CHAPTER 3

LITERATURE REVIEW OF TOOLS AND TECHNIQUES

3.1 SENSOR NETWORK ARCHITECTURE

A sensor network has a large number of sensor nodes, which are deployed densely in the area of interest and one or more data sinks or base stations that may be located close to or inside the sensing region, as shown in Figure 3.1. The sink sends queries or commands to the sensor nodes in sensing region while the nodes collaborate to accomplish the sensing tasks and send the sensed data to sink. Meanwhile the sink also serves as a gateway to outside networks e.g. Internet, it collects data from sensor nodes, performs simple processing on it and after that sends the relevant information via Internet to the user.

For sending data to base station sensors may use a single hop long distance transmission, which leads to the single hop network architecture as shown in Figure 3.3. However long distance transmission is costly in terms of energy consumption. Energy used in communication is much higher than sensing and computation. For example energy consumed for transferring one bit of data to a receiver at 100m away is equal to the needed to execute 3,000 instructions.
The ratio of energy consumption to communicate one bit over wireless medium to process the same bit is in the range of 1,000 – 10,000 [4]. Hence transmission energy dominates the total energy consumed in communication and as a result the transmission power increases exponentially with transmission distance. Thus, volume of traffic and transmission distance should be decreased to prolong the network life. For this purpose, multihop short-distance communication is highly preferred. In most sensor networks, sensor nodes are densely deployed and neighbour nodes are close to each other, which makes it feasible to use short-distance communication. In multihop communication, a sensor node transmits its sensed data toward the sink via one or more intermediate nodes, which can reduce the communication energy. The architecture of a wireless sensor network can be organized into two types: flat and hierarchical.

![Figure 3.1 Sensor Network Architectures [4]](image-url)
3.2 FLAT ARCHITECTURE

In flat network, each node plays the same role in performing a sensing task and all the nodes are peers. Global identification of sensor nodes is not feasible in a sensor network because of their large number. Data gathering is usually accomplished by using data-centric routing. The sink transmits a query to all nodes in the sensing region via flooding and only the sensor nodes that have the data for this query, will respond to the sink. Figure 3.2 illustrates the typical architecture of a flat network. The well known protocols considered in flat based routing are: Sequential Assignment Routing (SAR), Directed Diffusion (DD), Energy Aware Routing (EAR) etc.

![Figure 3.2 Flat Network Architectures [4]](image)

SAR (Sequential Assignment Routing) [31] was one of the first protocols for WSN which considered QoS issues for routing decisions. SAR algorithm minimizes the average weighted quality of service metrics in network life. SAR
makes a routing decision based on three factors: energy resources, QoS planned for each path, and the packet’s traffic type, which is implemented by a priority mechanism. To resolve reliability problems, SAR uses two systems consisting of a multipath approach and localized path. The multipath tree is defined through the avoidance of low-energy nodes and quality of service is guaranteed while taking into account that the root tree is located in the source nodes set and it ends in the link nodes set. In other words, SAR creates a multipath table whose main objective is to obtain energy efficiency and fault tolerance. Fault tolerance and easy recovery are ensured by the protocol, however it suffers from certain overhead when tables and node states must be maintained or refreshed. This problem increases, especially when there are a large number of nodes.

**DD (Directed Diffusion) [32]** is a data-centric and application aware paradigm. Sensors data are named by attribute value pairs. The objective of the directed diffusion paradigm is to aggregate the data coming from different sources by deleting redundancy, which drastically reduces the number of transmissions. This has two main consequences: first, the network saves energy and extends its life. Secondly, it counts on a higher bandwidth in the links near the sink node. The latter factor could be quite persuasive in deciding to provide QoS in real-time applications. The directed diffusion is based on a query-driven model, which means that the sink node requests data by broadcasting interests. Requests can originate from humans or systems and are defined as pair values, which describe a task to be done by the network. The interests are then disseminated through the network. This dissemination sets up gradients to create data that will satisfy
queries to the requesting node. When the events begin to appear, they start to flow toward the originators of interest along multiple paths. This behaviour provides reliability for data transmissions in the network. Another feature of Directed Diffusion is that it caches network data, generally the attribute-value pair’s interests. Caching can increase efficiency, robustness, and scalability of the network.

**REEP (Reliable and Energy Efficient Protocol) [33]** is a fault tolerant protocol, which has been motivated by the existing network layer data-centric routing protocol directed diffusion. REEP consists of five important elements. These are: sense event, information event, request event, energy threshold value and request priority queue (RPQ). A sense event is a kind of query, which is generated at the sink node and is supported by the sensor network for acquiring information. The response to this query is the information event, which is generated at the source node. It specifies the detected object type and the location information of the source node. After receiving this information, request events are generated at the sink node and are used for path setup to retrieve the real data. The real data in any sensor network are, the collected or processed information regarding any physical phenomenon. Each node in REEP uses an energy threshold value for checking which each node agrees or denies to take participation in a path setup. It gives more reliable transmission of any event information or real data. RPQ is a kind of first-in-first-out (FIFO) queue, which is used in each node to track over the sequence of information event reception from different neighbours. It is used to select a neighbour with highest priority in order to request for a path setup when it
is failed in path setup, without invoking periodic flooding. The authors of REEP used four performance metrics like average packet transmission, average data loss ratio, average delay and average energy consumption to analyze and compare the performance of both protocols DD and REEP. The performance of REEP has been found better than directed-diffusion routing protocol.

