the group immediately upon reception of the message and leaves the group after the delivery zone moves away. Data aggregation services in a mobile entity-centric group can also be implemented on top of mobicast by invoking aggregation algorithms after receiving the mobicast message.

There are two major challenges of this protocol;

- Sensor nodes are likely to be deployed in an ad hoc fashion, e.g., by dispersing them from airplanes. The topology of this type of sensor network would thus be rather random. More specifically, this type of network can contain "holes". A mobicast session might be stopped prematurely because of a hole too big on its path.

- The mobicast delivery zone moves through the physical space. Two nodes close in physical space can be relatively far away in terms of network hops due to the presence of holes in the network. This presents a challenge to timely delivery of mobicast messages, i.e., a mobicast protocol needs to consider potential latency due to long underlying network paths, in order to achieve timeliness.

In WSANs, the actuators and the sensor nodes may be mobile. In this case, it seems that this protocol is useful when there is a few actuator in the network. On the other hand, since many actuators which move inside the sensor field request more data from many regions which sometimes are intersect, the number of packet transmission reaches very high value. Therefore, it is not useful in that case. In addition, while an actuator moves inside the southeast region at the sensor field, it may require any data from the northwest region at the sensor field. In this case, this protocol has a huge challenge for WSANs.

(3) Energy-Conserving Data Placement and Asyncronous Multicast in Wireless Sensor Networks [39]

In this protocol, observers who join the asynchronous multicast tree specify a period at which the requested data should be reported. Flurries of changes in the environment need not be individually reported if they occur at time-scales smaller than this period. Different observers may specify different period requirements for the same measurement. For example, an observer who is close to the measured activity may request a higher reporting rate than a distant observer.

This protocol achieves four main functions;

1. It determines the number of data caches for each focus locale where important events or activities are taking place,

2. It chooses the best location for each cache such that communication energy is minimized,

3. It maintains each cache consistent with its data source at the corresponding focus locale,

4. It feeds data to observers from the most suitable cache instead of the original sources.

The middleware caching service runs on every sensor node. Sensor network that is dense enough such that a data cache can be placed at any arbitrary physical location in the monitored region, offering new degrees of freedom to the data placement algorithm. In addition, it is assumed that the nodes know their location.

Each node on the multicast tree rooted at the sensor maintains a location pointer to its parent as well as a location pointer to each of its children. One can think of these pointers as an application-layer routing table. For each child, the node maintains the *maximum propagation rate*, which is the maximum of all requested update rates of all observers served by that child. A node never forwards updates to a child at a rate higher than the child's maximum propagation rate. This way, flurries of environmental updates that exceed some receivers' requested rates are not propagated unnecessarily to those receivers.

C. WIRELESS SENSOR AND ACTUATOR NETWORKS

Interfacing to the physical world involves exchanging energy between embedded nodes and their environments. This takes two forms: sensing and actuation. Whatever the sensed quantity (temperature, light intensity), the sensor converts a particular form of energy (heat, light) into information. Actuation lets a node convert information into action. An actuator moves part of itself, relocates spatially, or moves other items in the environment. Sensing and actuation are together the means of physical interaction between the nodes and the world around them[1].

Interfacing to the physical world is the main idea of WSANs which we review. In general, we meet the WSANs which have the architecture shown in Figure 14 in control engineering. But we interest in a WSAN which are composed of a large number of tiny sensor nodes and actuators. So, the start point of this thesis is to connect the actuators to the sensor nodes in this kind of network efficiently. Note that we use the term WSAN for this kind of network.

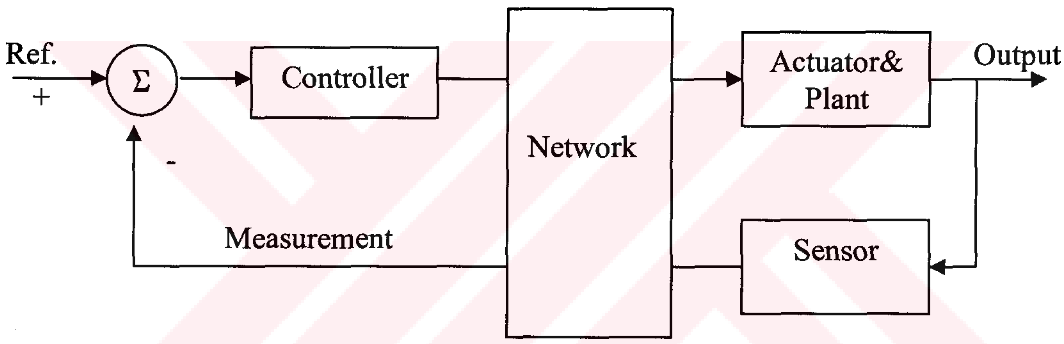


Figure 14. Basic Sensor and Actuator Network in control engineering[40].

WSAN architecture is motivated by applications such as fire fighting systems where an actuator can activate a single sprinkle at a corner of a room based on the data coming from multiple temperature and humidity sensors.

Another example of a WSAN application is the project to kill algae as a similar thought to SUBMAR project[40,46]. In this project, the network that consists of large number of nodes in a special region informs the actuators about the presence of algae at the bottom of sea. When the algae is sensed, the nearest actuator moves to this region and kills the algae by using poison in its tank.

1. Architecture and Main Features of WSANs

WSANs mainly compose of two kind of nodes; the *sensor nodes* sense the phenomenon from the environment by using their sensor(s). In this architecture, since a node may have more than one sensors, it senses many types of data which we describe later as a *task.* The sensed data is conveyed to the actuators by the sensor nodes. The sensor nodes are autonomous, and constrained in communication bandwidth, memory and capacity, and available power like as in all WSNs. Actually, the collecting information from the environment in WSANs is as the same in WSNs. But the major difference between WSNs and WSANs is the *actuator node.* In this architecture there is no sink node. The actuators collect the sensed data, then they obtain meaningful information from these sensed data. Thus, they control several devices attached to them based on these information. Actuators may coordinate with the other actuators in the field but decide independent from them.

All the constraints in WSNs is valid in WSANs. In addition, the following design criteria must be in a WSAN[1,40,41,42];

- The network should be robust,
- The network must be able to reconfigure,
- The network must be able to redistribute,
- The network must enable the cooperation of heterogeneous nodes,
- The network must be scalable,
- The sensors in an aware environment need to be able to calibrate automatically and must adapt to the environment as needed.
- The network must exhibit fast dynamic response to the environmental changes,

- Collaborative signal and information processing algorithm must cope with a great deal of uncertainty in physical hardware, sensor measurements and communication.

In general, the features of WSANs are as the same as in WSNs. In addition features of WSANs are as follows[1,42,43];

- Untethered network nodes and infrastuctureless mesh network topologies reduce deployment cost.
- The sensor nodes may be scattered or located to the sensor field,
- The actuators can be deployed inside the sensor field, and they may be collocated or dispersed,
- The actuators may have more power and memory capacity and computation capability than in sensor nodes,
- In WSANs, some actuators have the same kind of device(s) to actuate, some actuators actuates the same device(s) and some actuators may have completely different devices to actuate.
- The actuators may connect to the end users via internet or satellite system.
- A common problem in both sensing and actuation is uncertainty.

2. Design Criteria for Routing in WSANs

Distributed embedded systems such as WSANs require new programming models and software tools to support the rapid design and prototyping of sensing and control applications[43]. Therefore, system designers for WSANs must explicitly write code to[43];

- Maintain sensor connectivity in a neighborhood,
- Discover the best node for handoff,

- Invite neighbor nodes into the group/tree,
- Handle communication delays and failure.
- Use limited resources in a node efficiently,
- Prolong the lifetime of a network by taking some precaution ,
- Feed many actuator with the sensed data that a sensor node obtains.
- Fetch the sensed data from many sensor nodes to deliver an actuator.

a. Why Power Aware Routing

In WSANs, the routing protocol must be well-designed by the designers. The critical challenges are how the sensed data is conveyed to the actuators and which path is the most efficient path since the energy is a most critical resource for a node in WSANs like as in WSNs. If the energy in a node is not used efficiently, a lot of nodes die in the short time period. In this case, there will be holes in the network and the nodes don't communicate with each other efficiently. Therefore, the routing protocol must handle the limited power in a node to prolong to the lifetime of the network.

In our scheme, the nodes are aware of the minimum power level and the total power level along the path which is between a source node to the actuators. The sensor nodes choose the path which they use to convey any sensed data to the actuators according to these criteria. The detail of this process will be mentioned in section III.C.1.c.

b. Why Many to Many Routing

Manycast is a group communication scheme that enables communication with an arbitrary (client specified) number of group members. Anycast and multicast communication are special cases of manycast in which the target number of group members is one and infinity, respectively[12]. In our protocol, manycast generally is chosen by the source sensor node which senses a phenomenon. It communicates simultaneously with *k* of *m* number nodes according to the interest of the type of this data. By the way, how the *k* are chosen from the *m* nodes depends on our protocol of which the detail will be mentioned in section III.C.2.

In addition, Many to Many Invocation (M2MI) is useful for building a broad range of systems, including multiuser applications (conversations, groupware, multiplayer games); systems involving networked devices (printers, cameras); wireless sensor networks; and collaborative middleware systems[44]. Instead of routing messages from point to point to the particular destination devices, M2M Protocol broadcasts messages to all nearby devices, taking advantage of the wired or wireless network's inherent broadcast nature. Based on the message contents, the devices then decide whether and how to process the message.

M2MI offers these key advantages over existing systems[44]:

- M2MI-based systems do not require central servers; instead, applications run collectively on the proximal devices themselves.

- M2MI-based systems do not require network administration to assign addresses to devices, set up routing, and so on, since method invocations are broadcast to all nearby devices.

- Consequently, M2MI is well-suited for an ad hoc networking environment where central servers may not be available and devices may come and go unpredictably.

- M2MI-based systems do not need complicated ad hoc routing protocols that consume memory, processing, and network bandwidth resources.

- Consequently, M2MI is well-suited for small mobile devices with limited resources and battery life.

- M2MI simplifies system development in several ways. By using M2MI's high-level method call abstraction, developers avoid having to work with low-level network messages. Since M2MI does not need to discover target devices explicitly or set up individual connections, developers need not write the code to do all that.

- M2MI simplifies system deployment by eliminating the need for always-on application servers, lookup services, codebase servers, and so on; by eliminating the software that would otherwise have to be installed on all these servers; and by eliminating the need for network configuration.

Since an actuator may fetch the data from many sensor nodes and a sensor node may feed many actuators with the sensed data, M2M protocol is very suitable for WSANs. In our protocol, this scheme applies by broadcasting the data packets to the neighbor sensor nodes which interest in that type of data while conveying the data packets to the actuators. The detail of this process also will be mentioned in section III.C.2.

c. Why Data Centric Routing

The actuators in WSANs may require one type of data or many types of data to actuate their attached devices. For example, an actuator actuates the attached device base on only temperature value, another one requires temperature and humidity values at the same time. Therefore, the actuators in WSANs must inform their interests for the task (temperature, humidity, pressure…etc.). When a sensor node which has received these interests obtains the relevant data from its sensors, it sends this data to the relevant neighbor to convey to the actuator. For this reason, our protocol is data centric which is mentioned before.

In addition, data aggregation is the key concept in data centric routing. In general, there are two data aggregation methods[5]; optimal and suboptimal aggregation. The detail of these methods are mentioned and the performance of data-centric routing in WSNs is modeled and analyzed in [5]. Note that in our protocol, we prefer the *duplication suppression* function which is one of the simplest function for data aggregation which is mentioned in section III.C.4.

III. THE NEW POWER AWARE MANY TO MANY DATA CENTRIC ROUTING SCHEME FOR WIRELESS SENSOR AND ACTUATOR NETWORKS: PAMR

A. THE FEATURES OF PAMR

The major features of PAMR are as fallow:

- The PAMR is a practical and easily comprehensible routing protocol for WSANs,
- The PAMR does not require large computational memory capacity,
- The PAMR complies with various existing protocol,
- We have three design parameters in PAMR; *listening time* for realizing whether the uplink nodes convey the transmitted data packet or not, *threshold value* for critical power level in a node, *cache time* for saving a transmitted data packet for a very short time in a node.

