A Survey of Routing Protocols for Wireless Sensor Networks

AMRITA JYOTIPRADA, NAVJEET SANDHU and D. MANIVANNAN
Department of Computer Science
University of Kentucky

Wireless Sensor Networks (WSN) have proven to be exceedingly useful in varied real world applications, solving critical and sometimes lifesaving problems. WSNs are being used to detect forest fires, landslides, earthquakes, study marine biology, air quality, etc. They have also found useful applications in military, industrial settings and security monitoring. Due to such a wide variety of applications, each having different and sometimes unique set of requirements, new contributions are continuously being made. In this paper, we first outline the main design considerations and goals for developing routing protocols for WSNs. Then we use these design parameters as a framework to survey some of the recently proposed routing protocols for WSNs. Finally we do a performance and feature comparison of these protocols.

Keywords: Sensor networks, ad hoc networks, Routing in sensor networks

1. INTRODUCTION

Usually a Wireless Sensor Network consists of thousands of sensing devices that communicate with the base station (also called as sink) using multi-hop data transmission. The base station then sends the collected data to a command center or to a user either directly or over the Internet. The sensors are deployed across a geographical area that needs to be monitored or they may be mounted on moving objects like vehicles, animals, etc. for tracking purposes. Some of the applications of WSNs include:

—Detection of natural disasters such as earthquakes, volcanoes, tsunamis, forest fires, landslides, etc.
—Studying animal behavior such as migration patterns, marine biology.
—Monitoring environmental pollution such as air and water pollution.
—Military applications such as monitoring enemy movements on the battlefield, tracking the position of troops and vehicles.
—Health industry applications such as monitoring patient health.
—Weather monitoring.
—Industrial applications such as monitoring the engine parameters like oil pressure/temperature, engine RPM, etc.
—Structural monitoring in large buildings, bridges, etc.
—Security applications such as home intrusion detection.

A typical sensor node in a WSN consists of the following components:

Energy harvester. It is used for generating electricity from ambient energy sources such as sunlight, wind, vibration, heat, etc. This component is optional and may not be present in all types of WSN nodes.

Authors' addresses: Amrita Jyotiprada, Navjeet Sandhu and D. Manivannan, Department of Computer Science, University of Kentucky, Lexington, KY 40506. Email: {amrita.jyotiprada, navjeet.sandhu}@uky.edu, mani@cs.uky.edu,

Energy storage. These are typically batteries, but if an energy harvester is present then supercapacitors can also be used (with possibly unlimited recharge cycles). In a WSN node, energy is consumed for sensing, routing (processing the control/data packets), transmitting and receiving the packets.

Sensors. These are small sensing devices that sense the occurrence of a particular type of event within their sensing area. These events could be changes in temperature, pressure, motion, visibility, position, or the presence of harmful chemicals, radioactivity, etc. Each node may have more than one sensor for sensing different types of events. The sensors generate analog signals in response to an event which are then passed to an analog to digital converter (which could be a separate device or performed by the processor unit).

Positioning system. These devices are used to determine the position of the sensor nodes. These could be either Global Positioning Systems (GPS) which are more expensive, or they could be based on triangulation technique. Triangulation technique makes use of the signal strengths from known points to determine a nodes position within the network. This component is optional and may not be present in all types of WSN nodes.

Processor. Its primary functions are signal processing, data/control packets processing and controlling other components of the sensor node. Low cost microcontrollers are used instead of general purpose microprocessors, as they consume less power compared to the later.

Memory. The processor usually has some internal cache, but in some cases there might be extra memory present in the form of flash memory.

Wireless transceiver.. Transceiver is a device which is responsible for transmitting and receiving the control/data packets. These are usually radio devices but can also be acoustic, optical or infrared devices depending upon the transmission media and energy constraints. E.g., for underwater WSNs use of radio or optical transceivers are not possible because of high attenuation of radio waves and high scattering of optical waves underwater for distances greater than 100 meters. Therefore for underwater WSNs, acoustic transceivers are used as sound waves can travel long distances much more efficiently without suffering from high attenuation and scattering.

Figure 1 shows the structure of a typical wireless sensor network and the components of a sensor node. Energy efficiency is usually the primary design consideration while developing a new WSN protocol. However, there could be several other requirements that need to be satisfied. The applications of WSNs are diverse and they may be deployed in completely different environments, each having a unique set of ambient conditions that must be handled. Additionally, each of these applications could have their own set of requirements, e.g., some applications would need real time data delivery, and others may want secure and reliable data delivery. This is why there are several WSN protocols, proposed over the years, each designed to address a specific set of application requirements.

In this paper, we present a survey of routing protocols for WSNs that have been proposed in the recent past. Even though there have been other surveys on WSN protocols, our objective is to study and present a broad range of protocols that address a very diverse set of application requirement that have been proposed recently. In [Al-Karaki et al. 2004], the authors present routing challenges and design issues in WSNs. They study and classify several routing protocols based on network structure and protocol operation. In [Yick et al. 2008] issues in WSNs related to operating system support, supporting standards, storage, physical testbed, control and management are presented. They classify and compare various physical layer, data-link layer, network layer and transport layer protocols for WSNs. However, [Al-Karaki et al. 2004] and [Yick et al. 2008] were published in 2004 and 2008 respectively. The survey [Radi et al. 2012], published in 2012, focus their study only on some of the existing multipath routing protocols for WSNs and compare various multipath routing techniques from network application point of view. In this paper, we focus our study on a diverse set of WSN protocols that were published in recent years. For example, some of the protocols we study are designed for energy harvesting

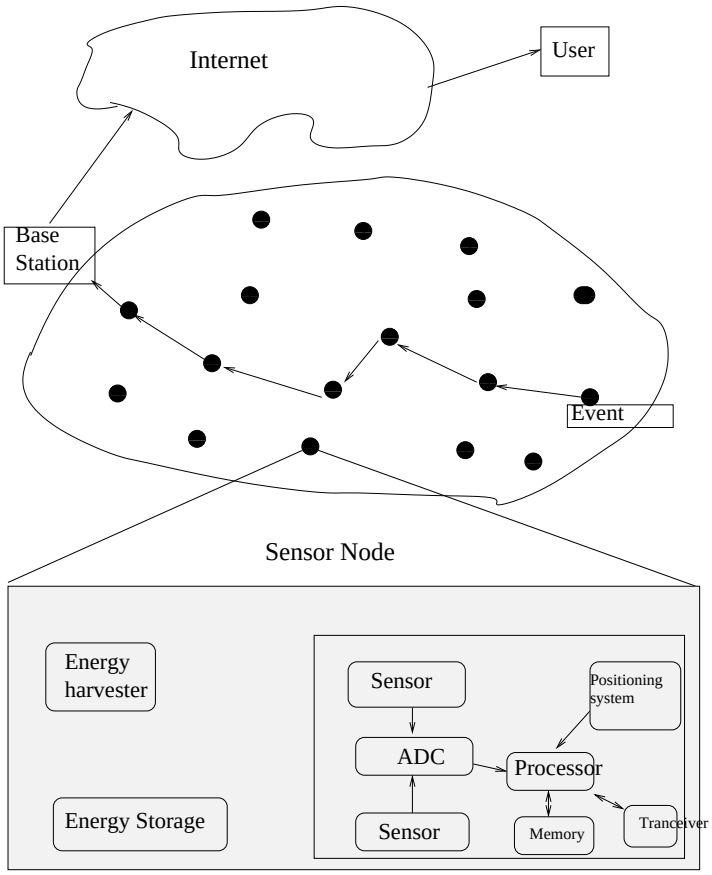

Figure 1: Components of a Wireless Sensor Network.

sensor nodes, some provide QoS guarantees, while others support mobility and so on. The rest of the paper is organized as follows. In Section 2, we outline various design considerations and goals that are critical for designing a routing protocol for WSN. In Section 3, we use these design parameters to study and compare a select set of protocols published in the literature over last five years. Section 4 presents a comparative analysis of the discussed protocols. Finally, we conclude the paper in Section 5.

2. DESIGN CONSIDERATIONS AND GOALS

There are several factors that influence the design of routing protocols for WSNs. Which of these factors have higher significance depend on the application area of the WSN. For example, if we are designing a WSN for disaster prevention such as detection of forest fires, tsunami, volcano, etc., then real time data delivery and delay guarantee are the most critical requirements. On the other hand, if we are designing WSNs for military or security applications then sender authentication and secure data transmission are of utmost importance. Sometimes the environment in which a WSN is deployed is also a critical design consideration. For example, underwater sensor networks for studying marine biology, offshore exploration, pollution and environmental monitoring, etc. make use of acoustic transceivers instead of radio or optical transceivers. Following are some major design considerations and goals of wireless sensor networks.

2.1 Topology

Topology of a WSN depends on the deployment strategy of the sensor nodes. It could be manually deployed in a predetermined topology or it could be randomly scattered across a two or three

dimensional area. The density of sensor nodes could be sparse or dense. The sensor nodes could be homogeneous i.e., all nodes have similar capabilities such as same processing power, energy, transmission range, etc. or they could be heterogeneous i.e., some of the nodes have longer transmission range or more energy.

The topology is usually ad hoc. In an ad hoc WSN there is no preexisting infrastructure and each node participates in routing. The routes are dynamically computed based on the topology and state of the sensor nodes in the network. Clustering approach can also be used for deploying ad hoc WSNs. In a cluster based WSN, the network could be configured such that the low powered sensors communicate to the high powered sensor nodes (cluster head) within a cluster. The cluster heads then form a backbone network for inter-cluster communication for sending data to the base station. Clusters for WSNs can also be formed dynamically based on available energy of nodes. All these factors have significant impact on the design of routing protocols.

2.2 Transmission Media

The transmission media determines the type of transceivers that will be used in the sensor nodes. For most of the terrestrial WSNs, radio transceivers are used. In some applications such as intrusion detection, optical transceivers (lasers) are used. But the use of optical transceivers require clear line of sight and is also affected by the environmental conditions such as presence of smoke, dust, etc.

For underwater WSNs, acoustic transceivers are used because radio waves suffer from high attenuation and optical waves suffer from high scattering, at distances greater than 100 meters. However, sound travels at a slower speed ($\tilde{1}$500 m/s underwater) compared to radio or optical waves. The bandwidth is limited to few tens of kHz due to high environmental noise at frequencies lower than 1 kHz and high transmission loss at frequencies higher than 50 kHz. Underwater acoustic channels also suffer from multipath and fading problems. Therefore, the type of transceivers used affect the design of protocols.

2.3 Localization

Each sensor node in a WSN needs to know its own position within the network. However the nodes could be randomly scattered and the senor nodes may not know their position at the time of deployment. This problem of identifying a sensor nodes own position is called localization. There are several techniques for solving the localization problem. One such technique is use of global positioning system (GPS). However the cost of GPS devices could be prohibitive for large scale dense WSNs, which may contain hundreds or thousands of sensor nodes. Another technique involves triangulation, which makes use of the signal strengths from known points to determine a nodes position within the network.

Additionally, for some routing protocols a sensor node also needs to know the position of other nodes (the destination node or the base station). For such scenarios a location service that enables the nodes to share their location information with other nodes is used. An example of distributed location service is Grid Location Services.