**EAR (Energy Aware Routing) [34]** is a reactive protocol to increase the lifetime of the network. This protocol maintains a set of path instead of maintaining or reinforcing one optimal path. The maintenance and selection depend on a certain probability, which depends upon how low energy consumption of each path can be obtained. The protocol creates routing tables about the paths according to the costs. Localized flooding is performed by the destination node to maintain the paths alive.

### 3.3 HIERARCHICAL ARCHITECTURE

In hierarchical network, sensor nodes are organized into clusters, where the cluster members send their data to the cluster head and the cluster heads serve as relays for transmitting the data to sink. A node with lower energy can be used to perform the sensing task and sends the sensed data to its cluster head at short distance, while a node with higher energy can be selected as a cluster head to process the data from its cluster members and transmits the processed data to the sink. This process can not only reduce the energy consumption for communication, but also balance traffic load and improve scalability when the network size grows. Since all sensor nodes have the same transmission capability, clustering must be periodically performed in order to balance the traffic load
among all sensor nodes. Moreover, data aggregation can also be performed at
cluster heads to reduce the amount of data transmitted to the sink and improve the
energy efficiency of the network [35]. The major problem of clustering is the
cluster heads selection and its organization [15]. Many clustering strategies have
been proposed in literature for this. For example, according to the distance
between the cluster members and their cluster heads, a sensor network can be
organized into a single-hop clustering architecture or a multihop clustering
architecture as shown in Figures 3.3 and 3.4 respectively [4]. Figure 3.5 shows
multitier clustering architectures [4].

![Figure 3.3 Single-hop Network Architecture [4]](image)
Figure 3.4 Multihop Clustering Architecture [4]

Figure 3.5 Multitier Clustering Architecture [4]
LEACH (Low Energy Adaptive Clustering Hierarchy) [17], [18] is the most popular clustering algorithms with distributed cluster formation for WSNs. It does the selection of cluster head randomly among the nodes during each round. Each round operation is further divided into two phases: set-up phase and steady phase. In set-up phase of a round each sensor node generates a random number between 0 and 1 and if this number is less than threshold value \( T(n) \), then sensor becomes a cluster-head for this round. Threshold value \( T(n) \) is calculated by using the equation (3.1)

\[
T(n) = \begin{cases} 
\frac{p}{1-p \times \left( r \ mod \ \frac{1}{p} \right)} & \text{if } n \in G \\
0 & \text{otherwise}
\end{cases}
\]

Here \( p \) is the desired percentage of nodes that can become the cluster head, \( r \) denotes the current round and \( G \) represents the set of nodes which are not the cluster head in the last \( 1/p \) rounds. Once cluster-heads are selected in a round, they advertise about their status to all other nodes in the network that they are new cluster-heads. When sensor nodes receive this advertisement, they determine the cluster with which they can be associated depending upon the advertisement signal strength of various cluster heads. The sensor nodes inform the appropriate cluster head that they are the members of it and cluster head assigns a TDMA schedule for each of its members to get their data. In steady state phase, the sensor nodes transmit data to their respective cluster head. Each node sends data to respective CH during its allotted time slot. Each node minimizes its energy
consumption further by entering into a sleep mode for the remaining time. Cluster head further aggregates the data before sending it to the base station. Re-clustering of the network is done after a certain time spent in the steady state phase.

**ESCAL (An Energy-Saving Clustering Algorithm Based on LEACH)** [37] uses LEACH as its base but in order to save the energy further cluster heads do not send the aggregated data to the base station directly. Cluster heads send the data to nearby cluster heads, which are closer to the base station and in this way the energy of the cluster heads are conserved by avoiding the long distance transmission. One of the disadvantages of the LEACH is that the cluster heads rotations do not take into account the remaining energy of sensor nodes [81]. A node may not have sufficient energy to complete a round and may be selected as a cluster head.

**An Energy Efficient Routing Scheme in Wireless Sensor Networks** [38] applied both LEACH and a new approach for cluster head selection. When the remaining energy of a node is larger than 50% of the initial energy then LEACH algorithm is used as in equation (3.1). Otherwise a new approach which considers the remaining energy in each node is applied for cluster head selection as shown in equation (3.2).

\[
T(n) = \begin{cases} 
\frac{p}{1-\frac{1}{p} \mod \frac{1}{p}} \times \left( 2 \times p \times \frac{E_{\text{residual}}}{E_{\text{init}}} \right) & \text{if } n \in G \\ 0 & \text{otherwise} 
\end{cases} \tag{3.2}
\]
Here \( p \) is the percentage of nodes that can become the cluster head, \( E_{\text{residual}} \) is remaining energy of a node and \( E_{\text{init}} \) is initial energy of a node. When \( T(n) \) is greater than generated random number between 0 and 1, it becomes a cluster head. After selection of cluster head, cluster formation is carried out. Joining to a cluster is done by calculating a cost which includes the remaining energy and signal power strength of cluster head. A node joins a cluster head, which have the maximum cost value. Cost of a cluster head \( i \) can be calculated by using equation (3.3).