The WSAN architecture that PAMR is designed for has the following features[31]:

- Several actuators, and a high number of sensor nodes that may be attached multiple sensors are densely deployed in a sensing field.
- The sensor nodes communicate with each other through short-range radios and they are constrained in communication bandwidth, memory capacity, and available power.
- An actuator evaluates the sensed data received from the sensor nodes in order to make a decision for actuating its attached devices.
- A sensor node may feed many actuators with the sensed data.
- An actuator may fetch the sensed data from many sensor nodes.

- Each actuator has a unique identifier.
- Both actuators and sensor nodes may be mobile.

B. THE CONTENTS OF MESSAGES

In PAMR protocol, 6 types of messages are used as shown in Figure 15. Note that (a-e) are control packets where (f) is a data packet.

a. Task registration message.

header	node_id	actuator_id	echelon	minPA	totalPA	task(s)

b. Task deregistration message.

header	node_id	actuator_id	task(s)

c. Update message.

header	node_id	actuator_id

d. Reroute message.

header	node_id	task

e. Deactivation message.

header	node_id

f. Data message.

header	uplink_node_id(s)	time	task(s)	data(s)

Figure 15. The message formats in PAMR.

An actuator broadcasts a *task registration message (TAREM)* to inform the network about its interest for the sensed data. When a node does not need a previously registered sensing task anymore, it deregisters itself from that task by broadcasting a *task deregistration message (TADREM)*. An *update message* is used to construct a new route to replace a failed route. A *reroute message* is used to be invalid the forwarded data message when the node is unsuccessful to rebuilt a new route. A *deactivation message* indicates that the source node will be unavailable soon due to energy depletion. *Data messages* convey the sensed data between sensor nodes and actuators.

In message header, the message type descriptor exists for all messages. In addition, the TAREMs have one more message descriptor which the sensor nodes use to determine if the message is new or not. The actuators initialize this second descriptor as 1. When one of the actuators requires to transmit a new TAREM, it multiplies this value with -1 and then broadcasts the message.

C. THE OPERATION OF A WSAN WITH PAMR

In our protocol, actuators register their interest for data that they need to the nodes in the sensor network by broadcasting a *registration message*. The *registration message* contains fields such as the number of hops from the actuator, i.e., echelon, the node that has the minimum power available along the route, and the total power available in the route [3], which are updated by every node that relays the message. While the *registration message* is being disseminated through the network, the sensor nodes relaying the *registration message* build up their registration table by inserting a registration record with the fields extracted from the registration message, which are the actuator identification, the sensing tasks registered by that actuator, the uplink node for the actuator, and the other metrics related to the power efficiency of the routes. Sensor nodes derive a routing table from the registration table where there is a single record for every unique uplink node and sensing task pairs. When a sensed data packet is received, that is forwarded to the uplink nodes looked up from the routing table. Hence a many-to-many multicast tree is obtained for each sensing task, i.e., the sensed data measured by a specific type of sensor. Actuators can also deregister from a sensing task by broadcasting a *deregistration message*. A node that receives a *deregistration message* updates its registration and routing tables accordingly. We also provide procedures that make our scheme adapt itself to the changes in the number, locations, and the sensing task interests of the actuators, as well as power available in the nodes.

1. Route Discovery

An actuator may run based on multiple types of sensed data such as temperature, humidity and proximity. Similarly several actuators may use the same type of sensed data. Therefore, every actuator must inform the sensor network about the type of sensed data for which it is interested in. This is achieved by broadcasting a TAREM, i.e. task registration process, as shown in Figure 16.

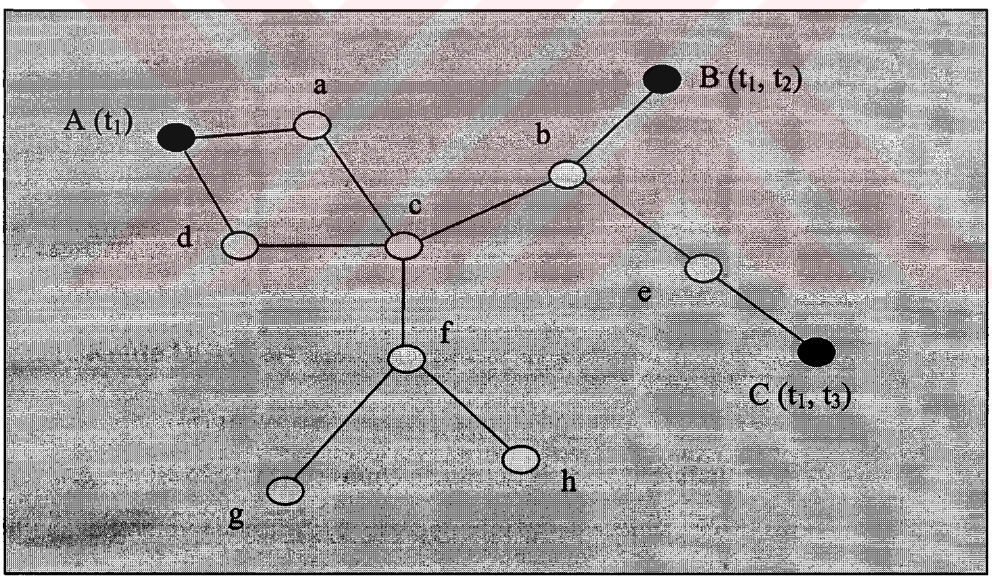


Figure 16. Task dissemination in PAMR.

The nodes maintain two tables: a *registration table* and a *routing table* as shown in Tables 1 and 2 respectively. The *registration table* of a node is the list of actuators that have registered at least one task to the network. The *routing table* is the list of neighboring nodes that the sensed data is forwarded to, i.e., uplink nodes.

When a new sensed data packet is received, this table is looked up for the uplink (i.e., next hop) nodes. The same type of sensed data may be relayed to multiple nodes. Therefore, there may be multiple records for the same task in a routing table. However, there is a single record for every unique uplink node and sensing task pair. The *routing table* is derived from the *registration table.*

Table 1. The registration table *of Node c in Figure 16.*

Actuator Id	Uplink Node Id	Echelon	minPA	totalPA	Task(s)
A	a	2	5	5	t_1
A	d	2	4	4	t_1
B	b	2	7	7	t_1,t_2
C	b	3	3	10	t_1,t_3

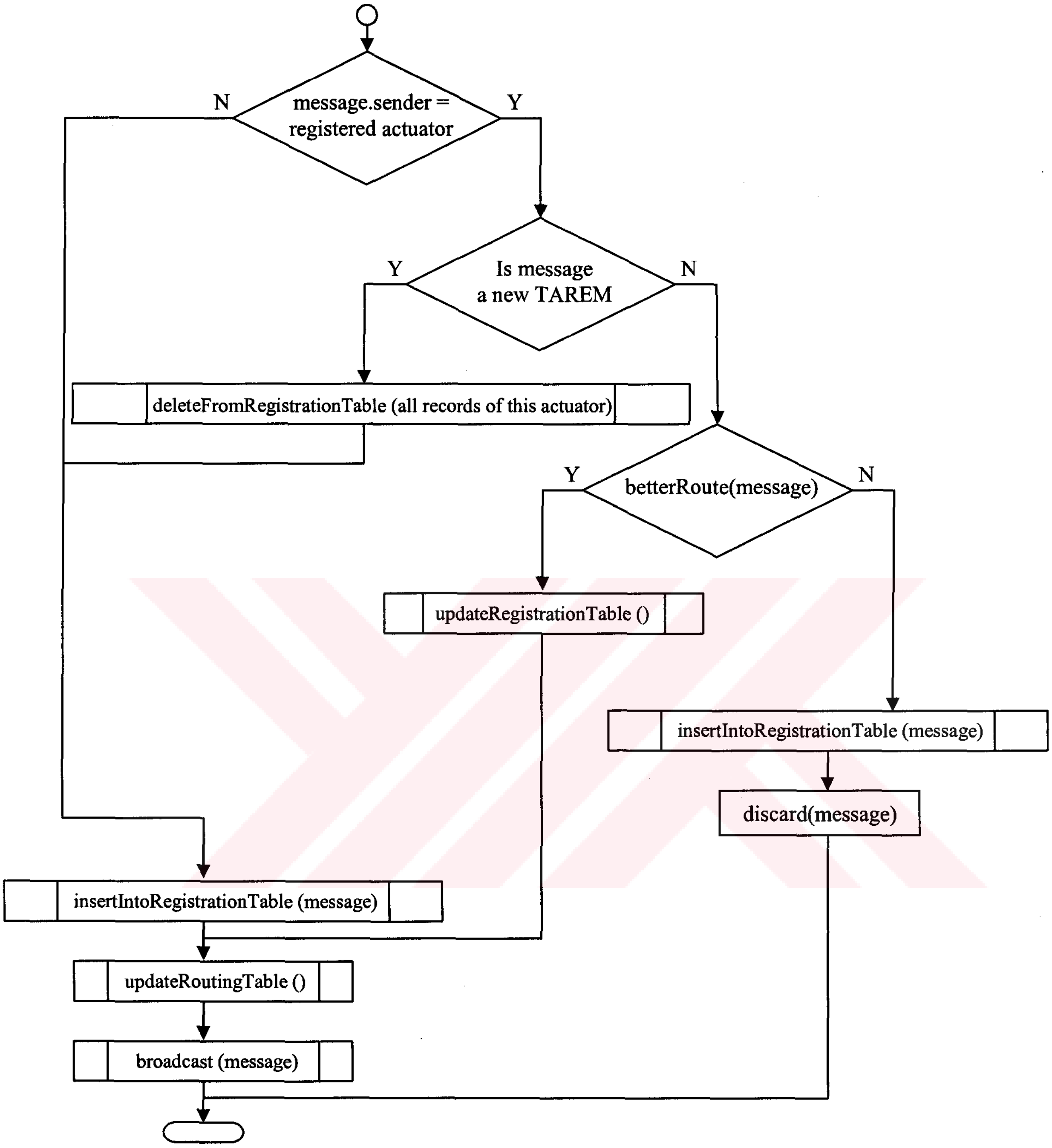
Table 2. The routing table *of Node c in Figure 16.*

Task(s)	Uplink Node Id
t_1	a
t_1	b
t_2	b
t_3	b

a. Task Registration Process

An actuator broadcasts a TAREM for the tasks that it is interested in. A *TAREM* is composed of the fields shown in Figure 15.a: *node identification (node_id), actuator identification (actuator_id), echelon, minimum power available (minPA), total power available (totalPA) and task(s).* node_id is the identification of the sending node. When an actuator broadcasts a TAREM, it initializes the node_id field with zero and this field updated by the nodes that repeat the message. Every node that repeats a TAREM replaces the node_id field with its own id. Echelon means the minimum number of hops required to reach a node from an actuator. The totalPA is found by summing up the power available in every *node* along the route. The minPA is the power available in the *node* that has minimum power along the route. We will give more detail about echelon, minPA and totalPA in section III.C.1.c. A node that relays a TAREM adds its power available (ownPA) to the totalPA. It also replaces minPA field with its ownPA, if the ownPA is lower than the minPA value. Before transmitting a TAREM, the actuator initializes the echelon and totalPA with 0 and the minPA as the maximum possible PA value.