2.4 Energy

Energy is one of the most important criteria for designing protocols for WSNs. This is because most of the sensor nodes are powered by limited energy sources such as batteries. So the operational lifetime of the network depends on the lifetime of these energy sources. If a WSN protocol is not energy efficient, then the sensor nodes will end up consuming more energy for routing and this will reduce the network lifetime. This is undesirable as in most cases it is not feasible to replace/recharge the batteries. Therefore to prolong the network lifetime, energy efficiency is one of the principal design goals of a WSN protocol.

Sensor nodes can also be powered entirely by ambient energy without the need of batteries. The nodes use ambient energy harvesting for converting the ambient energy such as solar, wind, heat, vibration, etc. to electricity. The generated electricity is stored in supercapacitors which

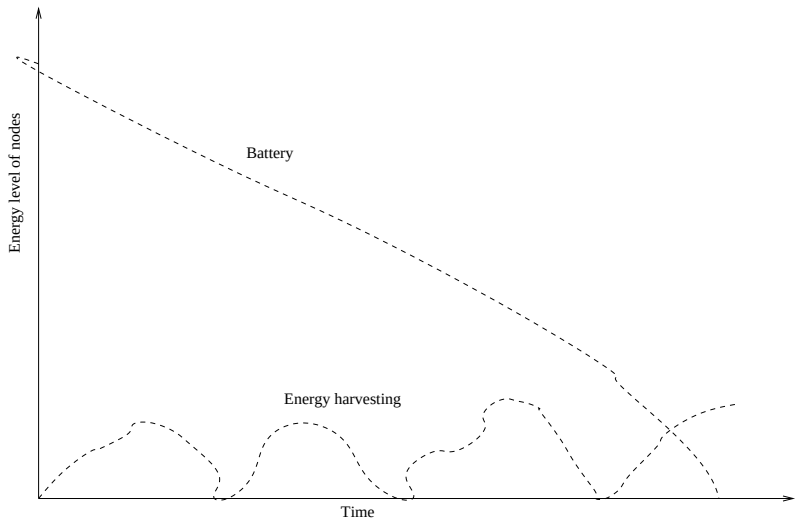

Figure 2: Energy characteristics of sensor nodes

have virtually unlimited recharge cycles. However, the energy characteristics of sensor nodes that are powered by ambient energy harvesting are very different from those powered by batteries. Therefore the protocols for such kind of WSNs must handle the unique energy characteristics for ensuring efficient and optimal routing.

2.5 Synchronization

Some protocols make use of the local clocks of sensor nodes to determine the transmission, receive and sleep cycles. For such protocols, the local clocks of the sensor nodes in a WSN needs to be synchronized. This is all the more important in WSNs that use TDMA MAC protocols. If the local clocks of the sensor nodes in a WSN are not synchronized then this would result in high collisions, which in turn will result in high retransmissions and wastage of limited energy resources. The longevity of these two types of energy are given in Figure 2.

2.6 Fault Tolerance

Fault tolerance is another crucial consideration in the design of resilient WSNs. In a WSN, some of the sensor nodes may fail because of physical damage or their batteries are exhausted. Sometimes the links may be blocked because of some physical obstruction. In such cases, the WSN should still be able to perform its function i.e. send the sensor data to the base station. To build a fault tolerant WSN, the protocols should be designed such that they should be able to dynamically compute new paths to route around the failed nodes or links.

2.7 Data Aggregation

Data aggregation techniques are used to combine the data from multiple sensor nodes to reduce the amount of data that need to be transmitted to the base station. In a WSN, an occurrence of an event may be sensed by more than one sensor node resulting in the same event data being generated by multiple nodes. Data aggregation helps in reducing the transmission of these duplicate data packets, thereby increasing the efficiency of the network and also helping in power conservation. It also helps in improving the scalability of the network and prolonging the lifetime of the network, as there could be significant amount of data that is generated in a large sensor network.

2.8 Coverage

The sensing range of a sensor node is limited. So, for effective monitoring or tracking, coverage is an important design consideration. Sensor deployment strategies can be used to maximize the sensing area coverage. Additionally, there are several protocols that try to maximize the sensing area coverage by coordinating a nodes sleep and wakeup times so that the sensing area is effectively monitored or the target is continuously tracked.

2.9 Scalability

Scalability is usually a common design consideration for all routing protocols. A WSN can have hundreds or thousands of sensor nodes, each generating data for sensed events. All this sensor data need to be sent to the base station. If the routing protocol is not designed to handle these large number of nodes then it may not be suitable for such large scale WSNs. Several techniques are used for ensuring scalability. One such technique is data aggregation and compression. Data aggregation can be used to reduce the number of duplicate packets that are transmitted. Whereas data compression can help reduce the size of the data packets that are sent. However, power consumed for data compression must also be kept in mind.

2.10 Security

There are several aspects of security that need to be considered in a WSN. These include:

Data confidentiality. It ensures that no other node except the sender and receiver are able to comprehend the data. This is usually done by establishing a secure channel between the communicating nodes. A common technique is to encrypt the data, using shared secret keys, before transmitting it.

Data Integrity. It ensures that the data is not modified by a malicious node. Message authentication code (MAC) is typically used of providing data integrity. The sender nodes generates a MAC using the data to be transmitted and a secret key. The generated MAC is appended with the message and sent to the receiver node. The receiver node can then verify the integrity of the received message by computing the MAC, using the received data and the shared secret key, and comparing it with the appended MAC.

Data freshness. It is important for a node or base station to know the freshness of the received data. This helps to discard any stale data which may no longer be useful.

Sender authentication. This technique allows the receiving node to verify the authenticity of the sender. It is important to block any malicious node from injecting false or spurious data in the network. MAC can be used for sender authentication.

There are several WSN applications that need secure data delivery. E.g., military applications that involve monitoring the enemy movement or tracking the position of troops and vehicles.

2.11 Quality of service

Some WSN applications may also have quality of service requirements such as real time data delivery. This requires the sensor data to be delivered to the base station within a certain delay bound from the time of event occurrence. Any delay in data delivery would render the received data useless and may have severe implications. For example, WSNs used for warning systems for tsunami, forest fire, enemy movement, volcano, etc. require real time data delivery. However, the WSN protocols designed for guaranteeing quality of service also need to be energy aware. This is because WSN nodes have limited power and if the protocols are not energy efficient, it would result in reduced network lifetime.

Next, we use these design parameters to study select protocols published in the literature over last five years.

3. ROUTING PROTOCOLS

Over the years, several routing protocols have been proposed for WSNs. Each of these protocols tries to address a unique set of application requirements or tries to improve on some of the earlier protocols. There are several ways to classify these WSN routing protocols.

Based on the role that each sensor node performs in routing, WSN protocols can be classified into flat or hierarchical cluster based routing. In flat routing, all nodes in the network perform the same role. The sensor data is forwarded in a multi-hop fashion using the neighbor nodes in the network. In cluster based routing, the network nodes are logically subdivided into smaller groups, called clusters. Each cluster has several sensor nodes and one or more (usually one) cluster heads at any given time. The cluster heads are either elected dynamically among the sensor nodes or pre-assigned at the time of deployment. The sensor nodes within a cluster forward their data to the cluster head which forwards it to other cluster heads or directly to the base station. A cluster head may also aggregate and compress the data before forwarding. Cluster based routing has several advantages. It not only provides an effective way to reduce the energy consumption, but also uses less bandwidth and is robust and scalable.

Several other ways to classify the WSN protocols include QoS based, multipath based, query based, etc. In the rest of this section we study some of the WSN protocols proposed in last 5 years. We classify these protocols based on topology, support for node mobility and quality of service.

3.1 Cluster-Based Routing Protocols

As mentioned in section 2.1, in cluster-based WSNs the network nodes are grouped into clusters. Each cluster has a node called cluster head. All the member nodes of a cluster send their data to the cluster head. Next, the data is routed to the base station by using inter-cluster routing, i.e. by sending the collected data from one cluster head to another, till it reaches the base station. In this section we study some of the recent protocols for cluster-based WSNs. We further classify the protocols based on whether or not they support node mobility.

3.1.1 Cluster-Based Protocols that Support Node Mobility

3.1.1.1 Cluster Based Routing Protocol for Mobile Nodes in WSN (CBR Mobile-WSN)

Routing protocols such as DSDV [Perkins et al. 1994], DSR [Johnson et al. 1996], AODV [Perkins et al. 1999] proposed for wireless ad hoc networks are not suitable for WSNs as they require high power consumption. The flat multi-hop routing protocols designed for static WSN do not support mobility in WSN and so do the hierarchical routing schemes. LEACH-Mobile [Kim et al. 2006] (which is a Low Energy Adaptive Clustering Hierarchy-Mobile) protocol is able to support mobility of sensor nodes but it leads to high packet loss and high power consumption. The CBR Mobile-WSN protocol, proposed in [Awwad et al. 2009], is an adaptive time division multiple access (TDMA) scheduling and round free cluster head protocol that implements low packet loss technique along with efficient power consumption.

The CBR Mobile-WSN protocol has two phases, setup phase and steady state phase. In setup phase, the cluster heads are randomly elected by the network nodes based on the received signal strength. The TDMA schedule is also determined in this phase. A sensor node sends registration message to its cluster head to become its member and the cluster head then creates a TDMA schedule and sends the schedule back to the sensor node. The cluster heads are assumed to be stationary. Once the cluster head is selected, it broadcasts an advertisement message to rest of the sensor nodes by using carrier sense multiple access with collision avoidance medium access control (CSMA/CA MAC) protocol. In the advertisement phase, the non-cluster head sensor nodes keep their receivers on to receive the advertisement message from their cluster head. Once, the nodes receive advertisement message from the cluster heads, the received signal strength is compared to decide which cluster it wants to join. Once, the sensor node decides the cluster to join, it sends registration message to inform the cluster head via CSMA/CA MAC protocol. In

the schedule creation phase, the cluster head creates a TDMA schedule based on the number of nodes in the cluster and assigns each node a time slot in which the node can transmit. This schedule is broadcasted to all the sensor nodes in the cluster. Then, a node sends data to its cluster head in the TDMA scheduled time slot in steady state phase. A sensor node switches its radio transmitter on, adjusts its transmission power and sends its data, on receiving a data request from the cluster head. It minimizes energy dissipation by turning off the radio at the end of the data transmission.

If a cluster head does not receive data from its member in response to a data request, then the packet is considered lost. The membership of this sensor node, under its cluster head, is also lost. On the other hand, the sensor node also tries to establish membership with a new cluster to avoid packet loss if it doesn't receive data request message from its cluster head for some time. This may happen if the sensor node has moved out from its cluster. If a sensor node receives the data request message from its cluster head but it doesn't have data to send, then the node will not take any time slot and the time slot will be assigned to another sensor node which has data to send. This process ensures efficient bandwidth utilization. Each cluster head reserves some free time slot, so that the incoming nodes from other clusters will be able to join the cluster, and also send join-ack message to the new free cluster. Simulation results show that CBR Mobile-WSN reduces the packet loss by 25% compared to LEACH-Mobile protocol [Kim et al. 2006].