\[
\text{Cost}(i) = CH(i)_{\text{remain}} + CH(i)_{\text{signal}}
\]

Here \( CH(i)_{\text{remain}} \) and \( CH(i)_{\text{signal}} \) denotes the remaining energy and signal strength of cluster head. CH determines a TDMA schedule for its members and informs the member nodes about this schedule. The nodes, then transmit the sensed data to the cluster head during its time slot. A sensor node sends data to cluster head only when a certain condition is satisfied such as “Does the temperature exceed 30 degrees?” If the condition is not satisfied then node goes to sleep mode to reduce the energy consumption.

Cluster based protocols like LEACH have shown a factor of improvements when compared with its previous protocols. Further improvements were done by forwarding the packets to one of its neighbour node only. This method is named as PEGASIS (Power Efficient Gathering in Sensor Information System) [39]. Instead of forwarding the packets from many cluster heads as like in LEACH
protocol, here in PEGASIS each node will form a chain structure to the base station through which the data would be forwarded to the BS.

In PEGASIS energy efficiency is achieved by transmitting the data to only one of its neighbour node. Here the collected data are fused and this will be further forwarded to its immediate one hop neighbor. Since all the nodes are doing the data fusion at its place there is no rapid power depletion of nodes which are present near to the base station. Also in this method each node will get the chance of forwarding the gathered data to the base station. But when the sensor measurements are aggregated to be a single packet, only a fraction of the data generated by the sensor is given to the base station. In some applications when a particular sensor measurement is needed, it fails to give it to base station. But apart from the function of the routing protocol we can make the sensor network database to follow the multi resolution scheme where the aggregated data will be present in the root node and the finer data can be obtained by further tree traversal mechanism. Though the Directed Diffusion [32] and Rumor routing [40] techniques come under tree based approach, but in terms of energy efficiency they lack behinds PEGASIS model.

**H-PEGASIS (Hierarchical PEGASIS) [41]** decreases the transmission delay of packets to base station and suggests a solution for data gathering by using energy delay metric. Two approaches are being used here to avoid signal interference and collisions among the sensors. In the first approach signal coding CDMA is used to avoid collisions. Second approach use spatially separated nodes for transmission at the same time. A chain of nodes with CDMA capable nodes construct a node
chain to from a hierarchical architecture and low level hierarchy nodes transmit the data to upper level hierarchy. Thus, in this way data is transmitted in parallel, which reduces the transmission delay considerably and this delay can be expressed as O (log N).

Figure 3.6 Chain Structure of PEGASIS Protocol

EB - PEGASIS (An Energy-Efficient PEGASIS-Based Enhanced Algorithm in Wireless Sensor Networks) [42] is an energy efficient chaining algorithm in which a node will consider the average distance of formed chain. If the distance from the closest node to its upstream node is longer than distance thresh (the distance thresh can be obtained from an average distance of formed chains), the closest node is a "far node". If the closest node joins the chain, it will emerge a "long chain". In this condition, the "far node" will search a nearer node on formed chain. Through this method, the new protocol EB-PEGASIS avoids "long chain" effectively. EB-PEGASIS avoids the dying of some nodes earlier than others to
prolong the lifetime of the sensor network. It not only save energy on sensors, but also balance the energy consumption of all sensor nodes

**TEEN (Threshold sensitive Energy Efficient Sensor Network protocol) [43]** is a protocol which responds immediately to any sudden change in a sensed attributes e.g. temperature, pressure etc. This type of responsiveness is important for time-critical applications where network operates in a reactive mode. TEEN uses data-centric mechanism along with a hierarchical approach for its operation. The nodes which are close to each other, form a cluster and this process continue to next level until the sink is reached. Fig 3.7 depicts the operation of TEEN. After the construction of clusters, the cluster head broadcasts two threshold values: hard and soft thresholds for sensed attributes. The minimum value which triggers a node for transmitting the sensed data to its cluster head is called the hard threshold value. Hence the hard threshold value represents the range of interest of sensing value and reduces the number of transmissions to cluster head significantly. Whenever a node senses a value beyond the hard threshold, it will transmit the data again only when the sensed attributes change by a value equal to or greater than other value known as the soft threshold. Thus the soft threshold further reduces the number of transmissions when there is a very little or no change in the sensed attribute value. For controlling the number of packet transmissions one should adjust hard and soft threshold values. The disadvantage of TEEN is that it cannot be used where periodic reports are required because the user will not get any data if the thresholds have not reached.
APTEEN (Adaptive Threshold Sensitive Energy Efficient Sensor Network protocol) [44] is an extension to TEEN which capture data periodically and also reacts to time critical events. The architecture of the protocol has been just similar to TEEN whereby nearby nodes from clusters. Cluster heads broadcast transmission schedule to all its members along with hard and soft threshold of sensed attributes. Cluster head further performs data aggregation to save the energy of the network. APTEEN uses three different query types, namely: historical, one-time and persistent. Historical query analyses the past data, one-time query is for taking a network snapshot view and persistent query is for monitoring an event for a period of time. The experiment results have shown that energy dissipation and network lifetime of APTEEN are between LEACH and TEEN. Multiple level cluster formations and over head complexity are the main drawbacks of this algorithm.