Sensor nodes use Flow Chart 1 to relay TAREMs. If a node receives a TAREM, it first checks if the TAREM is sent by a new actuator. When the registration table does not have any entry for the actuator which sends this TAREM, it means that this actuator is not registered by the node. Therefore, the message is inserted into the registration table as a new record. On the other hand, if the node has any entry in the registration table for this actuator, it means that the node registers this actuator. Then, the node checks this message to determine whether this TAREM is received previously. In other words, it is checked that the registered actuator sends another TAREM into the network and the node receives this message or not. If a new TAREM of the registered actuator is received for the first time, node deletes all records of this actuator from the registration table and it inserts this new TAREM into the registration table. Thus, the node can compare with the other TAREMs which will be received later. When the received TAREM is sent by the registered actuator and TAREM is received before, the message is checked to determine if it is reached from a better route based on echelon, minPA and totalPA fields as explained before. If it is a better route, the registration table is updated. Otherwise, the message is discarded.



Flow Chart 1. Flow chart of Task Registration Algorithm.

After updating the registration table if it is needed, a TAREM is relayed in one of the following cases:

- The registration table does not have any entry for the actuator in the TAREM.

- The registration table has at least one entry for the actuator but all entries are sent by the actuator before last received TAREM.

- The registration table has at least one entry for the actuator and the last received TAREM is reached from better route than the current route of this actuator in the registration table.

When the node decides to relay the TAREM according to the above criteria, routing table is rearranged. Then, the uplink_node_id, echelon, minPA and totalPA fields in the TAREM are modified as explained before, and the modified TAREM is broadcasted. Note that our task registration scheme is different from flooding because sensor nodes relay only selected TAREMs.

We can explain how TAREM dissemination process runs by using the example shown in Figure 16. The TAREM sent by Actuator *A* is relayed by both Nodes *a* and *d*. Lets assume that Node *c* receives the TAREM relayed by Node *a* before the TAREM relayed by Node *d*. As soon as Node c receives the TAREM from Node *a*, it checks its registration table and finds out that this is the first TAREM from Actuator *A*. Therefore it inserts a record into its registration table, i.e., the first record in the registration table shown in Table 1. Then it inserts a new record for Task *1* through Node *a* into its routing table because Actuator *A* registers for Task *1* and the best route available in the registration table for Actuator *A* is through Node *a*. Since the received TAREM triggers a routing table update, the TAREM is modified and relayed by Node *c*. The TAREM relayed by Node *d* is also received by Node *c*. Although an actuator already in the registration table sends this TAREM, and the task registered by the actuator is also in the registration table, the relaying node is *c* which is a new uplink node for Actuator *A*. Therefore a new record is inserted into the registration table. However, we do not need to update the routing table because the current route for Actuator *A* and Task *1* is better than the new route. Therefore, Node *c* does not relay the TAREM received from Node *d*.

b. Task Deregistration Process

Task deregistration process is very similar to task registration process. A task deregistration message is made up of three fields as shown in Figure 1.b.: *node_id, actuator_id* and *task* fields. When an actuator does not need data from a task registered before, it broadcasts a task deregistration message. A node that receives a TADREM deletes the records related to the specified actuator and task from registration table. Then the routing table is updated according to the new registration table. If there is a need to change the routing table, the TADREM is repeated. Otherwise it is discarded. Therefore, some deregistration messages may not be relayed even by the first node.

We assume that the actuator B registered for Task *1* and *2* before as shown in Figure 16. When the actuator B does not need t_2 anymore, it broadcasts a TADREM for t_2. The node b receives this TADREM and updates the record of this actuator in its registration table. Then, it checks its routing table if there is a need to change. Since the actuator B does not require t_2, the node B has not got any node to convey t_2, either. Therefore, its routing table is changed and this TADREM is broadcasted to the other neighbor nodes of the node b. The same algorithm is processed by all received nodes.

c. Power Efficiency

We mentioned about echelon, minPA and totalPA fields in a TAREM before. Remember that *echelon* means the minimum number of hops required to reach a node from an actuator, the *minPA* is the power available in the node that has minimum power along the route and a node that relays a TAREM adds its power available (ownPA) to the *totalPA*.

In Figure 17 where circles represent the coverage areas of nodes, echelons are shown for the actuator. Since only *nodes A* and *B* are in the range of the actuator, they are the only nodes in e*chelon* 1. The nodes that are in the range of *A* and *B* make e*chelon* 2.

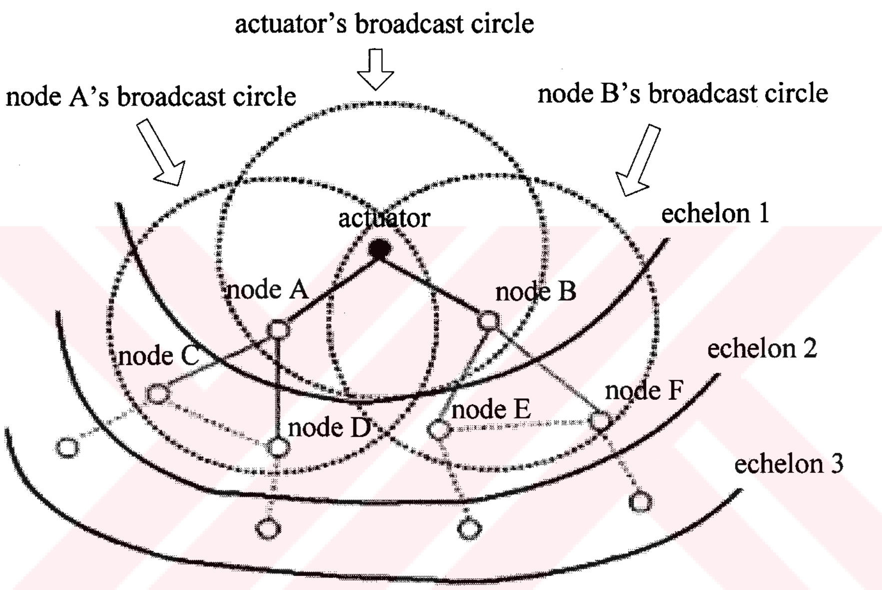


Figure 17. Echelons.

The power efficiency of a route is determined by the echelon, minPA and the totalPA fields in the related TAREM. The route with the lowest echelon is selected as the uplink route. If there are multiple routes that have the same echelon value for the same actuator, the route with the highest minPA is preferred. When both the echelon and minPA values are the same for multiple routes, the highest totalPA route is used. Hence the nodes that have less power available comparing to the other nodes in the network are used less frequently to relay the messages of the other sensor nodes.

This structure will be valid for sensor networks where transmission power is not controlled. In this case, the minimum echelon has the minimum cost because nodes transmit at the same power level at each hop. On the other hand, if nodes have power control capability, echelon will not be valid criteria since a message may be sent two different nodes in the same echelon with different power consumption. So, if the nodes transmit a packet by using a different power level for different nodes like as in [47], minimum hop route is not essentially the most power efficient route. In this case, a field that indicates the total power required conveying a packet from a source to a destination replaces the echelon field in the TAREM. Thus, our scheme can also be used in sensor networks where the transmission power of nodes is controlled according to the transmission distance.

2. Data Dissemination Process

There is two forms to transmit a data packet:

- When a node receives a *data message* and its id is in the *uplink_node_id* field of the message, the node forwards the data packet to the uplink node(s) for the task(s) in the *task* field of the message.

- When a node senses any phenomenon for registered task. It forwards the data packet to the uplink node(s) for the this task.

Before forwarding the data packet, the node replaces the *uplink_node_id* field with the id(s) of the new uplink node(s) which is found by the node from its routing table. Furthermore, when a node has more than one type of data to convey to the same neighbor nodes, the special delimiter is used after the each task and data field pair in the data packet.

For example lets assume that Node *c* in Figure 16 receives a data message, and finds out that Node *c* is in the uplink node list, and this data message is for Task 1. In its routing table shown in Table 2, there are two records for Task *1*: one is for Uplink Node *a*, and the other is for Uplink Node *b*. Therefore, it first replaces the ids in the *uplink_node_id* field of the data message with *a* and *b*, and then broadcasts the received data message, i.e., relays it.

3. Failure Detection and Recovery

a. Death of Nodes

One of the design parameters for PAMR is a threshold value for *critical power level* in a node. This threshold value is determined for the nodes before deployment according to the battery capacity and approximate density of the nodes in the sensor field. A node whose power level reaches to the threshold value can not transmit any message except the message of sensed data that it obtains from sensors in a little period of time. For this reason, it broadcasts a deactivation *message* which consist of node_id field as shown in Figure 15.e The node transmits this message after inserting its own_id into the node id field. All nodes received this message delete relevant records from their registration table. If sender is an uplink in receiver for one of the actuators, the relevant record is also deleted from the registration table and a new uplink is selected from the registration table and then, the routing table is updated. If there is no record in the registration table for relevant actuator except deleted record, the node transmits an *update message* for this actuator to construct a new route.

b. Construct a New Route

There are two cases to broadcast an update message for a node. One of them is as mentioned above.

After broadcasting the data packet, the node listens for a predefined time interval, i.e. *listening time*, to realize whether the uplink node(s) repeats it. If the data packet is not repeated, the record related to the uplink node(s) is removed from the registration and the routing tables. Then, a new uplink node is selected from the updated registration table, routing table is rearranged and the data packet is resent to the new uplink node. This process is repeated until either it is detected that the broadcasted data packet is repeated by the uplink node or all records of this task(s) are removed from the registration table.

Since sensor nodes are prone to failure, it is possible to remove all uplink nodes for the task from the registration and routing tables at the end of this process. In this case, the second condition occurs to broadcast an *update message.* which consists of two fields as shown in Figure 15.c : *node_id* which broadcasts this message and *actuator_id* that the node requires to construct a link to. If there is a neighboring node that has an uplink node for the related actuator_id in its registration table, it prepares a TAREM and broadcasts it. After this, every process in received nodes is the same as task registration process.

4. Data Aggregation in PAMR

Data aggregation has been put forward as a particularly useful paradigm for wireless routing in both WSNs and WSANs. The idea is to combine the data coming from different sources enroute – eliminating redundancy, minimizing the number of transmissions and thus saving energy[5]. This paradigm shifts the focus from the traditional *address-centric* approaches to a more *data-centric a*pproach.

Data aggregation is the combination of data from different sources, and can be implemented in a number of ways. The simplest data aggregation function is duplicate suppression - in the example of Figure 4.b, if sources 1 and 2 both send the same data, node B will send only one of these forward. Duplication suppression is already practiced in commercial wireless messaging networks.

In the experimental set-up described in [5], the form of data aggregation used (duplicate suppression) reduces the traffic by up to 42%. Therefore we use duplicate suppression in our protocol.

The another design parameter in PAMR that we determine for data aggregation to overcome the implosion and overlap problem in PAMR, is *cache time*. The *cache time* expresses how long a node saves the transmitted data packet in its cache to compare with new received data packets. It is determined according to density of the nodes, sensor field range and memory capacity of the nodes.

We use the *cache time* due to following three reasons;

- The implosion problem generally appears in very short time after sensing.
- The nodes generally prefer the same nodes to convey the data packets due to power efficiency while approaching to an actuator.
- The nodes have limited memory capacity.

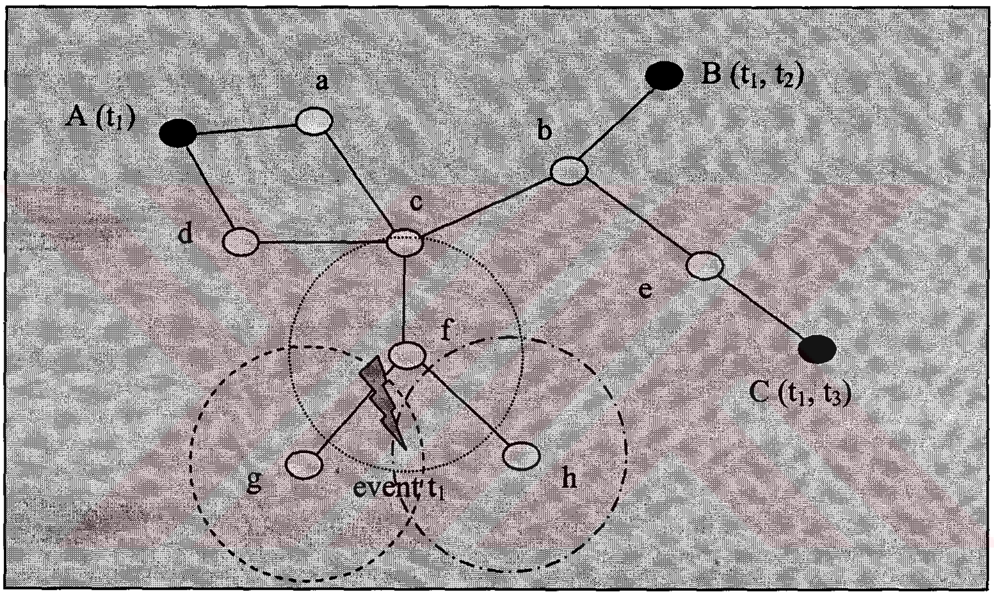


Figure 18. Data aggregation in PAMR.