3.1.1.2 Energy Efficient Routing Protocol for WSN with Node and Sink Mobility (EERP-NSM)

The EERP-NSM protocol, proposed in [Sarma et al. 2011], is a hierarchical and cluster based routing protocol that supports mobility of sensor nodes as well as the sink. The protocol has two phases, setup phase and data forwarding phase. The sensor field is divided into logical clusters once the deployment is completed and each cluster contains sensor nodes with different roles, i.e., gateway node, cluster head node and ordinary sensor node. Designing a WSN routing protocol for supporting mobility of sensor nodes as well as the sink is more challenging because of the following reasons:

—The topology of the sensor network becomes highly dynamic because of the mobility of sensor nodes and sink.

—Mobility may also cause link failures because of channel fading during data transmission which results in poor network performance.

—Quick depletion of energy happens when heavy traffic flows through a particular node, which may lead to node failure and result in network partitioning. Hence, unbalanced traffic load would result in node failure.

The protocol must handle all the above problems in addition to the common design challenges in WSN like limited power, memory, processing capability and communication bandwidth. Following are the two phases of this protocol:

Setup Phase. In this phase, the topology of the network is constructed. The self-organization of sensor field happens and it is logically divided into clusters. Each cluster contains one gateway node (GN), two cluster head nodes (CH Node) and the rest are the ordinary sensor nodes (OSN). Various activities in setup phase are:

Cluster Formation. After deployment of the sensor nodes in the field, the sink is responsible for the clustering of the nodes. It is assumed that the sink forms uniformly distributed clusters in the sensor filed.

Gateway Node Selection. The GN gathers data from the CH Nodes and forwards data to the base station. So, the GN must remain connected to the CH Nodes. Therefore, an ideal GN should have higher energy level and lower mobility. Hence, the sink selects GN based on the nodes remaining energy level, location information and mobility level.

Cluster Head Node Selection. Two CH Nodes are selected for each cluster by the base station to reduce bottleneck. The two CH Nodes inside same cluster maintain connectivity inside the

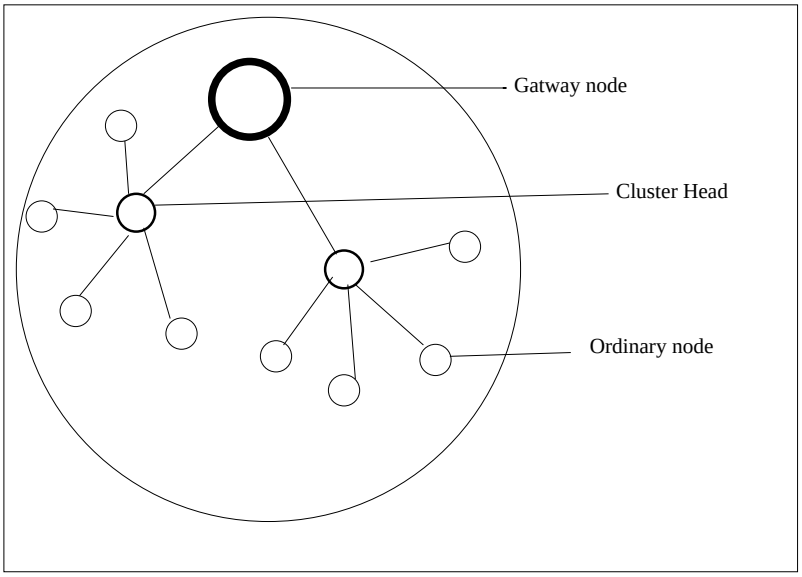

Figure 3: A single cluster formation in the network

cluster. The base station collects the location information from each node inside the cluster. The two CH Nodes are assumed to be geographically uniformly distributed and the two CH Nodes remain connected to the GN. Hence, the OSNs remain connected via direct links to at least one of the CH Nodes.

Communication Pattern for Routing. The sink distributes the communication patterns for OSNs via cluster heads. The CH nodes then aggregate and forward data to the GN inside the cluster and the GN then forwards the data to the base station. The communication patterns for the GN are decided in real time for a particular duration. A GN cant be an intermediate relaying hop more than once as the GNs near the sink will deplete energy quickly due to the relaying of data towards the sink.

Medium Access Control Information. TDMA based time slots are assigned by the Cluster Heads for their respective subordinate nodes.

Data Forwarding Phase. In this phase, the routes are decided based on the roles of the nodes in the sensor field. The OSNs sense the environment and forward the sensed data towards the CH. Then, the CH forwards the collected data towards the GN in the same cluster. Then the GN forwards the data towards the sink. GN may send data either directly or indirectly via a GN present in a different cluster. The CH node and GN do data fusion. They reduce the volume of transmitted data by removing redundant information and keeping useful information intact. Various activities in data forwarding phase are:

Mobility Management inside a Cluster. Routing is complex because of the mobility of the nodes and the Sink. Mobility may lead to link failures and network partition. So, GN is responsible for handling mobility in the network.

Re-clustering Initiation. If a cluster head loses connection with other cluster nodes, then the throughput will be affected, so this will lead to re-clustering. Re-clustering is also initiated when the lifetime of the cluster, determined during cluster setup, expires.

Simulation results show that when compared to LEACH [Heinzelman et al. 2000], this protocol has lower average energy consumption and as a result higher network lifetime. It also has lower control overhead and achieves higher throughput.

3.1.1.3 An Average Energy based Routing Protocol for mobile sink in WSN (AERP Mobile-Sink)

The AERP Mobile-Sink protocol, proposed in [Wang et al. 2008], is designed to support sink mobility. It uses an energy-efficient cluster-based mechanism that takes into account the average energy of a cluster. It also takes scalability, simplicity and system lifetime into consideration. The protocol assumes that the sensor nodes are static and randomly distributed in the sensor field with one mobile sink. It also assumes that the location of the static sensor nodes are known. The clustering scheme is similar to the one used in LEACH [Heinzelman et al. 2000].

To calculate the sinks routing schedule the sink first moves using random way-points mobility model and collects information about the average energy of each cluster and the cluster heads location. It uses this information to create a sink routing schedule queue. The sink then follows the routing schedule to visit each cluster for data collection.

A hello packet is used to notify the cluster heads about the sink's visit to a cluster. It contains cluster id and the time that the sink would require to move through all the cluster heads. The cluster head uses the time received in the hello packet, to determine the schedule for data collection (from the sensor nodes within its cluster). When the data collection time is up, it transmits the data, its current location and average energy to the mobile sink.

3.1.2. Cluster-Based Protocols that do not support Node Mobility

3.1.2.1. Novel Energy-aware Hierarchical Routing Protocol in WSN (EHRP)

The primary objective of the EHRP protocol, proposed in [Mollanejad et al. 2010], is to extend the lifetime of WSNs. EHRP hierarchically groups sensor nodes into clusters, where each cluster has one cluster head. These cluster heads then collect data from the member nodes and send it to the base station. In the first phase of the protocol, the cluster heads are selected and in the second phase a spanning tree is constructed based on the cluster heads for transmitting aggregated data to the base station. EHRP uses a new formula for cluster head selection, which can handle the heterogeneous energy levels better compared to other clustering algorithms. Only one node directly communicates with the base station because of the tree structure. Energy consumption due to all communication is computed using the free space model which helps to save energy and extends the network lifetime. In EHRP, each node has a neighbor table that stores information about its neighbors. The operation of EHRP is divided into rounds and each round has two phases, set-up phase and steady-state phase.

Cluster Formation Round. (Setup phase) All the nodes broadcast hello message within range r, at the beginning of each round. The hello message contains the residual energy of the nodes. Once the hello message is received, each node updates its neighbor table and the cluster head is selected based on a nodes residual energy, distance from base station and its distance from other neighbor nodes. The protocol consumes energy uniformly among all nodes, as the cluster heads are rotated during the entire lifetime of the network which results in extended network lifetime.

Routing Tree Formation Round. (Steady-State phase) In the steady-state phase, the cluster heads broadcast Tree_Msg within a radius R, and these messages contain the residual energy of the node. Based on the residual energy and some other factors like distance from base station, each cluster head then computes its parent node. By following this, a routing tree, rooted at the parent node with the largest computed value among all the cluster heads, is constructed.

All Cluster Head nodes do the following.

(1) Cluster heads broadcast Tree_Msg to the neighbor nodes
(2) After receiving Tree_Msg from all the neighbor nodes, compute distance from the neighbors using signal strength and update neighbor table
(3) Cluster head computes its PN (Parent Node)
(4) If (PN > all neighbors PN)
(5) Set to parent node

(6) Else, set to Child node
(7) Parent nodes broadcast TDMA schedule to their child nodes

Energy consumption by cluster heads per round in EHRP is lower when compared to LEACH [Heinzelman et al. 2000] and HEED [Younis et al. 2004]. This is because in LEACH and HEED, cluster heads send their data directly to the base station but EHRP constructs a spanning tree on the cluster heads. So the data is sent to the base station in multiple hops which reduces the energy consumption by cluster heads.

3.1.2.2 A Clustering Patch Hierarchical Routing Protocol for WSN (CPHRP)

The CPHRP protocol, proposed by [Lin et al. 2010], is designed to improve the network coverage rate and effective network lifetime of WSNs. The protocol assumes that the nodes are static and scattered randomly in the network. It uses a hierarchical multi-path tree routing mechanism to improve energy conservation. The nodes are partitioned into three classes: cluster nodes, sensing nodes and non-sensing nodes. The protocol improves the network coverage rate through clustering patch mechanism i.e., nodes that are not covered by any existing cluster, advertise themselves as a cluster-head. The probability that a node in CPHRP advertises itself as a cluster-head is calculated as

$$CH_{prob} = C_{probe} * \frac{E_{resident}}{E_{max}}$$

Where, C_{prob} is a constant 0.05, $E_{resident}$ is residual energy and E_{max} is initial energy.

Algorithm for cluster-head selection - If only the nodes that are part of a cluster were to perform sensing then the network will have blind areas. This is why CPHRP allows nodes, which do not belong to any cluster, to advertise themselves as cluster heads. If a node is not already a cluster head and it doesn't belong to any cluster and the probability is less than 1, then it chooses a random number u in the interval $(0, 1)$. If $u < CH_{prob}$, the node is announced as cluster-head and the value of CH_{prob} is doubled. If $CH_{prob} > 1$, then the node can directly advertise itself as a cluster-head.

The authors also provide an Algorithm for hierarchical multi-path tree construction of the cluster heads A cluster-head node communicates with the inner-cluster nodes and upper-cluster nodes before forwarding a packet to enable network-load balancing which determines if the packet is forwarded by the upper-cluster node or inner-cluster node or by itself. Using the above mechanisms CPHRP form a hierarchical multipath tree, which is used for routing the sensor data to the base station. Simulation results show that CPHRP achieves better network coverage and better energy conservation compared to HEED [Younis et al. 2004].

3.1.2.3 The Novel Energy Adaptive protocol for Heterogeneous WSN (NEAP)

The NEAP protocol, proposed in [Golsorkhtabar et al. 2010], is a hierarchical clustering, energy adaptive protocol designed to reduce energy consumption and maximize network lifetime in a heterogeneous wireless sensor network. In this protocol, the cluster-heads are elected by probability and cluster formation is based on the nodes current battery power. Nodes send the sensed data to their cluster head which is periodically elected using a modified clustering algorithm. The cluster heads aggregate the data of their cluster nodes and forward the data to the base station. The sensed data of neighboring nodes in a network are usually correlated, so data aggregation can be used to reduce the traffic to the base station. NEAP uses this observation to increase the efficiency of the network.