SEP (Stable Election Protocol) [45] is an extension of the LEACH protocol for heterogeneous network. In SEP a small fraction of the nodes has more power than the normal nodes to create a heterogeneous network. For prolonging the stable region, SEP maintains energy consumption in a balanced manner.

DB-SEP (Distance-based Stable Election Protocol) [46] is a variation of the stable election protocol which elect cluster heads on the basis of initial energy and the distance of nodes from the base station.

EECDA (Energy Efficient Clustering and Data Aggregation Protocol for Heterogeneous Wireless Sensor Networks) [47] proposes a three level heterogeneous model where some percentages of the nodes have more energy
than the normal nodes which are known as advanced nodes. Further in advanced nodes some fractions of the nodes have even more energy than the normal nodes which are known as super nodes. EECDA is the protocol for heterogeneous WSNs which improve the network lifetime and stability by combining cluster based routing and data aggregation techniques. Novel cluster head election and a path of a maximum sum of energy residues are used for data transmission in EECDA for achieving the objectives.

![Hierarchical Clustering in TEEN and APTEEN](image)

**Figure 3.7: Hierarchical Clustering in TEEN and APTEEN [43]**

**HEED (Hybrid energy-efficient distributed) [48]** algorithm is a distributed clustering algorithm for WSN. It favours nodes with high residual energy to become the cluster heads and periodically executes re-clustering to achieve load
balancing. So, the nodes that have become cluster heads once will have a low probability of becoming cluster heads again, which ensures that the entire node will carry the role of being a cluster head equally. HEED uses node degree as a fitness function if the requirement is to distribute the load among the cluster heads and the inverse of the node degree if the requirement is to create dense clusters. The mean of the minimum power levels required by all the nodes should be within the node's transmission range to reflect the communication cost within a cluster. In the clustering phase, node sets its probability for becoming a cluster head by using the following equation.

\[ p_i = C_{prob} \times \frac{E_{residual}}{E_{max}} \]  \hspace{1cm} (3.4)

Where, \( E_{max} \) corresponds to a fully charged battery, \( E_{residual} \) is the current residual energy and \( C_{prob} \) is the initial percentage of nodes that can become cluster head, it only limits the initial cluster head announcements and will not have direct impact on final clusters. Each node iteratively doubles its probability until its probability reaches to 1. It announces itself as a cluster head and the nodes which hear the ADV message withdraw itself from the election process and have joined this advertised cluster head. If it hears from more than one cluster, a regular node breaks ties, according to one of the above fitness function. The announcement messages are delayed based on the node's residual energy, meaning that the nodes with high residual energy will advertise themselves before the low-energy ones do. HEED assumes that a node has two levels of transmission range: low-level transmission range for intra-cluster communication
and a high-level transmission range for inter-cluster communication that should be at least double the low-level one to ensure inter-cluster communication since HEED does not adapt the use of gateways to provide the desired connectivity. HEED is completely distributed, terminates within a fixed number of iterations, produces well-distributed clusters over the field in terms of cluster size, scales for very large networks, and significantly increases the network lifetime.

**DAECC (Density-Aware Energy-Efficient Clustering) [49]** is a novel clustering algorithm for non-uniformly distributed sensor networks to save energy and prolong the network life.

**CELRP (Cluster Based Energy Efficient Location Routing Protocol) [50]** is a protocol which utilizes the high-energy of the base station to perform most energy efficient task. By using BS, the sensor nodes are relieved from performing the energy intensive computational task, such as cluster setup, cluster head selection and routing formation. The sensor nodes have made clusters and divided into different quadrants. Each quadrant contains two clustering and sensor nodes that transmit data with two hops data transmission. CH is selected based on the node with maximum residual energy and minimum distance to the base station in each cluster. The CH with the highest energy residual is chosen as the CH leader between all the other CHs. The nodes send data to the CH, and finally the CH sends data to the BS.

**A Novel Energy Efficient Routing Algorithm for Hierarchically Clustered Wireless Sensor Networks [51]** is a protocol in which sensor nodes are hierarchically divided into different levels using the hop number of transmissions
to base station. CHs are selected autonomously and communicate with the base
station using multi-hop transmissions which employ the multi-hop planar model,
whereas non-CH sensor nodes communicate with CH sensor nodes directly.

DEEC (Design of a Distributed Energy-Efficient Clustering Algorithm for
Heterogeneous Wireless Sensor Networks) [52] is an energy-aware adaptive
clustering protocol for heterogeneous networks. In DEEC, every sensor node
independently elects itself as a cluster-head based on its initial energy and residual
energy. To control the energy expenditure of nodes by means of adaptive
approach, DEEC use the average energy of the network as the reference energy.
Thus, DEEC does not require any global knowledge of energy at every election
round. Unlike SEP and LEACH, DEEC can perform well in multi level
heterogeneous networks.