When a node transmits a data packet, it saves this packet in its cache during *cache time*. After the node receives a new data packet, it checks if new data packet is the same as in the cache. If it does, the new data packet is discarded. Otherwise, the node saves it in the cache and broadcasts to relevant neighbor(s). The node deletes the saved record when the *cache time* of one of the transmitted data packets expires. Therefore, the node does not need a high memory capacity.

For example, node g, h and f in Figure 18 sense an event of t_1 at the same time. We assume that node f broadcasts the data packet before the other nodes. When node f receives the data packet transmitted by node g and h in the *cache time* of event t_1 determined by node f, it checks first its cache. Since these data packets are as the same as in the cache, node f discards these packets instead of conveying.

IV. PERFORMANCE EVALUATION

We evaluate the performance of the PAMR protocol by using the ActSenSim (Actuator-Sensor Network Simulation) developed in the Naval Science and Engineering Institute. In our simulation, we have a sample WSAN with PAMR and Directed Diffusion. We analyze the resulting packet traffic in the network. The detail information about simulation is explained in Appendix-1.

A. SIMULATION SCENARIO

In our sample network, we use following constant in all implementation:

- Varying number of sensor nodes are randomly deployed in a sensor field 160×160 units in size.
- Varying number of events are randomly created in a sensor field.
- Uniform distribution is used to create the events and deploy the nodes.
- The radio range of the nodes is 40 units.
- The sensing range of the nodes is 20 units.
- A node saves the transmitted data packet in its cache during 500 units.
- All sensor nodes can sense all type of data.
- Our simulations run for 36000 units.
- All nodes and the events are stationary.

B. COMPARING WITH DIRECTED DIFFUSION

In this section we compare the performance of PAMR and Directed Diffusion in the scenario where the actuators are stationary. We prefer to compare with Directed Diffusion for the following reasons;

- Directed Diffusion studies more realistic problems.
- Directed Diffusion protocol is data centric like our protocol.
- Directed Diffusion consumes the energy of the nodes more efficient and conscious than many other protocols.
- Directed Diffusion is a well-known and widely used protocol.

The actuators are considered as the sinks of the network in the comparison. We assume the following requirements of interests which are broadcasted by the sinks in Directed Diffusion;

type : the number of type is determined according to user input and a sink interests in all types of data.

interval : it is determined according to user input.

rect : it is determined according to user inputs and covers all the sensor field.

timestamp : it is determined according to user input and it expresses the start time of the event.

expiresAt : it is end of the simulation.

We have three actuators which interest in three types of data in the simulations whose results are shown in Figures 19-21. In addition, the creation of events is started after 1000 units , interval of events is 1000 units and two events are created per interval in simulations whose results are shown in Figures 19-22.

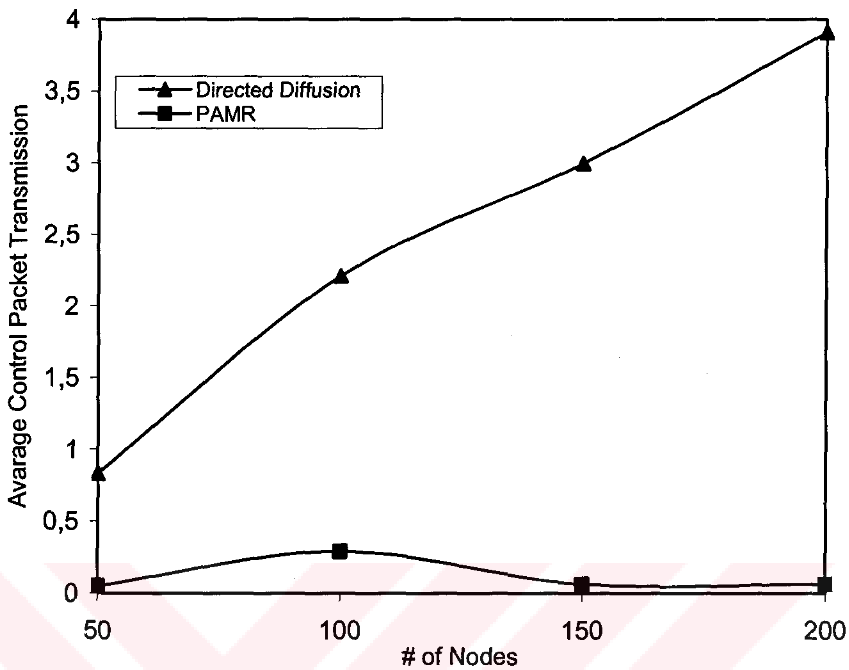


Figure 19. Average control packet transmission rate.

In Figure 19, we show average control packet transmission for varying number of sensor nodes in the sensor field. Remember that control packets are the message formats (a-e) shown in Figure 15. Average control packet transmission measures the ratio of the number of control packet transmission per node to the detected event, given by;

$$\text{Average control packet transmission} = \frac{\dfrac{\text{total number of control packet transmitted in the network}}{\text{number of node in the network}}}{\text{number of detected event}} \qquad (4.1)$$

PAMR is not sensitive to the number of nodes in the sensor field. This proves its scalability. It outperforms directed diffusion, and the performance gains of PAMR become higher for the larger network size.

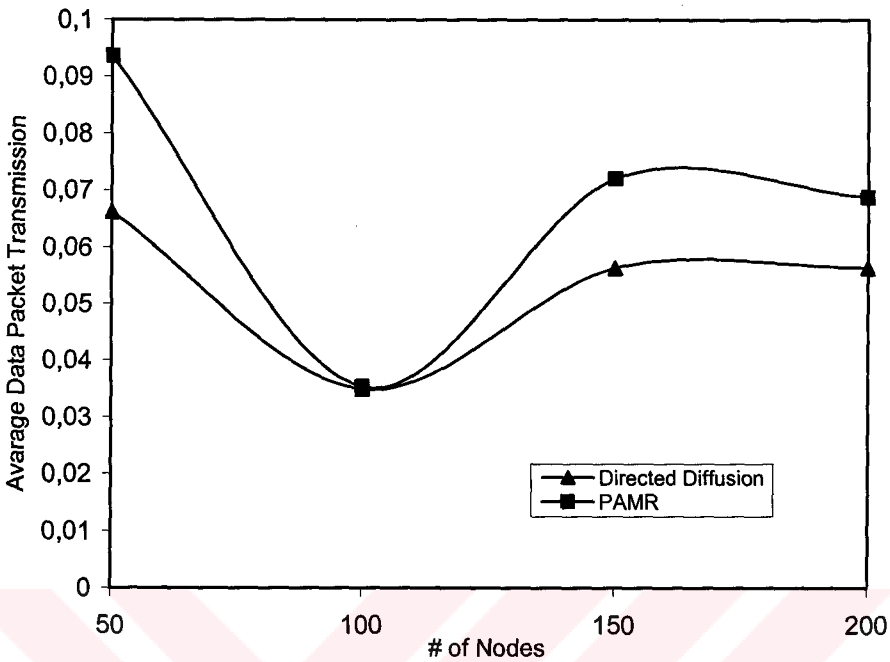


Figure 20. Average data packet transmission rate.

In Figure 20, the average data packet transmission during our simulation is shown for varying number of sensor nodes in the sensor field. The average data packet transmission measures the ratio of the number of data packet transmission per node to the detected event, given by;

$$\text{Average data packet transmission} = \frac{\dfrac{\text{total number of data packet transmitted in the network}}{\text{number of node in the network}}}{\text{number of detected event}} \qquad (4.2)$$

The data traffic is not sensitive to varying network sizes neither in PAMR nor in directed diffusion.

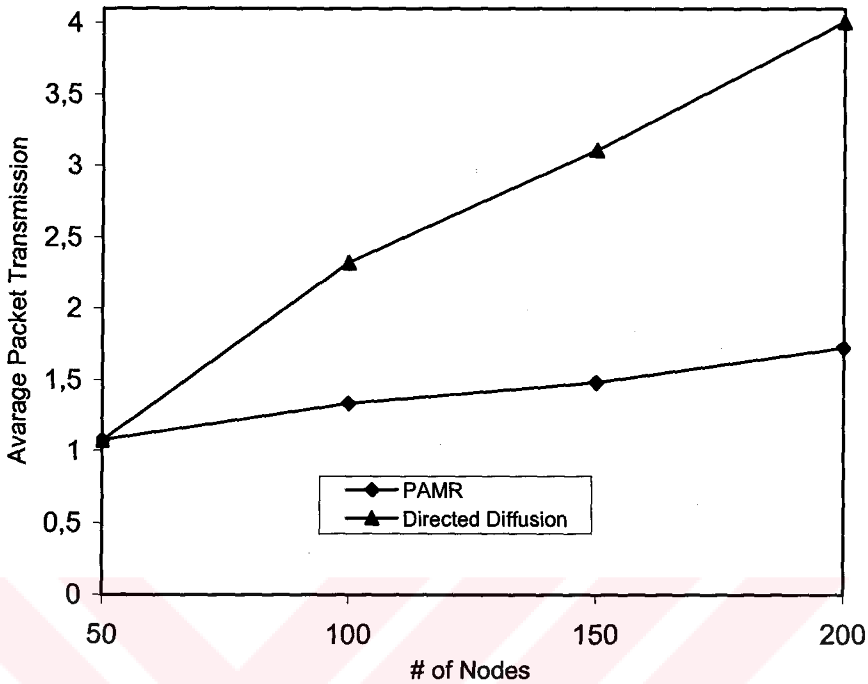


Figure 21. Average packet transmission rate for varying number of nodes.

In Figure 21, we show the average packet transmission rate including both data and control packets for varying number of sensor nodes in the sensor field. The average packet transmission measures the ratio of the number of both control and data packet transmission per node to the detected event, given by;

$$\text{Average packet transmission} = \frac{\dfrac{\text{total number of data and control packet transmitted in the network}}{\text{number of node in the network}}}{\text{number of detected event}} \tag{4.3}$$

Essentially, average control packet in Figure 19 and average data packet in Figure 20 must be used together to analyze the detection and transmission. Because, when a node senses a phenomenon, it finds a path and sends a message to the destination due to control and data packets. Therefore, the graph in Figure 21 is more suitable for analyzing. The number of packet transmissions in PAMR is lower than that in directed diffusion. Since the energy consumption of each sensor node is dominated by the cost of transmitting and receiving messages[7], the lifetime of the network in PAMR is higher than that in directed diffusion.