The protocol makes the following assumptions about the nodes: all nodes of the network are static and spread heterogeneously. Nodes will always have data to send to the end user. All nodes have limited and equal battery power. They have enough power to transmit and reach the base station if needed. The base station is fixed and not located between the sensor nodes. Nodes can adjust the transmit power and each node can support different MAC protocols and perform signal processing functions.

Each sensor node calculates a value $T(n)$ stochastically and generates a random number between 0 and 1. If this random number is less than $T(n)$, the node becomes a cluster head in this round. After several rounds the cluster heads converge. The cluster model is based on confidence value associated with the broadcasts from cluster heads. Confidence value of a cluster head is a function of the following parameters: distance between the cluster head and node, the current battery power of cluster head and number of nodes that are members of the cluster. The cluster model first checks whether cluster head would be able to support the current members at the maximum data broadcast rate, with its current battery power. A node decides to join a cluster if the head of the cluster is able to support it with the remaining power.

To reduce the probability of collision between the REQ messages during the setup phase, CSMA is utilized as the MAC layer protocol. In order to send the data, a cluster head senses the channel to check if anyone else is transmitting using the base station spreading code. If so, then it waits to transmit the data or else it sends the data using the base station spreading code.

The simulation results show that when compared to LEACH [Heinzelman et al. 2000], NEAP has better performance in cluster head selection and forms adaptive power efficient and adaptive clustering hierarchy. It also reduces the energy consumption and hence increases the network lifetime.

3.1.2.4 A WSN Protocol for Disaster Management (WSNPDM)

The WSNPDM protocol, presented in [Sana et al. 2007], is an energy efficient cluster based routing protocol for a hybrid of cellular and sensor networks. It tries to address the issue of data collection in the event of a disaster that may result in collapsed or unreachable base stations. The data collection framework and WSNPDM protocol can be easily deployed in an established cellular network.

Network Model The protocol is designed for a hybrid model of sensor and cellular networks. The network consists of a cellular infrastructure, i.e., several base stations, each serving the nodes within its cell. The nodes within the cell are sensor nodes that communicate with their base station. Additionally, there is an ad hoc relay station (ARS) that is positioned on every shared edge of cells. Zoning is used for clustering and routing, i.e., concentric circular zones are formed in each cell. It is assumed that every ARS supports two types of interface i.e., ad hoc relay interface (to communicate with other ARSs and sensor nodes) and cellular interface (to communicate with base stations of cellular network).

Sensor Network Protocol WSNPDM tries to increase the energy efficiency of a WSN. Its operation is divided into rounds and each round has a setup and data communication phase. The clusters are formed based on the residual energy of the node in the setup phase. Next, the cluster heads prepare a schedule for their respective clusters based on LEACH. In the data communication phase, data is first collected by the cluster heads from their respective clusters and then transmitted to the base station via ARSs.

Addressing Scheme. The protocol assumes stationary base stations and sensor nodes. It also assumes the use of GPS for maintaining up-to-date location information of all the nodes, by the base station. The addressing format is <location ID, Node Type ID>. The location ID identifies the location of a sensor node in terms of polar coordinates (r, θ) with respect to the base station as the origin i.e., $(0, 0)$. Based on the r value, each node belongs to one of the logical concentric circular zones. Each node in the cluster is provided with a Node ID which describes the functionality of the sensor.

Clustering and Scheduling. Each zone area is divided into sub-areas and each sub-area has a unique ID. This sub-area ID is used along with a nodes address for communication. Each sub-area is considered a cluster and at the beginning of each round, a cluster head is selected based on the residual energy of the cluster nodes. Every cluster node transmits a message containing its residual energy, address and the sub-area ID to its neighbor. When a node receives this message, it compares its own residual energy with the residual energy of the other nodes in the same

sub-area. The node with the maximum residual energy is selected as the cluster head for that sub-area, for the current round. Code-division multiple access (CDMA) is used to counteract the problem of radio interference by the neighboring clusters. Each cluster is assigned a spreading code to distinguish the data transmission.

Data Routing. Once the clusters are formed and the transmission schedule is prepared, nodes can send their data to the cluster head in their assigned slots. On receiving the data, the cluster head perform data fusion before forwarding the data to the nearest ARS in a multi-hop fashion. The cluster heads use geographic forwarding for minimizing the energy consumption. Once an ARS receives the data from a cluster head, it forwards it directly to the base station. If the base station has failed then it forwards the data to another ARS till it reaches a working base station.

WSNPDM is compared with LEACH [Heinzelman et al. 2000] protocol and the simulation result shows that it outperforms LEACH in terms of average energy dissipation, system lifetime and successful data delivery.

3.1.2.5 Base-Station Controlled Dynamic Clustering Protocol (BCDCP)

The BCDCP protocol, proposed in [Muruganathan et al. 2005], is a centralized clustering-based routing protocol that tries to increase the network lifetime by distributing the work load among all nodes so that energy consumption across all the sensor nodes is even. The main idea is to utilize base station for all energy intensive tasks such as setting up clusters, routing paths, periodic randomized re-election of cluster heads, etc. As with any clustering-based protocol, the cluster heads collect data from their respective cluster nodes and transmit the data to the base station via other cluster heads. To minimize overload at cluster head, the base station ensures that the cluster heads are uniformly distributed over the sensing area and have approximately the same number of nodes in each cluster.

A class-based addressing scheme, based on the node attributes and geographical location, is used. Each node's address is stored in the form of <Location ID, Node Type ID>. The location ID identifies the location of a sensing node in the specified region of the network and the type ID is based on the functionality of the sensor such as seismic, thermal sensing, etc. The protocol assumes that the base station is not energy constrained and the sensor nodes are homogeneous, immobile and have limited energy. BCDCP has 2 operation phases: setup and data communication.

Setup phase. This phase consists of cluster setup, cluster head selection, routing path formation among cluster heads and transmission schedule creation for nodes in each cluster. In each setup phase all the nodes in the network send their current energy level to the base station. The base station then calculates the average energy level of all the nodes in the network. Based on this, the base station chooses the nodes which have their energy levels higher than the average to form a set S. The cluster formation and head election (from set S) is done using an iterative cluster splitting algorithm as follows:

Step 1 From set S choose 2 nodes s1 and s2 that have maximum separation distance.
Step 2 Group the remaining nodes to either s1 or s2, whichever is closer.
Step 3 Balance the number of nodes in each cluster so that they have approximately same number of nodes.
Step 4 Split the set S into two new sets S1 and S2 based on step 3.

The above steps are repeated by the base station until the desired number of clusters are formed. Next step is setting up lowest-energy routing paths between the elected cluster heads. This is done by first creating a minimum spanning tree connecting all the cluster heads. Then a cluster head is randomly chosen to transmit the data to the base station. This random selection is done so as to distribute the transmission load among cluster heads. The last part of the setup phase is transmission schedule creation. To minimize the collision between the transmitting sensor nodes, sending their data to the cluster head, BCDCP uses time-division multiple access

(TDMA) scheduling. Each node within a cluster is assigned a temporary schedule creation ID (SCID) which determines their transmission slot. E.g., if the cluster head is assigned a SCID 00 then a node with SCID 01 transmits first, 10 transmits second, 11 transmits third and so on.

Data communication phase. This phase consists of data gathering, data fusion and data routing. Each sensor node transfers its sensed data to the cluster head as per the TDMA schedule. This data transmission consumes minimal energy because of small spatial distance between the cluster head and the sensing nodes. Next, the cluster head performs data fusion on the received data to reduce its size. This compressed data is forwarded along with the information required by the base station to identify and decode it. The cluster head to cluster head routing path used for data forwarding is the one created by the base station. BCDCP protocol also uses code-division multiple access (CDMA) codes to handle the radio interference issue caused by the neighboring clusters. A per cluster spreading code is used to distinguish the data transmitted by nodes within a cluster from the nodes in the neighboring clusters. The cluster heads also use the same code to route the data to the base station.

The protocol is compared to some cluster-based protocols like LEACH [Heinzelman et al. 2000], LEACH-C [Heinzelman et al. 2002] and PEGASIS [Lindsey et al. 2002]. Simulation results show that BCDCP reduces the energy consumption by 40% over LEACH and 30% over LEACH-C, thus improving the network lifetime.

3.1.2.6 Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks (TTSR)

The TTSR protocol, proposed in [Du et al. 2007], is an energy efficient secure routing protocol for heterogeneous sensor networks (HSNs). A HSN model typically consists of small number of high-end sensors (H-sensors), large number of low-end sensors (L-sensors) and a base station. The L-sensors form clusters with H-sensors acting as a cluster heads. The H-sensors form the backbone of the network. It is called two tier secure routing because the protocol consists of 2 parts; secure routing within the cluster (among L-sensors), called Intra-cluster routing and secure routing across the cluster (among H-sensors) called Inter-cluster routing.

Intra-cluster routing is concerned with the secure data transfer between L-sensors and their cluster head. Security is implemented using two way handshake mechanism. First, the node with the smaller ID sends a challenge message to its neighbor. The challenge message consists of the sending node's ID and a random nonce encrypted with a shared key and a MAC. The neighbor node responds with a pairwise shared key, a broadcast key and the incremented nonce (all encrypted using the shared key) and a MAC. This pairwise shared key is later used by both these nodes to encrypt the data packets before sending over the network. The two way handshake also helps in detecting unidirectional links. The protocol also has a route setup phase during which all L-sensors form a Minimum Spanning Tree or a Shortest Path Tree rooted at the cluster head. The established routes are then used for data forwarding to the cluster head.

The protocol has a route discovery phase during which it forms shortest path from H-sensors to the base station. These established routes are then used to relay data from cluster heads to the base station. In the event of failure, alternate paths are used for data forwarding.

The protocol uses MAC to address the requirements of data authentication and integrity. Data confidentiality is achieved by the use of symmetric key encryption. TTSR is effective against a variety of security attacks such as Sybil attack, wormhole attack, sinkhole attack, selective forwarding attack, manipulating routing information attack and Hello flood attack. The use of two way handshake mechanism, data authentication, and the route establishment done by the cluster head (where each node sends data only to its parent) helps in defending the network from all these different attacks.

Performance of TTSR is compared against Directed Diffusion [Intanagonwiwat et al. 2000] protocol and found to be better in terms of higher delivery ratio, lower energy consumption and lower end to end delay.

3.2 Flat Routing Protocols

As opposed to cluster-based WSNs, in flat WSNs the nodes are not organized into clusters and there is no communication hierarchy. Each node takes part in the routing process and communication is usually done by forwarding the data to neighboring node(s) till it reaches the base station. In this section we study some of the recent protocols for flat WSNs. We further classify the protocols based on whether or not they support quality of service.