EEHC (Energy Efficient Heterogeneous Clustered Scheme for Wireless
Sensor Networks) [53] is a robust protocol to improve the lifetime and
performance of the network system. EEHC uses weighted probability for the
election of cluster heads. Simulations results show that EEHC has extended the
lifetime of the network by 10% as compared to LEACH in the presence of the
same setting of powerful nodes in a network. Hence, the performance of the
proposed system is better in terms of reliability and lifetime.

3.4 LIMITATIONS OF EXISTING ROUTING ALGORITHMS

Based on the network architecture, routing algorithms can be divided into two
categories: flat routing and hierarchical routing. In flat, routing protocols are
similar to point to point networks; all the nodes play the same role and cooperate
with each other to accomplish a sensing task. A major disadvantage of the flat
routing protocols is that they are generally based on a data-centric approach. This
causes scalability problems as well as increased congestion among the nodes
closer to the sink. Distributed aggregation mechanisms are necessary for
decreasing the information content, flowing in each part of the sensor network.
Protocol such as directed diffusion is applicable to a subset of applications in
WSNs, since the communication is initiated by queries generated from the sink.
Thus Directed Diffusion (DD) is not a good choice for dynamic applications,
where continuous data delivery is required. Moreover, the query types as well as
the interest matching procedures need to be defined for each application.
Furthermore, the data-centric approach of flat topology results in application-
dependent naming schemes. Therefore each change in the application, the
schemes should be defined a priori. Flat routing protocols are effective for
same-scale network, but not good for large-scale networks. They generate more
data processing and bandwidth usage in large-scale networks.

Hierarchical or cluster based routing protocols divide the entire sensing region
into a lot of clusters according to the specific requirements. Each cluster has a
leader known as cluster head, which communicates directly with the sink for data
transmission.

A hierarchical architecture typically comprises two layers of routing where one
layer is used to select cluster-heads and the other is for routing [79], [80], [81],
[82]. The hierarchical cluster structures facilitate the efficient data gathering and
aggregation independent to the growth of the WSN, and generally reduce the total amount of communications as well as the energy spent. Cluster-based routing protocols have a variety of advantages, such as more scalability, less load, less energy consumption and more robustness [81]. Despite their advantages, cluster based protocols significantly rely on cluster heads and face robustness issues such as failure of the cluster heads. Moreover, cluster formation requires additional signaling, which increases the overhead in case of frequent cluster head changes. Therefore a tradeoff between increased energy consumption of the cluster heads and the overhead in cluster formation needs to be considered for efficient operation. Furthermore, intercluster communication is still a major challenge for many hierarchical routing protocols. Generally, cluster heads are assumed to directly communicate with the sink using higher transmit power. This limits the applicability of these protocols to large-scale networks, where single hop communication with the sink is infeasible. Thus hierarchical clustering mechanisms are generally required to provide multi-hop intercluster communication [5].
TABLE 3.1: Comparisons of Different Routing Protocols of WSN [82]

<table>
<thead>
<tr>
<th>Routing Protocols</th>
<th>Network Type</th>
<th>Communication Type</th>
<th>Routing Type</th>
<th>Scalability</th>
<th>Energy Efficiency</th>
<th>Latency</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAR</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Flat</td>
<td>Ltd.</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>DD</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Flat/Data Centric</td>
<td>Ltd.</td>
<td>Good</td>
<td>Very High</td>
</tr>
<tr>
<td>EAR</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Flat</td>
<td>Ltd.</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>REEP</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Flat/Data Centric</td>
<td>Ltd.</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>LEACH</td>
<td>Homogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>LEACH-C</td>
<td>Homogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>ESCAL</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>Energy LEACH</td>
<td>Homogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Low</td>
</tr>
<tr>
<td>PEGASIS</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Chain Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Very High</td>
</tr>
<tr>
<td>H-PEGASIS</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Chain Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Very High</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
<td>-------------</td>
<td>------------------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>EB-PEGASIS</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Chain Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Very High</td>
</tr>
<tr>
<td>TEEN</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Cluster &amp; Threshold Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Event based</td>
</tr>
<tr>
<td>APTEEN</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Cluster &amp; Threshold Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Event based</td>
</tr>
<tr>
<td>HEED</td>
<td>Homogeneous</td>
<td>Multi-hop</td>
<td>Cluster Based</td>
<td>Medium</td>
<td>Good</td>
<td>High</td>
</tr>
<tr>
<td>SEP</td>
<td>Heterogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>DB-SEP</td>
<td>Heterogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>DAEEC</td>
<td>Homogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Good</td>
<td>Low</td>
</tr>
<tr>
<td>DEEC</td>
<td>Heterogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Low</td>
</tr>
<tr>
<td>EEHC</td>
<td>Heterogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Low</td>
</tr>
<tr>
<td>EECDA</td>
<td>Heterogeneous</td>
<td>Single-hop</td>
<td>Cluster Based</td>
<td>Good</td>
<td>Very Good</td>
<td>Low</td>
</tr>
</tbody>
</table>