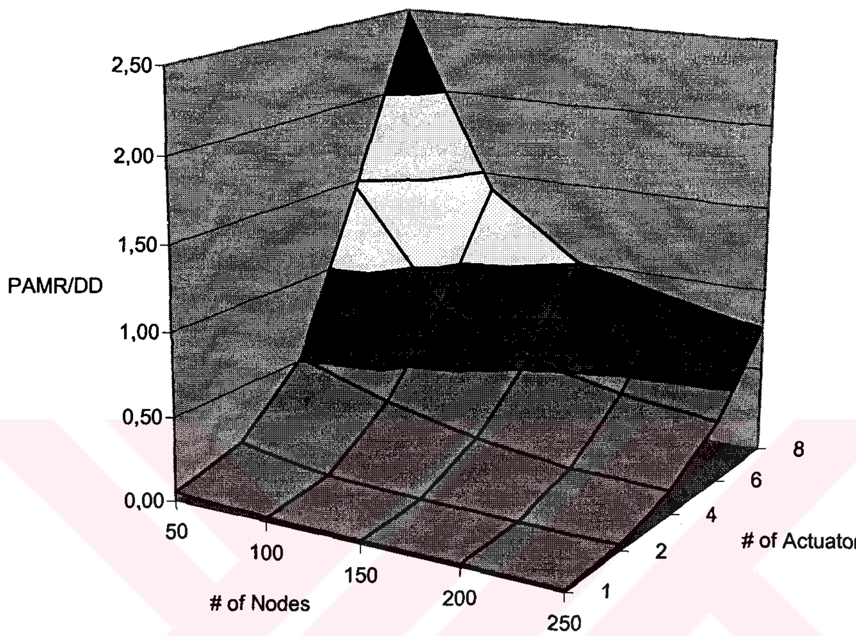
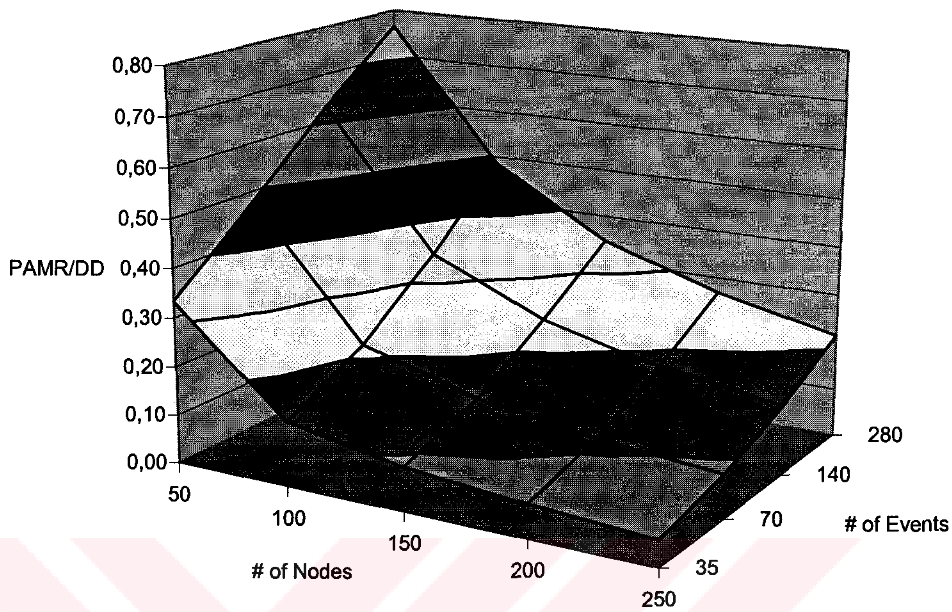
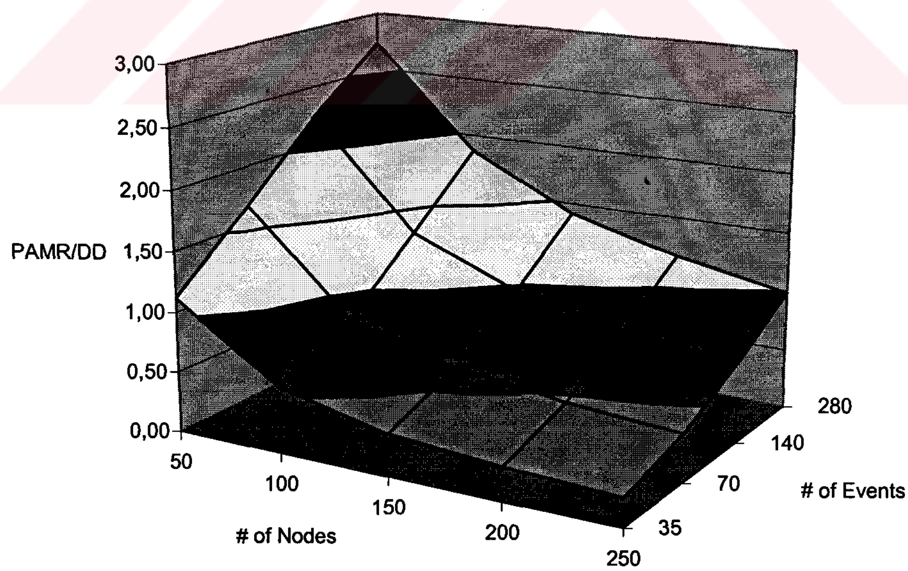


Figure 22. The ratio between the average number of packets transmitted in PAMR and in Directed Diffusion for varying number of actuators.

We analyze the impact of the number of actuators in the sensor field. We measure the ratio between the number of packets transmitted in PAMR and in directed diffusion. As shown in Figure 22, the performance gains of PAMR compared to directed diffusion is higher for the higher number of actuators in larger networks.



(a) The network with 3 actuators.



(b) The network with 6 actuators.

Figure 23. The ratio between the average number of packets transmitted in PAMR and in Directed Diffusion for varying number of events.

We analyze the impact of the number of events in the sensor field. We measure the ratio between the number of packets transmitted in PAMR and in directed diffusion. As shown in Figure 23, the performance in PAMR is not higher than in directed diffusion while the number of nodes in the network remains under a specific value. When there is enough sensor nodes in the network for efficient communication, the gain of PAMR becomes higher. In Figure 23.b. we can see that the transmission in PAMR is less than that in directed diffusion for 70 events after approximately 75 nodes and the gain becomes higher after that point.

In addition, while the number of nodes in the network increases, conveying the packets in the network becomes more efficient by means of multicast trees between the nodes and the actuators.

C. THE IMPACTS OF PAMR IN A WSAN

In Figure 24, we show the packet transmission per node during the simulation for varying number of actuators. The actuators interest in the same 3 types of data. Varying number of nodes are deployed and 70 events are created for each iteration.

The packet transmission in PAMR also increases naturally while the number of actuators increases since all the actuators interest in all type of data. In another respect, while the network becomes large, the packet transmission decreases prominently. Essentially, the multicast trees are constructed more effectively due to the number of sensor nodes. Thus, the packets are delivered to more actuators through the same path.

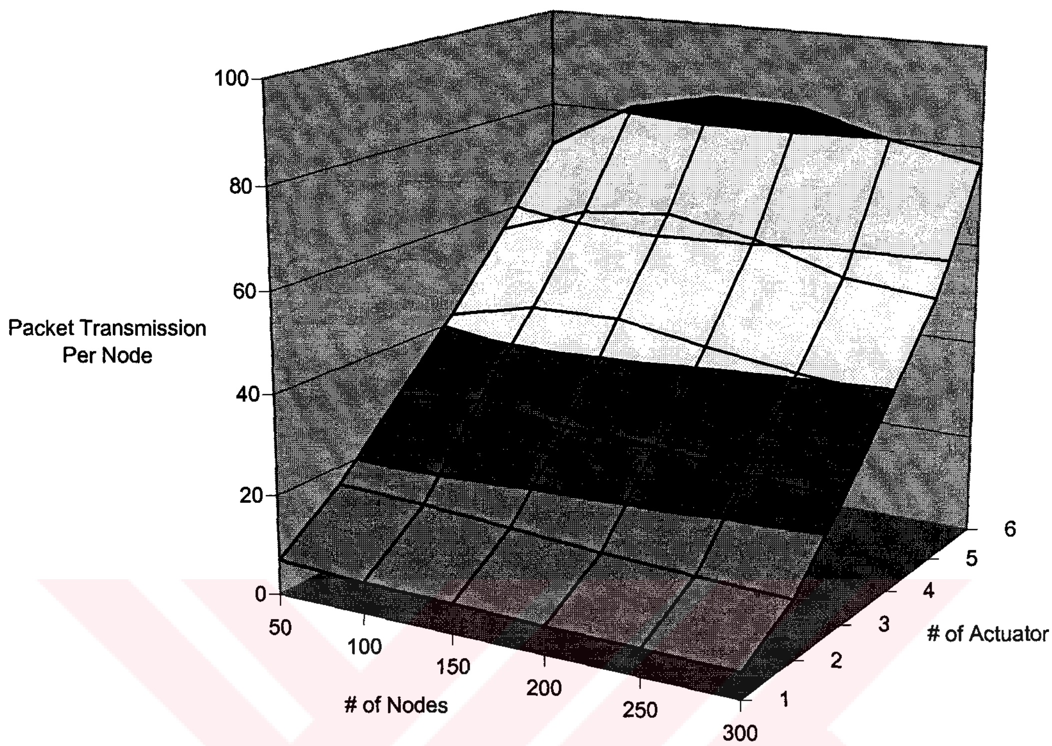


Figure 24. Packet transmission per node for varying number of actuators.

D. THE IMPACT BETWEEN THE ACTUATORS' DEPLOYMENT AND PACKET TRANSMISSION

We analyze the impact between the actuators' deployment and packet transmission in WSANs. We design 4 models and 4 task groups for analyzing as follows;

- Group 1: Each actuator has one task and all tasks that the actuators interest in are different than each other.

- Group 2: Each actuator has two tasks and one of these tasks is required by another actuator.

- Group 3: Each actuator has three tasks and two of these tasks are required by another only two actuators.

- Group 4: The actuators require all the same type of data.

As shown in Table 3, we assume that we have four tasks which are humidity, temperature, proximity and pressure in network. For Group 1; one of the actuators has task of humidity, the other has task of temperature, another one has task of proximity and the last actuator has task of pressure. The other groups are as shown in Table 3.

Table 3. The tasks that the actuators require.

	Group 1	**Group 2**	**Group 3**	**Group 4**
Actuator 1	Humidity	Humidity, Temperature	Humidity, Temperature, Proximity	Humidity, Temperature, Proximity, Pressure
Actuator 2	Temperature	Temperature, Proximity	Temperature, Proximity, Pressure	Humidity, Temperature, Proximity, Pressure
Actuator 3	Proximity	Proximity, Pressure	Proximity, Pressure, Humidity	Humidity, Temperature, Proximity, Pressure
Actuator 4	Pressure	Pressure, Humidity	Pressure, Humidity, Temperature	Humidity, Temperature, Proximity, Pressure

1. Model 1

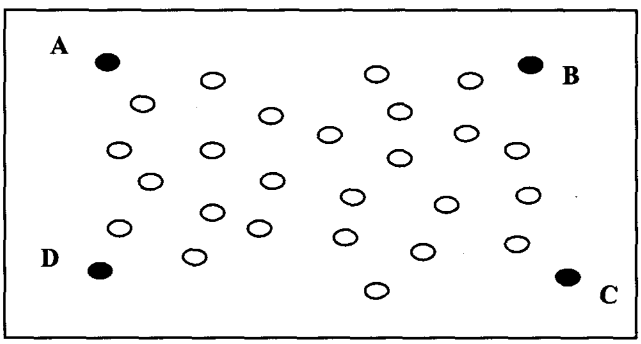


Figure 25.The deployment of the nodes in Model 1.

We define *Model 1 as* collecting data from the sensor nodes by the actuators which are deployed in the approximately corner of the sensor field as shown in Figure 25.