3.2.1 Flat Routing Protocols that support Quality of Service

3.2.1.1 A Delay Guaranteed Routing and MAC Protocol for WSN (DGRAM)

The DGRAM protocol, presented in [Shanti et al. 2010], is an integrated TDMA based MAC and routing protocol that tries to provide deterministic delay guarantees while minimizing the energy consumption by sensor nodes. The protocol assumes a circular sensing area with the sink at the center. It also assumes that the sensor nodes are uniformly distributed all around the sink and are stationary. The circular sensing area is logically partitioned into tiers and blocks. Each node computes, using an algorithm, to identify its tier, block and index within the block. For doing this it needs to know the position of all the other nodes in the network. This is achieved by a short beacon exchange phase where each node tries to learn the position of all the other nodes by exchanging control messages using CSMA protocol. Once the beacon exchange phase is over, each node in the network will know the position of all the other nodes in the network. Using the position information, all nodes can compute the number of nodes in each tier and block and also the number of slots per node. This information in turn helps the nodes to compute the structure and size of the superframe. Size of a superframe is equal to total number of nodes in the network multiplied by slots per node. Next, each node in tier Ci has to compute all the nodes in the tier Ci+1 for which it is going to be the designated receiver. Once all these steps are completed each node would have computed it transmission and receive slots. The actual data transmission is done by coordinated sleep and wake up cycles. Every time a node in tier Ci+1 wakes up to send data in its transmission slot, its designated receiver node in tier Ci wakes up to receive the data in its receive slot. Thus no separate routing protocol is needed.

The protocol provides delay guarantees. So it can be used in WSNs which need some quality of service in terms of delay in data delivery. It optimizes the energy consumption by efficiently managing the sleep and wake up cycles of sensor nodes. However, it has few limitations. First, it is designed for a very specific kind of topology i.e., a circular sensing area with sink at the center and uniformly distributed sensors. So it may not be as efficient for different topologies or uneven distribution of sensor nodes. Second, if several of the nodes in an area fail then the sensor nodes in the outer tiers would not be able to route around these failed nodes. The protocol also assumes that each node is programmed with the total number of nodes in the network and all calculations are based on this parameter. So, if new nodes are added in the network or some of the nodes fail because of physical damage or energy exhaustion, how this change in the number of nodes would be updated in all the nodes in network is not addressed.

3.2.1.2 Distributed Routing Algorithms for Underwater Acoustic Sensor Networks (UW-ASN)

Acoustic Sensor Networks (ASN) are networks where the communication between sensor nodes is done using sound waves instead of radio or optical waves. ASNs are especially useful for setting up underwater sensor networks because sound waves can propagate more efficiently compared to radio or optical waves. However the protocols developed for terrestrial networks that use radio or optical transceivers cannot be used for underwater WSNs because of the unique characteristics of underwater acoustic channel. Underwater acoustic channels have limited bandwidth capacity, generally limited to few tens of KHz. They have higher propagation delay because of low speed of sound (1500 m/s under water) and they also suffer from high bit error rates. This is why Pompili et al. propose a new UW-ASN protocol [Pompili et al. 2010] designed specifically keeping in mind the characteristics of acoustic channel and the three dimensional underwater environment.

The authors proposed two solutions, one for delay-sensitive and the other for delay-insensitive static WSNs. The main objectives of the protocol are to minimize the energy consumption, increase the efficiency of the acoustic channel and reduce the packet error rate. However, the last two requirements are conflicting as efficient channel utilization requires longer packets and lower packet error rate requires smaller packets. To solve this problem, a node sends a train of small packets without releasing the channel. So, when increased efficiency of the acoustic channel is needed, nodes can just increase the train length without increasing the packet size.

For every packet sent, each node jointly selects its best next hop, optimal transmit power and forward error correction (FEC) code. The optimal packet size is set off-line and the strength of FEC is computed dynamically based on the channel conditions. The use of FEC helps reduce the number of retransmissions needed as the corrupted data packets can be reconstructed at the receiver node.

The delay-insensitive routing algorithm allows a node to select its best next hop j* such that j* is closer to the base station and the energy required to successfully transmit a payload bit from node to the sink is minimum. The delay-sensitive routing algorithm is similar to the delay-insensitive algorithm except that it imposes two new constraints:

—The end-to-end packet error rate should be lower than an application-dependent threshold.

—The probability that the end-to-end packet delay be over a delay bound, should be lower than the application-dependent parameter.

The delay-sensitive routing algorithm also does not retransmit lost or corrupted packets at the link layer. The UW-ASN protocol has few limitations. First, if the acoustic channel has high error rate then the FEC will have high overhead. This will result in lower network goodput as the amount of data in each packet will decrease (as the packet size is fixed offline). Second, the protocol assumes that there are no voids in the network i.e., greedy routing is always possible. Even though recovery techniques can be used, they are considered to be out of scope of this protocol.

3.2.1.3. Energy efficient and QoS based routing protocol for WSN (EQSR)

The EQSR protocol, proposed in [Ben-Othman et al. 2010], is a multi-path and QoS based routing protocol. The quality of service parameters that it tries to address are minimizing the delay in data delivery, increasing throughput through data redundancy and higher aggregated bandwidth by splitting and sending the data using multiple paths. It uses the concept of service differentiation, i.e., assigning higher priority to real time data over non-real time data. It also tries to maximize the network lifetime by efficient utilization of energy. This is achieved by using node disjoint multiple paths for load balancing. The protocol uses a light weight XOR-based Forward Error Correction (FEC) algorithm to correct errors. This helps in reducing retransmissions and hence helps in conserving energy.

The protocol first initiates a path discovery phase which is done in three steps. The first step is initialization phase during which each node broadcasts HELLO messages to its neighbors to share its residual energy, available buffer space and interference in terms of signal to noise ratio of the connecting link. Next every node uses a link cost function to identify its most preferred next hop. The link cost function takes in the weighted values of the three parameters shared in the initialization phase. The next step is primary path discovery phase during which the primary path is established by sending route request messages from the sink to its most preferred node. This node, in turn, sends the route request to its most preferred node and so on, till the message reaches the source. Once the primary path discovery is completed, alternative paths discovery phase is run during which alternate node disjoint paths are established. Paths are maintained by appending control information to the data packets. The sending node selects a subset of paths from the total available multi-paths for transmitting data. These paths are further split to handle the real time and non-real time data.

The protocol also maintains two queues, namely, an instant queue for real time data and FIFO queue for non-real time data. A message is broken into segments, which are encoded using an FEC, and transmitted using the established multi-paths. At the receiving end, these segments are decoded and the errors, if any, are corrected using FEC, as long as the bit error is within an acceptable threshold. The decoded segments are then reassembled to construct the sent message.

The concept of service differentiation is the key to handle quality of service requirements. This is because the protocol gives higher priority to the real time traffic by assigning it routes with minimum end to end delay. Multi-path routing helps in load balancing which results in uniform energy consumption of sensor nodes. FEC helps in reducing retransmission of packets due to packet corruption en route. Aggregated bandwidth is improved because of splitting the data and sending it using multiple paths. However, it has few limitations too. The overhead due to queuing packets and encoding packets using forward error correction codes results in increased energy consumption. It assumes a fully connected and dense network, so it may not perform well for sparse networks. The protocol also uses node disjoint paths, so some of the alternative paths may not be optimal and may result in more aggregate energy consumption because of longer paths.

3.2.1.4 Multiconstrained QoS multipath routing in WSN (MCMP)

The MCMP protocol, proposed in [Huang et al. 2008], is a soft-QoS based routing protocol for WSNs. Soft-QoS means guaranteeing the QoS requirements with certain probability. It is close to hard-QoS when this probability approaches 1. Soft-QoS scheme is also different from end-to-end QoS as the paths are formed based on the local link state information. It uses multiple paths between source and sink nodes for providing QoS guarantees. To simplify the computational complexity of identifying routing paths, MCMP uses estimation and approximation of path quality. The main idea is to use probabilistic programming to formulate the optimization problem and then further convert it into deterministic linear programing using some approximation technique. This approach helps reduce the complexity of the problem and makes it easier to solve in the presence of resource constrained sensor nodes.

MCMP focuses on two main QoS constraints, reliability and delay. Both these constraints need to be satisfied in different ways. Reliability is defined in terms of packet delivery ratio and can be improved by multipath routing. Whereas delay can be handled by path diversity. If none of the paths in the network is capable of meeting certain delay requirements then that constraint is not feasible to meet. However if a message needs to be delivered with certain reliability then that can be achieved by choosing a subset of possible paths between the source and the destination. MCMP tries to choose the minimum number of paths, which would satisfy a delay requirement, to conserve energy.

To achieve the soft-QoS, MCMP makes use of a distributed link-based QoS routing scheme. It acquires the local link metrics and uniformly partition the current end-to-end QoS requirements to all downstream hops. All next hops are selected based on the link conditions and the reliability assigned by the preceding node. Routing loops are avoided by storing the minimum distance from each node to sink, in neighbor table. Only those neighbors which have fewer hop counts to the sink are selected for forwarding the packets.

Simulation results show that MCMP performs better than single path and braided multi-path routing proposed in [Ganesan et al. 2001].

3.2.2 Flat Routing Protocols that do not support Quality of Service

In this section, we discuss routing protocols that do not support quality of service.

3.2.2.1 Opportunistic routing in WSN powered by ambient energy harvesting (EHOR)

The ambient energy harvesting process is probabilistic because it depends upon the environmental factors. For example, if a WSN node is solar powered then the charge cycle will depend upon the availability of sunlight. Therefore, the exact sleep and wakeup schedules cannot be computed.

This is why an opportunistic routing protocol named EHOR is proposed in [Eu et al. 2010]. EHOR takes into consideration the unique properties and power characteristics of ambient energy powered WSN nodes. The objective is to maximize the goodput, data delivery ratio, efficiency and fairness of the network. It uses Jain's fairness metric to determine fairness.

EHOR creates logical regions in the network for determining the best nodes for forwarding data while reducing overhead and duplicate transmissions. The best forwarding nodes are generally those that are nearer to the sink than the sender. If a node that is not in the forwarding region receives the data packet, it will just drop it; if its in the forwarding region, then it will compute its region id to determine its transmission priority. Nodes with lower region id have higher priority to forward the data packet. Next, the protocol needs to determine which of the nodes in the forwarding region should transmit the packet to avoid collision. This is handled by assigning a unique transmission time slot to each node based on its region id. So, a node in j^{th} region will transmit only in the j^{th} time slot, if it has enough energy. While a forwarding node is waiting for its transmission time slot, it overhears other forwarding nodes to determine if some other node has already forwarded the data packet. If the packet has already been forwarded by some other node before a nodes transmission time slot, then it simply drops the packet. Every node just transmits one data packet in each charging cycle. Additionally a node stays in receive state only for a fixed period of time, after which it goes back to sleep.

Simulation results show that the use of this "regioning" approach helps in reducing transmission delays and improves goodput compared to conventional opportunistic and non-opportunistic routing protocols. The hop count is also lower. However the simulation results also show that in some scenarios GR-DD [Eu et al. 2009] (Geographic Routing with Duplicate Detection) performs better in terms of throughput.

3.2.2.2 The Fault-Tolerant Routing Protocol for High Failure Rate WSN (ENFAT-AODV)

ENFAT-AODV protocol, proposed in [Che-Aron et al. 2010], is an enhanced fault-tolerant mechanism for Ad hoc On-Demand Distance Vector routing protocol [Perkins et al. 2003]. The protocol designs a backup route scheme by creating a backup path for every node in the network. In case of a link failure, the node makes use of the backup route for next data packet delivery. By doing so, it reduces the number of dropped packets and continues transmitting the data packets in the presence of faults like node or link failures. Hence it increases reliability and availability compared to AODV.