### 3.5 NEED OF A SIMULATOR

Wireless sensor network is one of the hot research topics nowadays. A significant number of network details in WSN are still not standardized. It is very costly to
build a WSNs testbed and further running a real experiment on it is a cumbersome task. Large number of factors affect an experimental result thus on a testbed it is very difficult to create a variation among the multiple parameters under the same conditions. Hence running a real experiment on a testbed is always time consuming and costly. Thus for the development of wireless sensor network simulation is a very good thing. New ideas, schemes and protocols can be evaluated easily by using a simulation tool. One of the important features of a simulation tool is that it can segregate various factors of sensor network and helps in obtaining the optimum performance for various parameters. Thus simulation is good for studying and testing the new applications and protocols for wireless sensor network. This thing has brought a tremendous boom in the field of simulator development. WSNs simulation tools should have two important points before conducting an experiment and they are:

- Simulation model correctness.
- Suitability of a specific tool for implementing a model.

A right model based on solid assumptions is essential for getting the correct results. One of the fundamental tradeoff is: accuracy and necessity of details versus scalability and performance [54], [55]. The rest of this chapter describes the several main-stream WSNs simulators in more detail.

3.6 GLOMOSIM/QUALNET

Global Mobile Information System Simulator (GloMoSim) [56], [57], [58] is a discrete-event simulator for wired and wireless networks. It uses Parsec, which is a C based language for the sequential and parallel execution. GloMoSim and
Parsec both are the product of Parallel Computing Lab at UCLA. Similar to the OSI model, GloMoSim supports layered approach. To communicate between different simulation layers GloMoSim uses a standard API so that new layers and models can be integrated and exchanged easily. Its JAVA based graphical user interface is good for creating the simulation configuration and scenarios. GloMoSim have basic functionality for simulating wireless networks, as well as Adhoc networks (e.g. AODV, DSR). QualNet [59] is a commercial product of GloMoSim and was created in 2000. GloMoSim 2.0 was the last version released under an academic license. QualNet 6.01 is the latest version which includes enhancements such as network security library, WSNs library for ZigBee and new models for energy. QualNet 6.01 key features include scenario designer for creating the various network topologies, visualizer for viewing network scenarios in 2D & 3D, packets tracer to help in debugging, analyzer for statistical analyses and file editor for directly editing the various network scenarios.

3.7 OPNET MODELER WIRELESS SUITE

OPNET Technologies is a leading company that develops and markets software for the simulation, modeling, and analysis of computer networks. IT Guru and Modeler are the most popular products for network simulation. OPNET Technology Inc. has developed a commercial product known as OPNET Modeler Wireless Suite [60] for modeling and simulation of the wireless networks. It is a discrete event simulator and operates on a parallel simulation kernel of 32-bit/64-bit. The OPNET Modeler uses an object-oriented approach with a hierarchical modeling environment. OPNET has not included any special supports for WSN
routing protocols, but support for the various propagation and modulation
techniques including MAC layer & ZigBee (802.15.4) are provided. For modeling
of network, node and process OPNET uses three-tier architecture. Network
consists of node, links and subnets. A node represents a network device and
groups of devices, i.e. servers, workstations, WLAN nodes and IP clouds etc.
Links represent point-to-point and bus-links between the nodes. The node domain
covers the building of blocks also referred as modules, including processors,
queues and transceivers as well as the specification of interfaces between the
modules. The process domain consists of state transition diagrams, clocks of
C-code, OPNET Kernel Procedures (KPs) as well as state and temporary variables
[60].

3.8 TOSSIM

TOSSIM (TinyOS mote simulator) [61]-[63] is a simulator for TinyOS sensor
network. It is designed especially for TinyOS applications running on MICA
motes. TOSSIM captures the behaviour and network interactions not only on
packet level, but also at the network bit granularity. Thus TOSSIM is equally
good for simulating low-level protocols, and top-level applications. TOSSIM with
its external communication system can monitor the transmitted packets or can
inject new packets into the network. Three types of network model are provided
in TOSSIM: simple, static and space connectivity. In simple connectivity one cell
is used for all the nodes, static connectivity creates a network graph at start up and
it will never be changed again and in space connectivity transmission range of the
nodes may be changed and they move randomly in a squared region. TinyViz, a
JAVA based graphical user interface of TOSSIM which is used for visualizations and running all the simulation. Passive observe mechanisms of user interface inspects the radio ranges or debug messages by injecting new packets into the network and it can actively monitor the network.