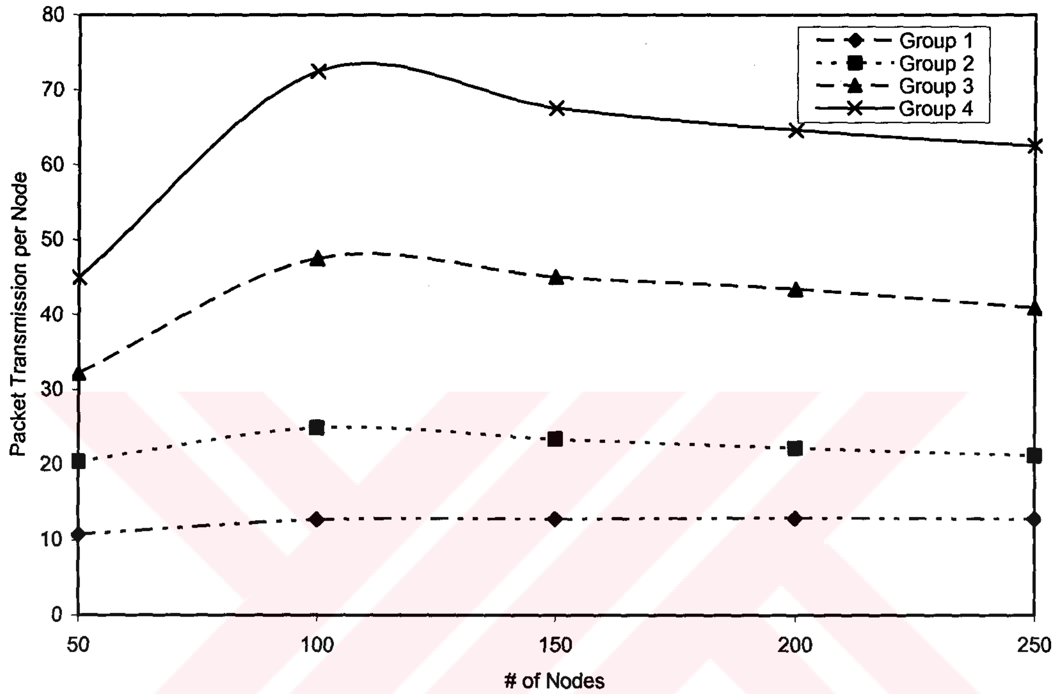


Figure 26. Packet transmission per node for varying number of nodes in Model 1.

In Figure 26, we show the packet transmission during the simulation for varying number of sensor nodes in the sensor field. The packet transmission measures the number of transmitted packets per node. The highest packet transmission in the network is observed in Group 4 where all the actuators require the same type of data. In addition, the packet transmission in all groups increases until the number of nodes reaches approximately 100. Then, the packet transmission in all groups decreases slightly with different ratios.

2. Model 2

We define *Model 2 as* collecting data from the sensor nodes by the actuators which are deployed in the approximately middle of the edges of the sensor field as shown in Figure 25.

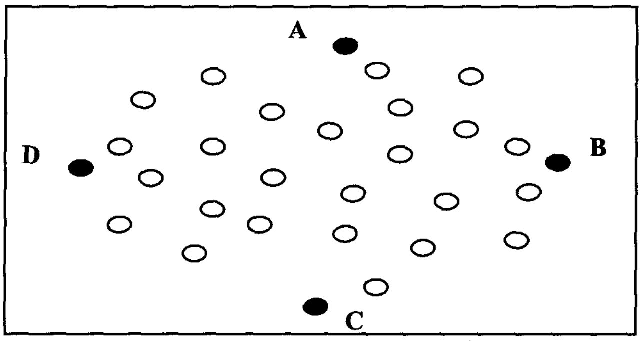


Figure 27.The deployment of the nodes in Model 2.

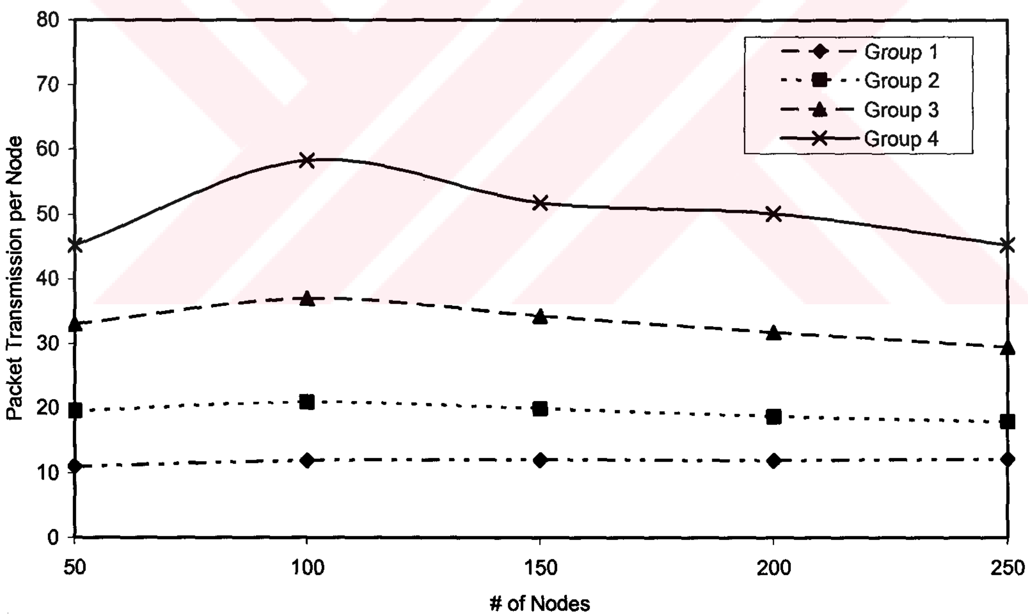


Figure 28. Packet transmission per node for varying number of nodes in Model 2.

In Figure 28, we see that the Group 4 has the highest packet transmission as in Model 1. In addition, the packet transmission in all groups again increases until the number of nodes equals approximately 100. On the other hand, the number of transmission in each group is less than model 1 and the packet transmission decreases faster as compared to Model 1.

3. Model 3

We define *Model 3 as* collecting data from the sensor nodes by the actuators which are deployed in approximately the center of the sensor field as shown in Figure 29. Note that the actuators are in the sensor field different from Model 1 and 2.

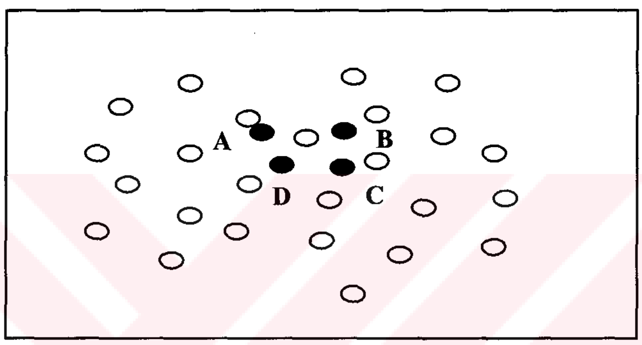


Figure 29.The deployment of the nodes in Model 3.

In this model, we can easily see that packet transmission is less than other models impressively as shown in Figure 30. There are two main reasons for that;

- While the sensed data packets are conveyed through the paths, they are relayed through too few different paths by copying. In another words, the data packets essentially are not conveyed through different path to be delivered to the different actuators since the actuators are very near to center of the sensor field.

- The distance between actuators are not large.

Therefore, a data packet which is required to deliver two different actuators is conveyed through almost the same path. Thus, the differences of packet transmission in groups are less than the other models.

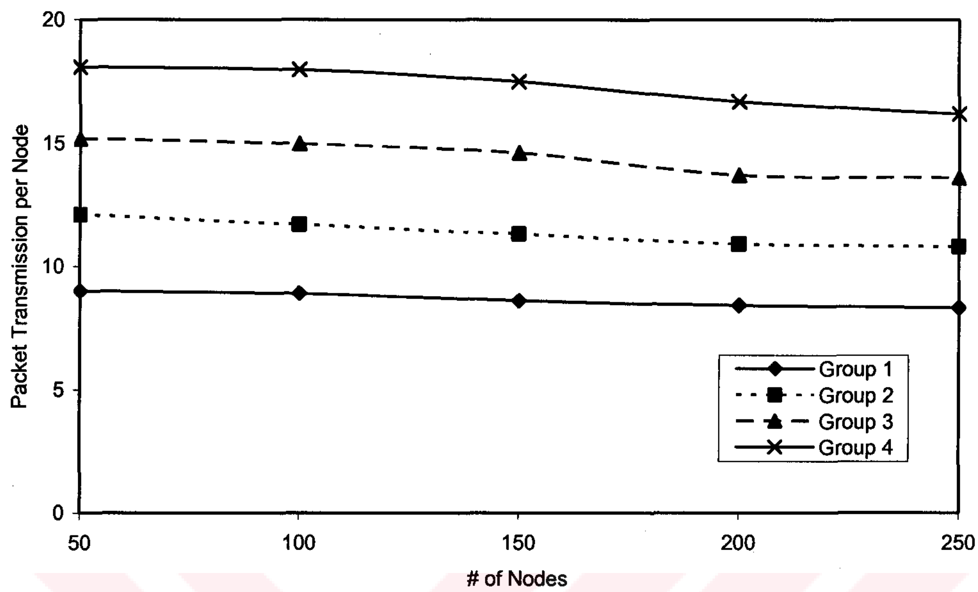


Figure 30. Packet transmission per node for varying number of nodes in Model 3.

4. Model 4

We define *Model 4 as* collecting data from the sensor nodes by the actuators which are deployed randomly in the sensor field according to uniform distribution as shown in Figure 31. Note that Figure 31 is shown for an example of an iteration and the actuators are essentially deployed in different location for each iteration.

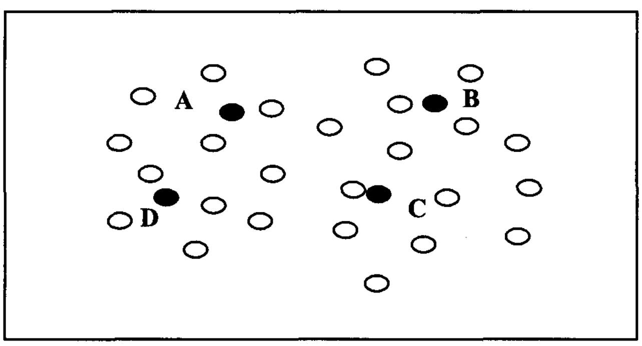


Figure 31.The deployment of the nodes in Model 4.

In Model 4, the packet transmission is not sensitive to varying number of nodes as shown in Figure 32. While the number of nodes in the network increases, a great difference except a little decreasing in the packet transmission is not seen in even Group 4 which has maximum packet transmission according to other groups. Although the nodes are deployed randomly, constructing the efficient multicast trees may be seen one of the major reasons for this.

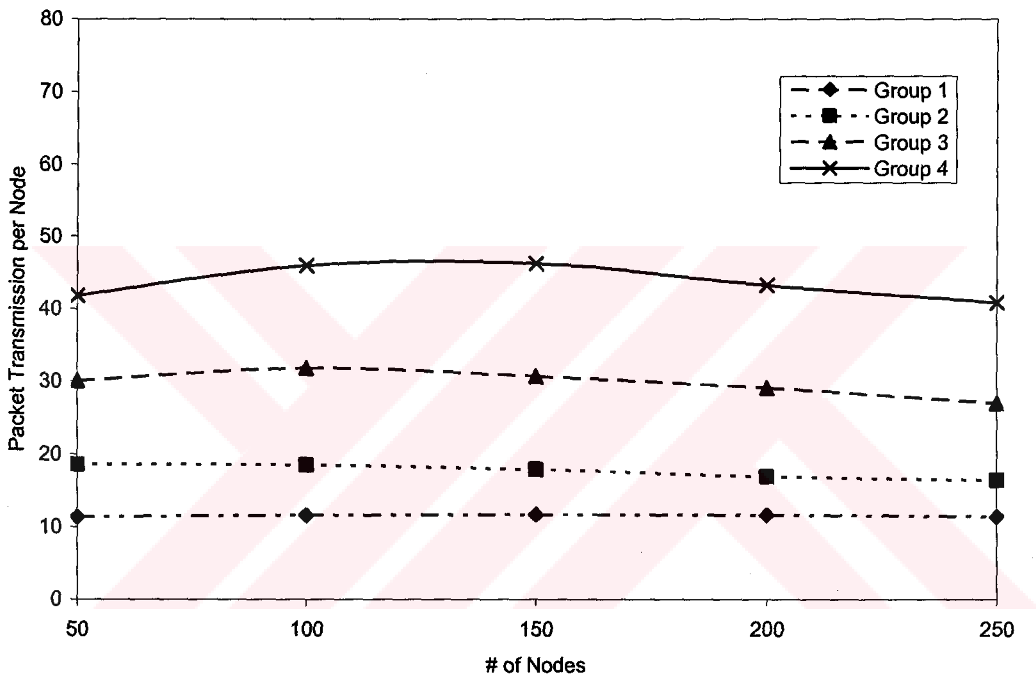


Figure 32. Packet transmission per node for varying number of nodes in Model 4.