ENFAT-AODV enables fault-tolerant, self-starting, multi-hop routing between participating nodes to establish and maintain a fault-tolerant wireless sensor network. The protocol uses a destination sequence number for each route entry to ensure loop freedom. In case of multiple routes to the destination, a requesting node selects route with the greatest sequence number indicating most fresh route.

The protocol runs a **Main Route Discovery** procedure to find main route to the destination. It broadcasts a Route Request (RREQ) packet for the destination. A route to the source is established at each intermediate node once the RREQ is received. Then receiving nodes checks the Flooding ID and Source IP Address in the packet to find whether the RREQ was received earlier or not. If the node is not the destination node and it doesn't have fresh enough route to the destination, then, it rebroadcasts the RREQ; otherwise, discards the received packet. If the receiving node is the destination or has a fresh enough route to the destination, it generates RREP (Route Reply) packet. Once the source receives the RREP, it records the route to the destination. If multiple RREPs are received by the source, then the route with the shortest hop count is chosen as the main route. During data flow, each node along the main route updates the timers associated with the route and maintains the routes in the routing table. If the route is not used for certain period of time, then that route is removed from the routing table.

Next, the nodes in the main route create the backup route towards the destination (by running a **"Backup Route Discovery"** procedure) by broadcasting a RREQ packet with "Backup flag" set. After broadcasting a backup RREQ, a node waits for RREP packet with "Backup

flag" set from the destination or an intermediate node which has fresh enough backup route information to the destination. When a backup RREQ reaches an intermediate node, it checks the "DistanceToDest" field in the backup RREQ. If the number of hops from intermediate node to the destination along the active path is greater or equal to the "DistanceToDest" field, then it discards the received backup RREQ to prevent route looping; otherwise, it generates a backup RREQ packet. Then, at the destination node, if the received backup RREQ has not been seen before, it generates a backup RREP and forwards it back towards the node which originates the backup RREQ; otherwise, it discards the received backup RREQ. Each node contains two separate routing tables i.e., "Main Route Table" and "Backup Route Table". When a node receives a backup RREQ or a backup RREP, it creates and updates the backup route information in its backup routing table instead of its main routing table. When a node along the main route detects a link failure, it immediately utilizes the backup route and makes it the main route and forwards the data packets through it.

Simulation results show that ENFAT-AODV routing protocol improves the throughput and the average end-to-end delay and it also decreases the control overhead. It achieves better reliability, availability and maintainability of the network.

3.2.2.3 A Reliable Lightweight Multi-Hop WSN Routing Protocol (MAW)

The MAW protocol, presented in [Patel et al. 2009], is a lightweight modified version of AODV [Perkins et al. 2003] routing protocol for unicast routing in WSNs. These modifications enable MAW to:

—Find the shortest path between two nodes.

—Reduce packet size by removing unused fields.

—Reduce the size of the routing table because it stores only reachable neighbors information.

—Mechanisms to avoid data broadcasting storm problems.

—Increase reliability of message delivery by use of node level acknowledgement scheme.

—Reduce control overhead by use of node level message retransmission scheme together with the acknowledgement timeout scheme which helps to reduce the number of route discovery messages.

—Topology independence.

Following is a list of major modifications to AODV:
Message types The following five types of messages defined in MAW protocol:

—Route Request Message (REQ) used to discover a route to the final destination.

—Route Reply Message (REP) used to prepare routing table from destination to the source node.

—Command Message (CMD) used for data query. On receiving this message, the source node initiates route discovery for the destination node defined in the message.

—Data Message (DAT) contains the data requested by the command message. It is sent by the destination node to the source node.

—Route Failure Message (RTF) broadcasted by a node if it fails to communicate with all the nodes in the routing table.

Message structure Unused fields like lifetime, separate sequence number for source and destination nodes are removed to reduce the packet size. This helps to reduce the power consumption because of reduced overhead.
Routing Table Stores only the information about the reachable neighbor nodes in the routing table. This makes the memory footprint smaller which is more suitable for WSN nodes. The routing table has two fields: Node ID (of nodes that are reachable) and Hop Count (number of hops required to reach the destination using corresponding node).

Route Discovery Mechanism On receiving the CMD message, the source node initiates route discovery. It broadcasts a REQ message for the destination defined in the CMD message. The intermediate nodes forward the REQ message till it reaches the destination node. The destination node replies with REP message designated for the source node. Each intermediate node increments the hop count by one in the REP message and stores the hop count and the sender ID in the routing table. Unnecessary data flooding is prevented by the acknowledgement scheme.

Command Message Forwarding Mechanism On receiving the REP message, the source node sends the command message to the nearest neighbor. The nearest neighbor is found by sorting the routing table based on the REP messages and hop count information. The same procedure is followed by the intermediate nodes till the message reaches the destination node. User defined command messages are also allowed.

Data Message Forwarding Mechanism When the destination node receives a CMD message, it responds with a DAT message corresponding to the type of the CMD message. To find the nearest neighbor it uses the same routing table sorting mechanism but chooses the neighbor in the reverse order as message is being sent over the same path but in reverse direction.

Acknowledgement and Acknowledgement Timeout Scheme A node level acknowledgement scheme is implemented. The acknowledgement timeout scheme checks if the acknowledgement is received within a predefined time. If not, then the same message is sent again to ensure message delivery.

Deadlock This happens when a node being used in the routing process fails. This may lead to unnecessary message transmissions by other nodes. It can be handled by limiting the number of retransmission and by the use of acknowledgment timeout scheme. After a specified number of retransmissions, the failed node selects the next nearest neighbor from the routing table. If there is no response from all the nodes in the routing table, the node broadcasts a route failure message.

Livelock This happens when two nearest neighbor nodes in the routing process transmit messages to each other thereby creating an unending loop which leads to reduction in message throughput. This issue can be resolved by comparing the node ID of the recent sender to the previous sender and the message sequence number.

Simulation results show that MAW uses only 12% of total program memory and just 16% of total RAM in MICAz platform. When compared to flooding, MAW achieves more than 90% savings in overall number of transmissions. It is also able to detect and recover from deadlock and livelock problems.

3.2.2.4 Distributed routing in WSN using energy welfare metric (MaxEW)

In social sciences, social welfare is a function of average and equality of an income population. The MaxEW protocol, presented in [Ok et al. 2005], uses this function to define the energy welfare metric which is used as a goodness measure for energy populations. The goal is to achieve both energy efficiency and energy balancing in a WSN. Each sensor node tries to maximize the local energy welfare which leads to globally efficient energy balancing. The protocol is also robust to diverse event generation patterns. Three event generation patterns are considered; uniform, random and repeated event generation.

To achieve improved network lifetime, both energy equality (for energy balancing) and energy welfare (for energy efficiency) are considered. The protocol assumes that the sensor nodes are randomly but uniformly distributed. Each node makes local routing decisions by using the energy welfare metric as a goodness measure. The nodes follow a schedule to coordinate their wake up and sleep times. When the nodes wake up, they share their current energy levels with their neighbor nodes. To conserve energy, nodes sleep when they are inactive. Each sensor node builds a routing table containing identification number, distance to base station and current residual energy of only its neighbor nodes. During the initial setup phase each node finds its direct distance to the base station. It then broadcasts a setup message to all its neighbors containing

its id, distance to base station and current residual energy. This information is stored by all the receiving neighbors for initializing their routing table. During each round, when a sensor node sends data to its neighbors, it also sends a control message containing its id, base station id and current residual energy. The neighbor nodes use this information to update their routing tables. If the neighbor nodes do not get this control message in a round, then the node would be removed from their routing table.

Once the routing table is setup, a node makes the routing decisions to maximize the local energy welfare. Each sending node considers alternative paths that are at most 2 hops to the base station. Based on the residual energy of itself and its neighbors, it selects who would be the best candidate for direct communication with the base station. Before forwarding the data, the sending node first calculates the energy welfare metric of itself and its neighbor nodes to evaluate each of the alternative path. The path that gives the highest energy welfare is selected. As direct transmission may consume more energy, the sending nodes may send the data using indirect multi-hop path if its better. After forwarding the data, the residual energy of the sending node will change and the routing table of the receiving node will be updated with this information. MaxEW also uses a mechanism to guarantee loop free routing paths.

Simulation results show that MaxEW performs better than other algorithms like direct communication, minimum transmission energy approach and self-organized routing [Rogers et al. 2005]. It achieves better energy balancing and is also robust to various event generation patterns.

3.2.2.5. Traffic-Aware Routing Protocol for WSN (TARP)

The TARP protocol, proposed in [Park et al. 2010], is traffic aware routing protocol based on a lightweight genetic algorithm. The sensor nodes are aware of the data traffic rate to monitor the network congestion. To avoid heavy traffic congestion, the data forwarding nodes are selected based on dominant gene sets in a genetic algorithm. The objective is to increase the efficiency and reliability of data transmission by reducing buffer overflow.

Figure 4 shows a flowchart for genetic algorithm. The solution to a complex problem is represented by the initial population of chromosomes. Next, the algorithm evaluates the fitness of original chromosomes and selects a parent chromosome to apply genetic operations. The algorithm then evolves the population through genetic operations like crossover and mutation. To select only the superior objects, the algorithm individually applies the fitness function to newly created objects. If no superior object is found then the algorithm reiterates.

Representation of Chromosome A representation of chromosome is needed for genetic algorithm, so a structure of real numbers is used to represent the chromosomes. The chromosomes are created based on the information about neighbor sensor nodes.

A FIT (FITness) value is calculated using the fitness function and an optimal chromosome is selected based on this value. The genetic algorithm decides superior chromosomes from the solution sets based on this FIT value. Next, the information about the selected chromosomes is sent to the child sensor nodes based in which they distribute the traffic loads to their neighbor nodes. CID (Child node ID) and CR (Child nodes data transfer rate) fields represent the child sensor nodes and their data transfer rate. NID and NR keeps the information about the neighbor nodes that are 2 hops away from the child sensor node. Several solution sets are built based on the FR (Forwarding Rate), NID (Neighbor node ID) and NR (Neighbor nodes data transfer Rate) values. Then, an optimal chromosome is chosen from these solution sets to forward the congestion traffic. For selecting the data forwarding nodes, TARP uses the data traffic rates of nodes that are within 2 hops of a child sensor node.

Fitness Function TARP manages traffic by changing the network topology. It distributes the congested traffic over the sensor nodes which reduces the probability of data loss due to queue overflow. The fitness function is defined as:

$$ADT = \sum_{i=1}^{n-1} \frac{(NR_i + FR)}{n}$$

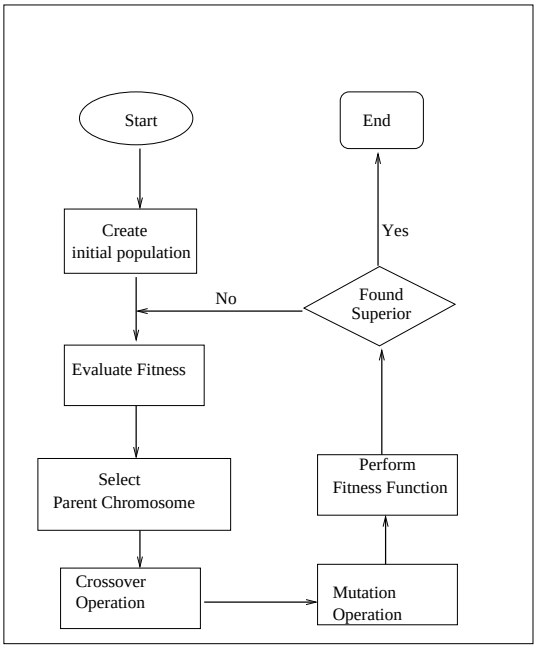

Figure 4: Genetic algorithm

$$Fitness^{-1} = \sqrt{\sum_{i=1}^{n-1} \frac{|ADT - (NR_i + FR)|^2}{n}} + ADT$$

Where, n = number of sensor nodes, ADT = Average Data Traffic of neighbor sensor nodes, NR = Neighbor nodes data transfer Rate, and FR = Forwarding Rate,

The fitness is inversely proportional to the average and the standard deviation of the data transfer rates of each sensor node. The sensor nodes with a high ADT signifies heavy traffic load.