3.9 OMNET++

OMNeT++ [64]-[66] is an object-oriented discrete event simulation framework for wired and wireless networks. Its generic architecture is good for simulation such as protocol modeling, querying network modeling, modeling of distributed hardware and multiprocessor system, hardware architecture validation etc. OMNeT++ is not a simulator itself but rather it provides tools and infrastructure for writing simulation framework. One of the basic tools of this infrastructure is the component based architecture for simulation models. Reusable components known as modules are integrated together for creating a Model. Modules combined together to form compound modules and are connected through the gates. Module nesting depth is unlimited till the resources of the system permit it. Messages are used by these modules to communicate with each other which may carry an arbitrary data structures. Messages can be passed between modules by using the predefined paths via gates and connections or they may be sent directly to the destination. Direct communication is generally used in wireless simulation. For customizing the modules behaviour or topology, modules have parameters. Lowest level module is known as simple modules and depicts the model behaviour. C++ language is used for the programming of simple modules. Various user interfaces are used in OMNeT++ simulations. For demonstration and
debugging purposes graphical, animating user interfaces are highly preferable, and command line interface is used for batch execution. The simulator along with user interfaces are highly portable and tested on the most commonly used operating systems like Linux, Mac OS/X, and Windows. Parallel distributed simulation is also supported by OMNeT++ and it uses several methods for communicating between different partitions of a parallel distributed simulation. OMNeT++ can also be used for parallel simulation algorithms classroom presentation. The commercially supported version of OMNeT++ is OMNEST. OMNeT++ is available free for academic and non-profit use. For using it commercially, one need to get license from Simulcraft Inc. Mobility Framework, MiXiM, Castalia, INET Framework and NesCT are some of the simulations frameworks based on OMNeT++ that can be used for the simulation of wireless sensor network.

3.10 NS-2

NS-2 (Network Simulator version 2) is a simulator build on OTcl (Object-Oriented extension of Tool Command Language) and C++ language. Initially, NS was focused on wired networks, but over time wireless support was also included in it. NS was developed from REAL network simulator. From 1995 onwards, NS gained support from DARPA (VINT). Currently the development is continued in collaboration with different researchers and institutes, including DARPA (SAMAN), NSF (CONSER) and ICIR. Sun Microsystems and the UCB Daedelus developed the wireless support for Network Simulator. In June 2009 NS-2.34 version was released. Due to its current major version number NS is also
often referred to as NS-2. The main idea of NS-2 is based on the separation of
control and data. OTcl is used for the control, i.e. the scripting of the model
topology and its parameters; C++ is used for the data, i.e. the programming of
each object in the simulation topology. The separations of these two areas are
achieved by the following approach: Each network component object is written
and compiled in C++. For each of these C++-objects a corresponding OTcl-object
is created by OTcl linkage. As a result, the C++ objects are controlled by OTcl.
In spite of the great number of contributing researchers, NS-2 has still low
supports for WSN specific protocols [67], [68]. SensorSim [69] is a framework
build upon NS-2, for modeling WSNs. Power models, sensor channel models,
lightweight protocol stacks for wireless micro-sensors, hybrid simulation and
scenario generation are some of the key features of it. Hybrid simulation mode
integrates the real application support and by using this, real node applications
can be run on a simulated node without doing any changes.

3.11 AVRORA

Avrora [70] is an open-source cycle accurate simulator for embedded sensing
programs. Avrora is language and operating system independent and can be used
for simulating and analysis of the programs written for AVR micro-controllers. It
supports different sensor platforms like Mica2 and MicaZ. Avrora can simulate at
the instruction-level, i.e. a real microcontroller program run in the simulator. This
approach provides an accurate simulation of devices and radio communication.
For example the energy usage can be predicted according to the number of clock-
cycles needed for the used instructions. Avrora is implemented in Java, which
offers great flexibility and portability because the simulation of machine code is operating system independent. Avrora runs one thread per node. The synchronization between the threads is only done when necessary, i.e. only to ensure the global timing and the order of radio communication. For the efficient execution of programs an event queue is utilized, which provides a cycle accurate execution of the device and guaranteed communication behaviour. Avrora scales up to networks of 10,000 nodes and performs 20 times faster than previous simulators with the emulating approach and the same accuracy. Moreover, Avrora contains further tools such as profiling utilities, an energy analysis tool, a stack checker, a control flow graph tool etc. However, Avrora lacks an integrated graphical user interface so that everything has to be done manually on the command line.

3.12 J-SIM

J-Sim [71], [72], [73] is based on the ACA (autonomous component architecture) and uses a component-based compositional simulation environment. ACA basic entities are components and they communicate with each other through their ports by sending and receiving data. The component behavior is specified at design time in contracts – their actual binding will be linked at the system integration time. The loosely-coupled component architecture of J-Sim enables the user to design, implement and test single components individually. New components can be easily added or exchanged for existing ones. On top of ACA a generalized packet switched network model called Internetworking Simulation Platform
(INET), is developed. INET defines the generic structure of a node and network components.

The combination of J-Sim as Java implementation and the component-based architecture makes J-Sim a platform independent, extensible and reusable simulation environment. Moreover, J-Sim provides a scripting interface that enables the use of scripting languages such as Perl, Tcl and Python. In the current release, version 1.3, Jacl, the Java implementation of a Tcl interpreter, is completely integrated into J-Sim. J-Sim provides a dual-language environment similar to NS-2. The basic classes are written in Java, while their linkage is glued together using a scripting language such as Tcl/Java. In comparison to NS-2, the classes/methods/fields do not have to be exported to be accessible by Tcl. A framework for sensor networks simulation is included, which has been built upon ACA, INET and the Wireless Protocol Stack. In the framework the following components are specified: sensor and sink nodes; sensor channels and wireless communication channels; and physical media covering seismic channels, mobility models and power models.