V. CONCLUSION

Wireless Sensor and Actuator Network which may be called a new group of sensor network compose of tiny sensor nodes which can have more than one sensor and the actuators which control the devices attached to them base on the sensed data. The collecting process of the sensed data from environment is the same in WSNs. In addition, the energy consumption is the most prominent challenge as in WSNs. The most research is about how to prolong the lifetime of the network. If the power consumption of a sensor node in the network reduces effectively, the node lives much more time. As a result of that, the lifetime of the network is prolonged. Hence, we start to review the routing protocols in WSNs at first since the power consumption during transmitting and receiving a packet is higher than in idle mode or computing mode of a sensor node, then we see that the existent routing protocols are not suitable for WSANs. The major difference between WSNs and WSANs is the destination to which the sensed data is conveyed by the sensor nodes. The sensed data is forwarded to multiple actuators in WSAN where one actuator may fetch data from multiple sensors and one sensor node may feed multiple actuators. Therefore, power aware many-to-many data centric routing (PAMR) is developed for WSANs.

In our scheme, actuators register the types of the sensed data that they need to the network by broadcasting a task registration message. During the dissemination of the task registration message, the multicast tree for the registered task is updated such that the most power effective route to the actuator that registers the task is included. This multicast tree has a many-to-many relation among the sensor nodes and actuators such that the sensed data generated by any sensor node is forwarded to every actuator that is interested in that type of data.

We evaluate the performance of PAMR using the WSAN simulation program, ActSenSim. We determine three parts for evaluation. In first part, the virtual WSANs use both of directed diffusion and PAMR at a time. In second part, we examine the augmentation of the packet transmission while the number of actuator increases. In the last part, the impact between the actuators' deployment and the packet transmission is examined. We conclude that our scheme is scalable and has high performance gains compared to directed diffusion. In addition, while the number of actuators increases, the packet transmission is also increased until a certain number of sensor nodes in the network. After reaching enough density of the sensor nodes, the packet transmission reduces with certain ratios. Since energy is mostly consumed during the transmission process, reducing the number of packet transmission means that it is managed to prolong the lifetime of the network. Besides, when the actuators are in approximately center of the sensor field or they are close to each other, the packet transmission is less than as compared to other forms of deployment, such as when the actuators are in approximately center of the edge of the sensor field or in approximately corner of the sensor field. In addition, the gain of power when the actuators are in the sensor field is higher than the one when they are at the edges of the sensor field. Moreover, our protocol also complies with the various existing lower or upper layer protocols since the routing decisions become very easy.

WSANs exist more autonomous and more well-known networks in a few years with the advance in distributed embedded systems. We can see a lot of application of WSANs in our daily life in the near future. Therefore, we plan to develop a WSAN with PAMR for daily life or military applications and adapt the mobile actuators and mobile sensor nodes in it.

REFERENCES

[1] D.Estrin, D.Culler, K.Pister, G.Sukhatme, "Connecting the Physical World with Pervasive Networks", IEEE Pervasive computing, pp.59-69, 2002.

[2] W.BHeinzelman, A.P.Chandrakasan, H.Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks", IEEE Transactions on Wireless Communications, Vol. 1, No. 4 pp.660-670, October 2002.

[3] Ian F.Akyildiz, W.Su, Y.Sankarasubramaniam, E.Cayirci, "A Survey on Sensor Networks", IEEE Com.Mag.,pp.102-114, Aug.,2002.

[4] J.Agre, L.Clare, "An integrated architecture for cooperative sensing networks", IEEE Computer Magazine, pp. 106-108, May 2000.

[5] B. Krishnamachari, D. Estrin, S. Wicker, "Modeling Data-Centric Routing in Wireless Sensor Networks", IEEE INFOCOM, 2002.

[6] L.B.Ruiz, J.M.Nogueira, A.F.Loureiro, "MANNA: A Management Architecture for Wireless Sensor Networks", IEEE Communications Magazine, February 2003.

[7] H.S.Kim, T.F.Abdelzher, W.H.Kwon, "Minimum-Energy Asynchronous Dissemination to Mobile Sinks in Wireless-Sensor Networks",ACM SenSys'03, Los Angeles, Nov. 2003.

[8] S.Tilak, N.B.Abu-Ghazaleh, W.Heinzelman, "A Taxonomy of Wireless Micro-Sensor Network Models", Mobile Computing and Communication Review, vol. 6, num. 2

[9] K.Anipindi, "Routing in Sensor Networks", University of Texas at Arlington, 2002.

[10] G.J Pottie, W.J.Kaiser, "Wireless Integrated Network Sensors", Communication of the ACM, vol 43, no 5, May 2000.

[11] J.Elson, D.Estrin, "Time synchronization for wireless sensor networks", Proceedings 15th International parallel and distributed processing symposium, pp. 1965-1970, 2001.

[12] C.Carter, S.Yi, P.Ratanchandani, "Manycast: Exploring the Space Between Anycast and Multicast in Ad Hoc Networks", ACM Mobicom, September 2003.

[13] A.Erdoğan, "Data Aggregation Related Issues In Wireless Sensor Networks And The New Sectoral Sweeper Scheme", Master Thesis, Naval Science and Engineering Institute, İstanbul 2003.

[14] A.Manjeshwar, D.P.Agrawal, " TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks", IEEE IPDPS'01, 2001.

[15] S.Kim, S.Son, J.Stankovic, S. Li, Y. Choi, "SAFE: A Data Dissemination Protocol for Periodic Updates in Sensor Networks", Proceedings of the 23rd International Conference on Distributed Computing Systems Workshops, IEEE 2003.

[16] J.Mirkovic, G.P.Venkataramani, S.Lu, L.Zhang, "A Self-Organizing Approach to Data Forwarding in Large-Scale Sensor Netwoks", ICC'01, Finland 2001.

[17] L.Subramanian, R.H.Katz. "An Architecture for Building Self-Configurable Systems", IEEE/ACM MobiHOC 2000, Boston, August 2000.

[18] J.Kulik, W.R. Heinzelman, H.Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", ACM Mobicom'99, Seattle, 1999.

[19] C.Intanagonwiwat, R.Govindan, D.Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Network", ACM Mobicom'00, Boston, Aug. 2000.

[20] D.Ganesan, R.Govindan, S.Shenker, D.Estrin, "Highly-Resilient, Energy-Efficient Multipath Routing in Wireless Sensor Networks", Mobile Computing and Communications Review, Vol. 1, No. 2, 2002.

[21] S.M.Hedetniemi, S.T.Hedetniemi, A.Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks", Networks vol. 18, pp. 319-349, 1988.

[22] D.Braginsky, D.Estrin, "Rumor Routing Algorithm for Sensor Networks", WSNA, ACM September 2002.

[23] F.Ye, A.Chen, S.Lu, L.Zhang, "A Scalable Solution to Minimum Cost Forwarding in Large Sensor Networks", ICCCN, Dallas, Oct., 2001.

[24] K.Sohrabi,J.Gao, V.Ailawadhi, G.J. Pottie, "*Protocols for Self-Organization of a Wireless Sensor Network*", IEEE Per.Com., October 2000.

[25] R.C.Shah, J.M.Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks", The Proc. of the IEEE Wireless Communications and Networking Conference, Orlando, March 2002.

[26] P.Krishna, N.H.Vaidya, M. Chatterjee, D.K.Pradhan, "A Cluster-based Approach for Routing in Dynamic Networks", Research report supported in part by AFOSR under grant F49620-94-1-0276, Texas Advanced Technology Program under grants 009741-052-C and 999903-029.

[27] W.R.Heinzelman, A.Chandrakasan, H.Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", IEEE Proc. Hawaii Int'l.Conf. Sys. Sci., pp.1–10, Jan. 2000.

[28] S.Lindsey, C.S.Raghavendra, "PEGASIS: Power-Efficient Gathering in Sensor Information Systems", IEEE ICC'01, 2001.

[29] K.Du, J.Wu, D.Zhou, "Chain-Bases Protocols for Data Broadcasting and Gathering in the Sensor Networks", IEEE, Proceedings of the International Parallel and Distributed Processing Symposium, 2003.

[30] F.Ye, H.Luo, J.Cheng, S.Lu, L.Zhang, "A Two-Tier Data Dissemination Models for Large-scale Wireless Sensor Networks", MOBICOM, pp.148-159, ACM, September, 2002.

[31] T.Döğüşcü, E.Çayırcı, V.Coşkun, "Power Aware Data Centric Many to Many Routing in Wireless Sensor and Actuator Networks", Submitted to IEEE Milcom'04, 2004.

[32] V.Rodoplu, T.H.Meng, "Minumum Energy Mobile Wireless Networks", IEEE journal on selected areas in communications, vol 17, no 8, August 1999.

[33] K.Akkaya, M.Younis, "A survey on routing protocols for wireless sensor networks", Elsevier ADHOC 47, DTD=4.3.1, Nov. 2003.

[34] L.Li, J.Y.Halpern, "Minimum-Energy Mobile Wireless Networks Revisited", The Proceedings of IEEE ICC'01, Helsinki, Finland, June 2001.

[35] Y.Xu, J.Heidemann, D.Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing", ACM Sigmobile, Rome-Italy, July 2001.

[36] Y.Yu, R.Govindan, D.Estrin, "Geographical and Energy Aware Routing:a recursive data dissemination protocol for wireless sensor networks", UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023, May. 2001.

[37] L.H.Sahasrabuddhe, B.Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutarial", IEEE.Network,pp.90 -102, January/February.,2000.

[38] Q.Huang, C.Lu, G.Roman, "Spatiotemporal Multicast in Sensor Networks", ACM SenSys, November 2003.

[39] S.Bhattacharya, H.Kim, S.Prabh, T.Abdelzaher, "Energy-Conserving Data Placement and Asynchronous Multicast in Wireless Sensor Networs", MobiSys'03, pp.173-186, 2003.

[40] V.Höltta, "Wireless Sensor and Actuator Networks", Helsinki University of Technology, Control Engineering Laboratory, 2002.

[41] B.P.Gerkey, M.J.Mataric, “A Market-Based Formulation of Sensor-Actuator Network Coordination”, Proceedings of the AAAI Spring Symposium on Intelligent Embedded and Distributed Systems, pp.21-26, March 2002.

[42] I.Essa, “Ubiquitous Sensing for Smart and Aware Enviroments”, IEEE Personal Communications, October 2000.

[43] J.Liu, M.Chu, J.Liu, J.Reich, F.Zhao, “State-Centric Programming for Sensor-Actuator Network Systems”, IEEE CS and IEE Com.Soc, 2003.

[44] A.Kaminskyi, H.P.Bishof, “Many-to-Many Invocation: A New Paradigm for Ad Hoc Collaborative Systems”, Rochester Institute of Technology, IT Lab Technical Report 2002.

[45] Averill M.Law, W.David Kelton,“Simulation Modeling & Analysis”, Chapter 7, Second Edition, International Editions 1991.