Genetic Operations Fitness value of the chromosomes are modified by applying genetic operations like crossover and mutation to the chromosomes included in the solution set. As the algorithm progresses, the solution set will be composed of chromosomes with high fitness values. The data forwarding rate of the node (which is located in the bottleneck of the routing path) is transferred into new data forwarding rate. In case of traffic congestion, sensor node forwards its data traffic to the neighbor nodes based on the new mutated data forwarding value and then the data transfer rates of the neighbor nodes are also changed.

Operation of TARP - Each sensor node is aware of the data transfer rate of the neighbor nodes and it periodically broadcasts its own information which is stored by the neighbor nodes. The broadcasting is limited to the child sensor nodes to avoid broadcasting storm. When an events occurs, the surrounding sensor nodes start transferring the sensing data to the sink node. Once the sensor node is aware of the traffic congestion (if it is located in the bottleneck of the routing path), it sends the traffic congestion message to the child nodes in order to distribute the input data traffic. When the child nodes receive the congestion message, they send information about their neighbor nodes (that are located within 2 hops) back to the node (parent) which sent the congestion message. On receiving this information, the parent node creates chromosomes and allocates the data forwarding rate. Once the chromosome with highest fitness value is selected, the result is sent to the child nodes. Based on the forwarding rate value, the traffic is distributed among the child nodes to avoid traffic congestion at the parent node.

3.2.2.6. GPS Free Coordinate Assignment and Routing in WSN (VCap)

There are several localization techniques in WSNs. Many of these techniques require the presence of a GPS device in all or some (anchors) of the sensor nodes. However, the use of GPS devices make the sensor nodes expensive and hence a large scale deployment may become prohibitively expensive. Therefore there are several other localization techniques which do not depend on GPS. Virtual Coordinate assignment protocol (VCap), proposed in [Caruso et al. 2005], is one such solution which is used for constructing a hop distance based virtual coordinate system without the need for knowing the physical location of sensor nodes. It achieves this by first selecting three anchor nodes and then all the other nodes in the network store their positions in terms of the hop distance from these anchors.

VCap assumes the network to have a large number of uniformly distributed static nodes. The sensor nodes are also assumed to be homogeneous i.e., have same transmission range and available energy. Each node is required to have a unique ID which can be pre-assigned or randomly generated. The protocol has 4 phases which are used to elect the anchor nodes. It begins with first electing an anchor W which is only used for electing the other three anchor nodes X, Y and Z. At the end, each sensor node in the network is assigned a coordinate (x, y, z) which is its hop distance from anchors X, Y and Z respectively.

Following are the phases of VCap protocol:

Election of W Initially, the sink node generates a W_SET message which contains a hop counter. This counter, called as w coordinate, is initially set to 1 and is incremented by the forwarding nodes. If a node receives multiple W_SET messages, then it forwards the message with smallest w count.

Election of X This phase starts once all the nodes have been assigned a w coordinate. All the nodes with largest value of w within two-hop neighborhood (or the largest ID in case of a tie) decide that they are at the boundary and generate a X_SELECT message. The X_SELECT message contains their unique ID, w coordinate and a hop counter (called x coordinate) initialized to 1. The value of x coordinate is incremented by the forwarding nodes. If a node receives multiple X_SELECT messages, then it forwards the message with largest w (or the largest ID in case of a tie).

Election of Y This phase starts once the Election of X phase is complete. It is similar to the previous phase except for the difference in rule used to determine the eligibility of nodes to become Y. All the nodes with the largest value of x within two-hop neighborhood and whose w is greater than some constant will generate a Y_SELECT message. In case of a tie the node with largest ID will generate the message. The X_SELECT message contains their ID, x coordinate and a hop counter (called y coordinate) initialized to 1. The value of y coordinate is incremented by the forwarding nodes. If a node receives multiple Y_SELECT messages, then it forwards the message with largest x (or the largest ID in case of a tie).

Election of Z In this phase a Z_SELECT message is generated by the nodes that satisfy a given rule $\phi(x, y)$ and with largest value of w within two-hop neighborhood. The Z_SELECT message contains the node ID. If a node receives multiple Z_SELECT messages, then it forwards the message with largest ID.

Once these 4 phases are complete, each node is assigned a coordinate (x, y, z) which completes the virtual coordinate system. The use of W ensures that the X, Y and Z anchors are close to the boundary. In reality the same coordinates are shared by a set of nodes called a zone. So by using a geographic routing protocol, the messages are delivered to the destination zone. Within a zone, the packets are delivered to the destination node using proactive ID based approach.

Simulation results show that the performance of a simple greedy geographic routing with virtual coordinates is slightly less when compared with routing in case of physical coordinates. However, the difference is negligible at higher network densities.

3.2.2.7 Distributed Algorithms for Maximum Lifetime Routing in WSN (DAMLR)

In the Distributed Algorithms for Maximum Lifetime Routing in WSNs paper [Madan et al. 2006], the authors propose two routing algorithms, namely, partially distributed and completely distributed algorithms, for maximizing the lifetime of a WSN. The lifetime of a WSN is considered to be the time at which the first node in the network runs out of its energy. The problem of computing a flow that maximizes the network lifetime is formulated as a linear programming problem and then dual decomposition is used to exploit its separable nature. The paper further describes distributed subgradient algorithms to compute the data rates between each pair of nodes.

The network is assumed to be homogeneous and static with bi-directional links. TDMA is used for synchronizing the send/receive cycle of the nodes. The partially distributed algorithm needs the presence of a central node but has faster convergence at the cost of higher control overhead. Whereas the fully distributed algorithm has lower control overhead as it does not need communication with a central node. However it has slower convergence. Therefore the tradeoff is between faster convergence and lower control overhead.

The algorithms in this paper can also be applied to hierarchical networks. In this case, the algorithms can be used to find an optimal routing path between cluster heads.

4. COMPARATIVE ANALYSIS OF THE PROTOCOLS

In this section we do a comparative analysis of the protocols summarized in the previous section. The comparison is done in terms of the parameters defined in section 2. While efficient energy utilization is the most important consideration in the design of WSN protocols, there are several other application specific design goals that must be achieved. Therefore, not all protocols are designed alike. Some protocols like CBR Mobile-WSN [Awwad et al. 2009] and EERP-NSM [Sarma et al. 2011] are designed to support sensor node mobility, other protocols like AERP Mobile-Sink [Wang et al. 2008] just support sink node mobility. EQSR [Ben-Othman et al. 2010] and MCMP [Huang et al. 2008] protocols are designed primarily to provide QoS guarantees. The UW-ASN [Pompili et al. 2010] protocol is designed specifically for underwater WSNs. EHOR [Eu et al. 2010] is unique in its own right as it is designed for WSNs where nodes are powered by energy harvesting instead of conventional power sources like battery. WSNPDM [Sana et al. 2007] is designed for data collection in the event of a disaster that may result in collapsed or unreachable base stations. Some of the protocols we studied also make use of multipath technique for improving the reliability of data delivery or to meet delay guarantees. The topology of a WSN (flat or cluster based) for which the protocol is designed is another differentiating factor. A similar analysis is done for protocols surveyed in [Al-Karaki et al. 2004] and some of the parameters used in Table 1 are taken from that survey. Following are the parameters (used in Table 1) with respect to which we compare the protocols discussed in Section 3:

Classification the protocol is designed for flat or cluster based WSNs.
Sensor Nodes the sensor nodes in the WSN are identical or heterogeneous, i.e., whether or not all the sensor nodes have identical capabilities.
Transmission the type of transceiver used, i.e., radio, acoustic or optical.
Localization whether the protocol depends upon the location information of senor nodes in the network.
Energy Harvesting the sensor nodes are powered through energy harvesting or they use limited energy sources such as battery.
Mobility the protocol supports mobility of sensor nodes or not.
QoS the protocol provides some sort of quality of service such as delay guarantees or service differentiation between real time and non-real time traffic.
Multipath the protocol forms multiple paths for data forwarding.
Aggregation whether or not data aggregation is used for reducing the amount of duplicate data that is transmitted to the base station.

Query based the base station queries the sensor nodes in the network for collecting specific data.

Error Correction the protocol uses error correction techniques such as forward error correction code (FEC) for reducing retransmission of corrupted data.

While the value of most of these parameters are pretty straight forward to identify, it took a bit of studying to determine the value for scalability. To identify if a protocol is scalable to a large sensing area, we have tried to use the data from simulation models used by the authors for their testing. If this data is not present, we study the underlying design of the protocol to determine if it would be scalable to large sensing area.

Protocol	Classification	Sensor Nodes	Transmission	Localization	Harvesting Energy	Mobility	Scalability	QoS	Multipath	Aggregation	Query based	Error Correction
AERP Mobile-Sink	Cluster	Identical	Radio	Yes	No	Only sink	good	No	No	Yes	No	No
BCDCP	Cluster	Identical	Radio	Yes	No	No	good	No	No	Yes	No	No
CBR Mobile WSN	Cluster	Identical	Radio	No	No	Yes	good	No	No	Yes	Yes	No
CPHRP	Cluster	Identical	Radio	No	No	No	good	No	Yes	Yes	No	No
DAMLR	Flat	Identical	Radio	No	No	Limited	Fair	No	Yes	No	No	No
DGRAM	Flat	Identical	Radio	Yes	No	No	Good	Yes	NA*	No	No	No
EERP-NSM	Cluster	Identical	Radio	Yes	No	Yes	Good	No	No	Yes	No	No
EHOR	Flat	Identical	Radio	Yes	Yes	No	Good	No	NA*	No	No	No
EHRP	Cluster	Identical	Radio	No	No	No	Good	No	No	Yes	No	No
ENFAT-AODV	Flat	Identical	Radio	No	No	No	Fair	No	Yes	No	No	No
EQSR	Flat	Identical	Radio	No	No	No	Fair	Yes	Yes	No	No	Yes
MAW	Flat	Identical	Radio	No	No	No	Fair	No	No	No	Yes	No
MaxEW	Flat	Identical	Radio	No**	No	No	Fair	No	No	No	No	No
MCMP	Flat	Identical	Radio	No	No	Limited	Fair	Yes	Yes	No	No	No
NEAP	Cluster	Hetero	Radio	Yes	No	No	Good	No	No	Yes	No	No
TARP	Flat	Identical	Radio	No	No	No	Fair	No	No	No	No	No
TTSR	Cluster	Hetero	Radio	Yes	No	No	Good	No	No	Yes	No	No
UW-ASN	Flat	Identical	Sound	Yes	No	No	Fair	Yes	No	No	No	Yes
VCap	Flat	Identical	Radio	Yes	No	No	Good	No	No	No	No	No
WSNPDM	Cluster	Hetero	Radio	Yes	No	No	Good	No	No	Yes	No	No

Table I: Feature comparison of protocols

* Both DGRAM [Shanti et al. 2010] and EHOR [Eu et al. 2010] use region based forwarding instead of establishing fixed paths.