3.13 MATLAB

MATLAB (Matrix Laboratory), developed by MathWorks Inc. is a high performance numerical computation and visualization software package. MATLAB attracts the attention of scientists and researchers due to its strong analytical capabilities, flexibility, reliability, and powerful graphics. Its interactive environment provides hundreds of mathematical functions for providing solutions to a number of mathematical problems, including linear systems, nonlinear
systems, complex arithmetic, matrix algebra, signal processing, differential equation, optimization and different types of scientific computations. MATLAB has very good and easy to learn programming capability. Through external interfaces algorithm and code written in other scientific languages like FORTRAN and C can also be accessed in MATLAB. A number of optional toolboxes are written for special applications like signal processing, system identification, control system design, statistics, neural networks, fuzzy logic, symbolic computations etc.

MATLAB is enhanced by a very powerful Simulink program which can be used for modeling, simulating, and analysing dynamical systems. Simulink can model linear and nonlinear system in a continuous time, sampled time, or a hybrid of the two. Simulink has a powerful GUI and with it models can be built with simple click and drag mouse operations. Simulink GUI allows the user to draw models just as we draw with pencil and paper. Simulink provides a comprehensive block library of linear and nonlinear components, sinks, sources and connectors. Models are hierarchical and they can be customized to create our own blocks. After defining a model, it can be simulated by using integration methods. Scopes and other display blocks can be used for seeing the results when the simulation is running. MATLAB and Simulink are integrated, so that simulation, analysis and revision of models can be done at any point [55], [74].
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GloMoSim/QualNet</td>
<td>C and Parsec</td>
<td>• GloMoSim is freely available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Basic mobility and radio</td>
</tr>
<tr>
<td></td>
<td></td>
<td>propagation models, 802.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• QualNet is Commercial</td>
</tr>
<tr>
<td></td>
<td></td>
<td>version of GloMoSim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Support various battery,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Energy and ZigBee model</td>
</tr>
<tr>
<td>OPNET Modeler Wireless</td>
<td>Configuration by GUI;</td>
<td>• Support for different</td>
</tr>
<tr>
<td>Suite</td>
<td>internals C++</td>
<td>propagation, 802.11, ZigBee</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MANET protocols available</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• No special support for WSN</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Powerful tool with a nice GUI</td>
</tr>
<tr>
<td></td>
<td></td>
<td>but expensive</td>
</tr>
<tr>
<td>TOSSIM (part of TinyOS)</td>
<td>nesC</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td><strong>Simulation of all TinyOS based WSN protocols without modifications are available.</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>OMNeT++</th>
<th>basic modules C++; larger structures NED</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Several frameworks like MiXiM, NesCT Castalia add WSN functionality</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Eclipse-based IDE for development</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Active project with a huge user base</strong></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NS-2</th>
<th>C++; configuration OTcl</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Complex configuration</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Low support for WSN specific protocols</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Unclear situation due to large number of contributions from different users</strong></td>
<td></td>
</tr>
<tr>
<td>Device</td>
<td>Software Details</td>
</tr>
<tr>
<td>-----------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>Avrora</td>
<td>AVR microcontroller binaries</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>J-Sim</td>
<td>Java; configuration Tcl/Java</td>
</tr>
<tr>
<td>MATLAB SIMULINK</td>
<td>C, Java</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.14 REASONS OF MATLAB SELECTION

After reviewing several simulators, MATLAB has been selected as the best simulation tool for doing this research. The selection of MATLAB is based on several factors as listed below:

- MATLAB is a powerful fourth generation programming language and due to its strong programming capability and flexibility it helps in detailed simulation of sensor nodes and their architecture.
- The MATLAB desktop environment allows to work interactively with data, helps to keep track of files and variables and simplifies common programming/debugging tasks.
• The ability to auto-generate C code, using MATLAB Coder, for a large (and growing) subset of mathematical functions

• Communication channel modeling (SNR, effect of different noise schemes, interference, distance, etc.

• Support physical layer parameters, different modulation and encoding techniques.

• Powerful Simulink program for modelling, simulations and analysing dynamic systems

• The simulation results can be put in the MATLAB workspace easily for post processing and visualization.

• Using scopes and other display blocks the simulation results can be seen when the simulation is running.

3.15 CHAPTER SUMMARY

The chapter gives an introduction of wireless sensor network architecture and an extensive review of the protocols stack based on it has summarized. Most important considerations for these routing protocols are energy efficiency and network lifetime. Network architectural design has a big impact on the energy consumption and lifetime of a sensor network. Sensor nodes are energy constraint devices and they are deployed in a hostile environment so many to one traffic pattern with multihop short distance communication are preferred in sensor networks. In multihop networks, a hierarchical network architecture based on clustering can not only reduce the energy consumption for communication, but also balance the traffic load and improve scalability of a sensor network.
Need of a simulator for sensor network research and an extensive review of the existing simulation tools used in the wireless research community is also presented in the chapter. Various wireless network simulators are investigated to find a suitable candidate for this research. The selection of MATLAB for this research is based on several factors, including its acceptability and credibility among the scientists and researchers community. The key advantage of MATLAB is its strong analytical capabilities, flexibility, reliability, and powerful graphics. Next chapter describes the contribution of the thesis towards cluster based routing protocols for the homogeneous sensor network.