[46] P.Appelkvist, “Mechatronics Design Of a Robot Society – A Case Study of Minimalist Underwater Robots for Distributed Perception and Task Execution”, PhD Thesis, Automation Technology Laboratory, Helsinki University of Technology 2002.

[47] M.Kubisch, H.Karl, A.Wolisz, L.C.Zhong, J.Rabaey, “Distrubuted Algorithms for Transmission Power Control in Wireless Sensor Networks”, IEEE WCNC, March 2003.

APPENDIX-1 : ACTUATOR-SENSOR NETWORK SIMULATION (ACTSENSIM)

ActSenSim is a time-based simulation software running on Windows Operating System. It is written in Visual C++ 6.0 and used for modeling WSANs. ActSenSim in which the results of the simulation are written into the output files gives us the opportunity of testing our protocol and comparing with directed diffusion. We still goes on to develop to our simulation program.

When the program is started, the main menu bar is visible. At first, the user must select the option named "Information Input and Start Simulation" under option named "Sensor And Actuator Network" on menu bar as shown in Figure 33.

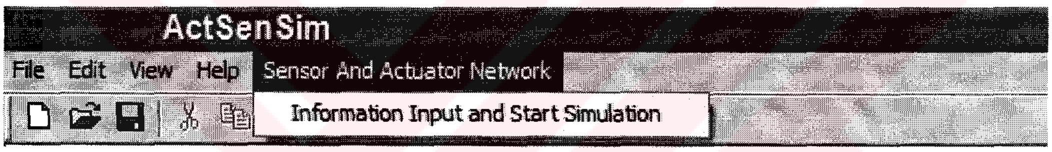


Figure 33. The main menu bar in ActSenSim.

The main screen is visible after this selection as shown in Figure 34. Now, the user must enter the information about the network which will be created. There are five main parts in this screen. In first part, the user enters the number of sensor nodes, actuators and data types which will be sensed, transmitting and sensing range in a node, the power supply of a node and cache time that is one of the design parameters of PAMR.

After all these inputs, The nodes in the network are created by pressing the button named "Create the nodes". Then, the part in which the user can enter the actuators' task(s) is visible. The user can easily see the task(s) of the actuators to which is assigned on the part named "Screen". The user sets the simulation time, i.e. time of the iteration, and the number of iteration in the simulation, i.e. the number of repetition in the simulation. After the sensor field range is entered the user sets the start time of the event, interval of events and the number of events per interval. At last, if the user wants to examine the results of directed diffusion protocol, the relevant check box must be selected before the user presses the button named "Start the simulation". Otherwise, the simulation is started for only PAMR. The program informs to the user about the simulation like simulation time, which level is running...etc. by using the part named "Screen".

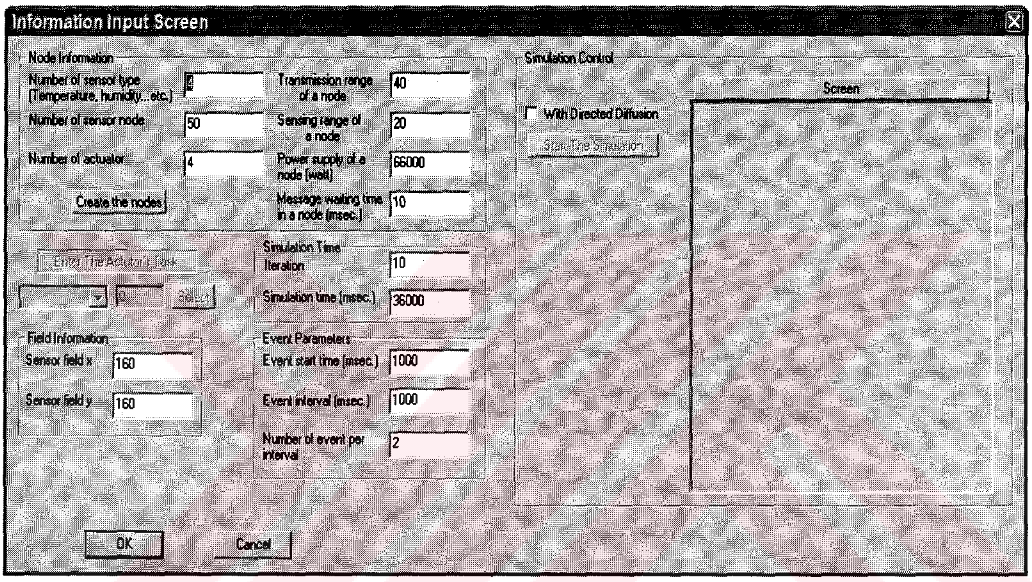


Figure 34. The main screen of simulation program.

A. DESIGN FEATURES

In ActSenSim, there is one major class; the *cnode*. The cnode class has all the attributes which are needed to use both directed diffusion and PAMR protocol. The cnode_sensor and cnode_actuator classes are derived from the cnode class. After cnode_actuator and cnode_sensor classes are defined, memory allocation is done according to the number of sensor nodes and actuators.

The number of events are another important parameter. It is computed according to the inputs of the user. So, memory allocation is also done dynamic. The events have type, value, time and location variables.

There are three important array for the nodes;

- "sensedData" array: When an event occurs, the information of this event is written into the "sensedData" array of the nodes which can sense this event according to the sensing range. Thus, we simulate that a node obtains the data from sensors.

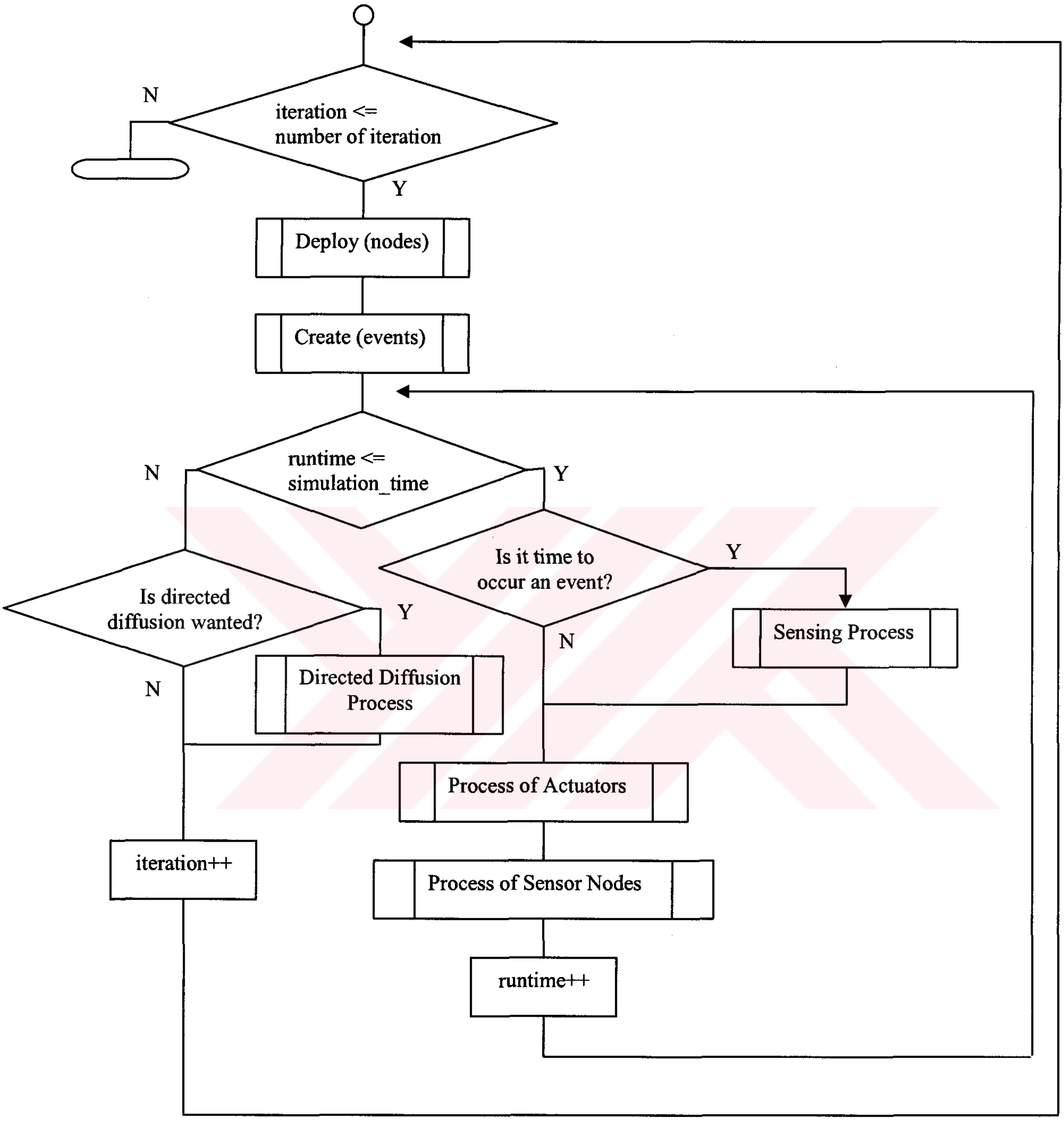
- "TransmitMessage" array: When a node to decide to transmit any packet according to algorithm (PAMR/directed diffusion), it writes the information of the packet into the this array. Then, when the transmit process is active, it is transmitted.

- "ReceiveMessage" array: A node writes the information of the packet which is in its "TranmitMessage" array into this array of the neighbor nodes which can receive this packet according to the transmitting range. Thus, we simulate the transmission between the nodes.

B. HOW THE SIMULATION RUNS

Our simulation flow chart is shown in Flow Chart 2. At first, the nodes are deployed according to uniform distribution which is mentioned about in [45]. Note that the random number generator function in this book is used for all uniform distribution. Then, the occurrence time of events is computed and the events are created and deployed.

While increasing the runtime, events are checked at first. If runtime equals the occurrence time of an event, the nodes which can sense this event according to sensing range obtain the value, time and type of data.



Flow Chart 2. Flow chart of our simulation.

Next process is for the actuators. The actuators are checked if they are in critical power level. Then, broadcasting and receiving processes for a packet are done in the actuators. If they broadcast any packet, they transmit it to the neighbor sensor nodes which are in transmitting range. If they receive any packet, they check which type of this packet is and then, they process it.

Another process is for the sensor nodes. At first, "sensedData" array of the sensor nodes are controlled. If they have any data which they sense, they transmit it to relevant neighbor node according to PAMR/directed diffusion. Note that the nodes which are death and in critical power level can not transmit. After checking the sensor node's power level, the nodes are controlled if they have any new received packet. If they have, the packet is processed according to PAMR/directed diffusion.

All above processes are repeated respectively until the time equals the runtime. Then, if the user wants to obtain the results of directed diffusion protocol, it is processed as the same as in PAMR but with its rules. At last, all processes for both of PAMR and directed diffusion are started again until iteration variable equals the value that the user inputs.

C. OUTPUT FILES

We analyze the simulation results by using the output files. The simulation requires a folder named "data" in the same directory of the program for output files. The four output files are created during the simulation:

Result_Simulation : This file contains the information of all iterations. The following eight parameters are written into this file.

- The number of *control packet* and *data packet* which are transmitted in the network,
- The number of *received events* ,i.e. the total number of events that are detected by all actuators separately,

- The number of *received data*, i.e. the total number of data packets that are received by all actuators,

- The number of *detected event*, i.e. the number of events that are detected by all actuators collectively,

- The number of *sensed event*, i.e. the number of event that at least one sensor node senses,

- *NoUplink.*, i.e. the number of sensor nodes that have no uplink,

- *NumberDeathNode,* i.e. the number of sensor nodes that are death.

Result_Deployment_x_# : This file is created for each iteration separately. The *x* axis of the location of the nodes are written into this file.

Result_Deployment_y_# : This file is created for each iteration separately. The y axis of the location of the nodes are written into this file.

Result_Transmit_# : This file is created for each iteration separately. What is happening in one runtime unit is written into this file during the iteration.