** In the MaxEW [Ok et al. 2005], authors mention that each node will *find its direct distance to the base station at the time of initial *setup. But they also mention that the protocol does not make use of any localization scheme.

5. CONCLUSION

Applications of wireless sensor networks are diverse and evolving. This is why we have so many different routing protocols, some of them trying to address challenges that are application specific. Applications may require routing protocols to support mobility of nodes, QoS guarantees (delay/reliability), scalable to large sensing area, absence of localization techniques, etc. WSN

routing protocol must also be able to function with very limited resources of sensor nodes such as energy, processing, memory, transmission range, etc. Therefore, the protocols need to be designed not only for the application requirements, but also keeping these limitations is mind. However, the primary objective of any WSN routing protocol continues to be efficient energy utilization for increasing the network lifetime.

In this paper, we first presented an overview of wireless sensor networks. Next, we discussed some of the important design considerations and goals of routing protocols for WSN. In Section 3, we summarized a diverse set of WSN routing protocols from recent literature. We classified the routing protocols based on topology i.e., cluster based or flat, support for node mobility and quality of service. Finally, we used the design parameters defined in Section 2 do a comparative analysis of these protocols.

REFERENCES

Al-Karaki, J.N and, Kamal, A.E. 2004. Routing techniques in wireless sensor networks: a survey. IEEE Wireless Communications, 11, 6, (Dec. 2004), 6-28.

Yick, Y., Mukherjee, B., and Ghosal, D. 2008. Wireless sensor network survey. The International Journal of Computer and Telecommunications Networking, 52, 12 (August 2008), 2292-2330.

Radi, M., Dezfouli, B., Bakar, K. A., and Lee, M. 2012. Multipath Routing in Wireless Sensor Networks: Survey and Research Challenges. Sensors, 12 (2012), 650-685.

Perkins, C., and Bhagwat, P. 1994. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proceedings of the ACM '94 Conference on Communications Architectures, Protocols and Applications, 1994.*

Johnson, D. B., and Maltz, A. 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. MOBILE COMPUTING, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, 153 - 181.

Perkins, C., and Royer, E. M. 1999. Ad-hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, February 1999, pp. 90-100.*

Kim, D. S., and Chung, Y. J. 2006. Self-organization routing protocol supporting mobile nodes for wireless sensor network. In *Proceedings of the 1st Int. Multi-Symposiums on Computer and Computational Sciences (IMSCCS'06), 2006.*

Awwad, S.A.B., Ng, C.K, Noordin, N. K. and, Rasid, M. F. A. 2009. Cluster Based Routing protocol for Mobile Nodes in Wireless Sensor Network. In *Proceedings of International Symposium on Collaborative Technologies and Systems, 18-22 May 2009, 233-241.*

Sarma, H. K. D., Kar, A., and Mall, R. 2011. Energy efficient routing protocol for Wireless Sensor Networks with Node and Sink mobility. In *Proceedings of IEEE Sensors Applications Symposium (SAS), 22-24 Feb. 2011, pp. 239-243.*

Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. 2000. Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *In Proceedings of the 33rd Hawaii International Conference on System Sciences, 2000.*

Wang, Y-H., Yu, C-Y., Chen, W-T., and Wang, C-X. 2008. An Average Energy based Routing Protocol for mobile sink in Wireless Sensor Networks. In *Proceedings of First IEEE International Conference on Ubi-Media Computing (July 31-Aug. 1 2008), 44-49.*

Mollanejad, A., Khanli, L. M., Zeynali, M., Bahrbegi, H., and Alasti, A. A.. 2010. EHRP: Novel energy-aware hierarchical routing protocol in wireless sensor network. In *Proceedings of International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), (18-20 Oct. 2010), 970-975.*

Younis, O., and Fahmy, S. 2004. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Transactions on Mobile Computing, 3, 4 (Oct.-Dec. 2004), 366-379.

Lin, J., and Liao, M. 2010. A clustering patch hierarchical routing protocol for wireless sensor networks. In *Proceedings of 5th International Conference on Computer Science and Education (Aug. 2010), 941-948.*

Golsorkhtabar, M., Nia, F. K., Hosseinzadeh. M, and Vejdanparast, Y. 2010. The Novel Energy Adaptive Protocol for heterogeneous wireless sensor networks. In *Proceedings of 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT) (9-11 July 2010), vol.2, 178-182.*

Sana, S., and Matsumoto, M. 2007. Wireless Sensor Network Protocol for Disaster Management. In *Proceedings of Information, Decision and Control (IDC 2007) (12-14 Feb. 2007), 209- 213.*

Muruganathan, S. D., Ma, D. C. F., Bhasin, R. I., and Fapojuwo, A. 2005. A centralized energy-efficient routing protocol for wireless sensor networks. IEEE Communications Magazine, 43, 3 (March 2005), 8-13.

Heinzelman, W. B., Chandrakasan, A. P., and Balakrishnan, H. 2002. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on Wireless Communication. 1, 4 (Oct. 2002), 660-670.

Lindsey, S., Raghavendra, C., and Sivalingam, K. M. 2002. Data Gathering Algorithms in Sensor Networks using Energy Metrics. IEEE Transactions on Parallel and Distributed Systems, 13, 9 (Sept. 2002), 924-935.

Du, X., Guizani, M., Xiao, Y., and Chen, H-H. 2007. Two Tier Secure Routing Protocol for Heterogeneous Sensor Networks. IEEE Transactions on Wireless Communications, 6, 9 (September 2007) 3395-3401.

Intanagonwiwat, C., Govindan, R. and Estrin, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of ACM MOBICOM (Aug. 2000), 56-67.*

Shanti, C., and Sahoo, A. 2010. DGRAM: A Delay Guaranteed Routing and MAC Protocol for Wireless Sensor Networks. IEEE Transactions on Mobile Computing, 9, 10 (Oct. 2010), 1407-1423.

Pompili, D., Melodia, T., and Akyildiz, I. F. 2010. Distributed Routing Algorithms for Underwater Acoustic Sensor Networks. IEEE Transactions on Wireless Communications, 9, 9 (September 2010), 2934-2944.

Ben-Othman, J., and Yahya, B. 2010. Energy efficient and QoS based routing protocol for wireless sensor networks. Journal of Parallel and Distributed Computing, 70, 2 (August 2010), 849-857.

Huang, H., and Fang, Y. 2008. Multiconstrained QoS multipath routing in wireless sensor networks. Wireless Networks, 14, 4 (August 2008), 465-478.

Ganesan, D., Govindan, R., Shenker, S., andEstrin, D. 2001. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. ACM SIGMOBILE Mobile Computing and Communications Review. 5, 4 (October 2001), 11-25.

Eu, Z. A., Tan, H-P., and Seah, W. K. G. 2010. Opportunistic routing in wireless sensor networks powered by ambient energy harvesting. Computer Networks, 54, 17 (December 2010) 2943-2966.

Che-Aron, Z., Al-Khateeb, W.F.M., and Anwar, F. 2010. ENFAT-AODV: The fault-tolerant routing protocol for high failure rate Wireless Sensor Networks. In *Proceedings of 2nd International Conference on Future Computer and Communication (ICFCC), (21-24 May 2010) vol. 1, 467-471.*

Eu, Z. A., Tan, H-P., and Seah, W. K. G. 2009. Routing and relay node placement in wireless sensor networks powered by ambient energy harvesting. In *Proceedings of IEEE WCNC, 2009.*

Perkins, C., Belding-Royer, E., and Das, S. 2003. Ad hoc On Demand Distance Vector Routing (AODV), RFC 3561, July 2003.

Patel, K., Chern, L. J., Bleakley, C. J., and Vanderbauwhede, W. 2009. MAW: A Reliable Lightweight Multihop Wireless Sensor Network Routing Protocol. In *Proceedings of International Conference on Computational Science and Engineering (29-31 Aug. 2009), vol.2, 487-493.*

Ok, C., Lee, S., Mitra, P., and Kumara, S. 2010. Distributed routing in wireless sensor networks using energy welfare metric. Information Sciences, 180, 9 (May 2010), 1656-1670.

Rogers, A., David, E., and Jennings, N. R. 2005. Self-organized routing for wireless microsensor networks. IEEE Transactions on Systems, Man and Cybernetics, Part A, 35, 3 (2005), 349-359.

Park, C., and Jung, I. 2010. Traffic-Aware Routing Protocol for Wireless Sensor Networks. In *Proceedings of International Conference on Information Science and Applications (ICISA) (21-23 April 2010), 1-8.*

Caruso, A., Chessa, S., De, S. , and Urpi, A. 2005. GPS free coordinate assignment and routing in wireless sensor networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (13-17 March 2005), Vol.1. 150-160.*

Madan, R., and Lali, S. 2009. Dissemination and harvesting of urban data using vehicular sensing platforms. IEEE Trans. on Vehicular Technology. 58, 2 (Feb. 2009), 882-901.

Amrita Jyotiprada received her M.S degree in Computer Science from University of Kentucky in 2014. She received her B.E degree in Computer Science and Engineering from Biju Patnaik University of Technology, India in 2006. She has over 5 years of experience in software development and her academic interests are in software quality assurance.

Navjeet Sandhu received his M.S degree in Computer Science from University of Kentucky in 2014. He received his B.E degree in Computer Science and Engineering from Biju Patnaik University of Technology, India in 2006. He has over 6 years of experience in software design and development. His academic interests are in software design. He currently works as a Software Engineer for IBM.

Dr. D. Manivannan is currently an associate professor of Computer Science at University of Kentucky, Lexington, Kentucky, USA. He received an M.Sc degree in mathematics from University of Madras, Madras, India. He received M.S and PhD degrees in computer and information science from The Ohio State University, Columbus, Ohio, in 1993 and 1997 respectively. He published his research work in the following areas: fault-tolerance and synchronization in distributed systems, routing in wormhole networks, routing in ad hoc networks, channel allocation in cellular networks, wireless personal area networks and sensor networks. Dr. Manivannan has published more than 50 articles in refereed International Journals (most of which were published by IEEE, ACM, Elsevier, and Springer) and International Conferences.

Dr. Manivannan is on the Editorial board of IEEE Transactions on Parallel and Distributed Systems, IEEE Communications Magazine, Information Sciences journal, Wireless Personal Communications journal, International Journal On Advances in Telecommunications, International Journal On Advances in Networks and Services and International Journal On Advances in Systems and Measurements. He served as a program chair for two International Conferences and served as program committee member for over 30 International Conferences. He also served as reviewer for more than 30 International Journals published by ACM, IEEE, Elsevier, Springer, Oxford University Press and others. He also served on several proposal review panels of US National Science Foundation and as external tenure reviewer for other universities.

Dr. Manivannan is a recipient of the Faculty CAREER Award from the US National Science Foundation. He is a senior member of the IEEE and a senior member of the ACM.